

Die T_EXnische Komödie

dante
Deutschsprachige
Anwendervereinigung T_EX e.V.

33. Jahrgang Heft 4/2021 November 2021

4/2021

Impressum

»Die T_EXnische 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-L^AT_EX-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, L^AT_EX-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 LuaHBTeX, Version 1.13.2 (TeX Live 2021) erstellt. Als Standardschriften kamen Libertinus Serif, Libertinus Sans Serif, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2400

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

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

Druck: Schleunungdruck GmbH
Eltertstraße 27, 97828 Marktheidenfeld

Redaktion: Luzia Dietsche (verantwortliche Redakteurin)

Mitarbeit:	Adelheid Bonnetsmüller	Eberhard Lisse	Ralf Mispelhorn
	Rolf Niepraschk	Stefan Pinnow	Bernd Raichle
	Christine Römer	Volker RW Schaa	Martin Sievers
	Herbert Voß		

Redaktionsschluss für Heft 1/2022: 15. Januar 2022

ISSN 1434-5897

Die T_EXnische Komödie 4/2021

Editorial

Liebe Leserinnen und Leser,

jetzt ist es soweit – ich habe meine Ordner mit den Namen »Chefred Interim« in »Chefredaktion« umbenannt. Bin ich doch bei der Wahl während der Mitgliederversammlung, die pandemiebedingt leider noch einmal online stattfinden musste, einstimmig in dieses Amt gewählt worden. Ich bedanke mich für das hierbei in mich gesetzte Vertrauen und hoffe, den Erwartungen gerecht werden zu können.

Tatkräftige Hilfe erhalte ich dabei jedes Mal vom Redaktionsteam und denjenigen, die das Material für unsere Mitgliederzeitung liefern. Diese Ausgabe ließ mich das Sprichwort »des einen Freud ist des anderen Leid« hautnah erleben. So begeistert ich über den Artikel von Herbert Voß bin, so heftig hatte ich mit seiner Einbindung zu kämpfen. Ohne Herberts direktes Eingreifen hätte ich es vermutlich nicht geschafft. Danke Herbert! Auch den Artikel über R und L^AT_EX konnte ich dank direkter Zusammenarbeit zwischen Walter Entenmann und mir erfolgreich zähmen. Danke Herr Entenmann für die unendliche Geduld!

Besonders glücklich bin ich, dass Henning Hraban Ramm uns die Welt von ConT_EXt nahebringt. Für mich ist das zugegebenermaßen völliges Neuland. Ich freue mich auf viele weitere Artikel zu dem Thema, sei es von Henning oder auch von anderen des ConT_EXt Kundigen. Und dann ist da noch die Hoffnung auf weitere Artikel, seien es Original oder Übersetzung, zur Entwicklung und Zukunft von L^AT_EX. Oder auch zum Thema Literaturbe- bzw. -verarbeitung, Indexerstellung, Schriften oder T_EX-Portierungen. Gibt es eigentlich eine T_EX-Anpassung für Android?

Ach, es gibt so viele interessante Themen und ich freue mich über alle Artikel, die mir zur Verfügung gestellt werden. Im Übrigen können auch ganz kurze, knackige durchaus von großem Nutzen für mich sein, so profan es klingt – als Bindeglied zwischen den großen Elaboraten. Abgesehen davon gibt es vieles, was durch einen kleinen Hinweis etliches an Information liefern kann, wie uns Christine Römer in ihrem Beitrag zeigt.

So wünsche ich Ihnen und Euch wieder viel Spaß beim Lesen und verbleibe mit T_EXnischen Grüßen
Luzia Dietsche

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

das Jahr bleibt turbulent. Im letzten Grußwort schrieb ich leichtfertig:

Ich bin durchaus erleichtert, dass wir das so wichtige Thema »Nachfolge Karin« abschließen können. Wir leiten als Verein damit einen Generationswechsel im Büro ein und ich hoffe natürlich sehr, dass Frau Engelmann uns mindestens so lange kompetent, sympathisch und professionell unterstützt und begleitet wie dies Karin getan hat.

Nun musste der Vorstand schon das zweite Mal in diesem Jahr nach einem Nachfolger bzw. einer Nachfolgerin für Karin Dornacher suchen. Der Vorteil dieser Wiederholungsübung lag darin, dass die Suche insgesamt wesentlich kürzer dauerte und auch die vertraglichen Dinge schon erprobt waren.

Aus der im Vergleich zum Frühjahr größeren Zahl an Bewerberinnen und Bewerbern fiel die Wahl auf Frank Gerhard. Er wird ab dem 1.11. an drei Tagen pro Woche im Büro sein. Karin wird noch ein paar Monate lang die Einarbeitung an einzelnen Tagen begleiten, so dass das geballte Wissen rund um die Verwaltung von DANTE e.V. nicht verloren geht und insbesondere die vielen Besonderheiten – mancher mag es Eigenheiten nennen – Berücksichtigung finden können.

Ich möchte mich dafür noch einmal ganz herzlich bei Karin bedanken. Uns allen sollte sehr bewusst sein, welches Glück wir haben, dass ihr der Verein so sehr am Herzen liegt.

Gleichwohl kann es im Einzelfall vorkommen, dass die eine oder andere Bearbeitung einer Anfrage in den ersten Wochen ein paar Tage länger dauert. Ich bitte dafür um Verständnis und bin zugleich sicher, dass Herr Gerhard sich schnell einarbeiten wird.

Ebenfalls ein gewisses Déjà-Vu-Erlebnis konnte man bei der diesjährigen Herbsttagung haben, die am 18. September als Onlinekonferenz stattfand. Wie schon

im Frühjahr war zu beobachten, dass die Mitglieder und T_EX-Interessierten das Angebot zwar erfreulicherweise stärker wahrnehmen, die räumliche Distanz aber sicherlich einiges der sonst üblichen kommunikativen Atmosphäre zunichte macht. Für 2022 planen wir aktuell die »Gegenprobe«, indem wir voraussichtlich nochmals in Magdeburg und Saarbrücken tagen werden, nur dann wieder (überwiegend) in Präsenz.

Einen Bericht zur Herbsttagung (Seite 11) sowie das Protokoll der Mitgliederversammlung (Seite 6) finden Sie in dieser Ausgabe, die Vortragsfolien sind zudem über unsere Website verfügbar.

Diese Ausgabe der Mitgliederzeitung beschließt das erste Jahr mit Luzia Dietsche als verantwortlicher Redakteurin. In Saarbrücken wurde sie nun auch ganz offiziell von der Mitgliederversammlung bestätigt. Ich bedanke mich bei ihr für die Energie, die sie bislang schon in das »Wagnis DTK« gesteckt hat. Das ein oder andere ist bereits in Bewegung geraten und wir Mitglieder sind weiterhin herzlich eingeladen und aufgerufen, die Redaktion tatkräftig zu unterstützen.

Liebe Mitglieder, der November hat begonnen. Bald schon werden die Städte in vorweihnachtliche Lichter getaucht sein und »Last Christmas« wieder zum Dauerbrenner im Radio werden. Dennoch wird auch dieser Jahresabschluss nicht ohne Einschränkungen, nicht »wie früher« ablaufen, sondern immer noch getrübt von Corona sein.

Passen Sie also auf sich auf und bleiben Sie vor allem gesund und optimistisch. Ich wünsche Ihnen schon jetzt besinnliche Feiertage, einen guten Jahresausklang und uns allen für 2022 viele schöne Stunden in geselligen Runden.

In diesem Sinne viel Vergnügen bei der weiteren Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Protokoll der 63. Mitgliederversammlung von DANTE e.V. am 18. September 2021 in Saarbrücken (remote per „Online-Schule Saarland“)

Volker RW Schaa

Zeit: 18. September 2021, 09:15 Uhr – 10:37 Uhr
Ort: Landesinstitut für Pädagogik und Medien
Saarbrücken
remote über „Online-Schule Saarland“
Abstimmungen über Anbieter „Vereins-Abstimmung.de“

Teilnehmer: 42 (nach Anmeldung)
34 (stimmberechtigt anhand der ausgegebenen PINs)

Leitung: Martin Sievers
Vorsitzender von DANTE e.V.

Protokollant: Volker RW Schaa
Schriftführer von DANTE e.V.

Die Mitgliederversammlung wurde entsprechend § 12 Abs. 2 fristgemäß durch Einladung in der Ausgabe 2/2021 von »Die T_EXnische Komödie« einberufen und ist gemäß § 13 Abs. 2 beschlussfähig.

TOP 1: Begrüßung und Tagesordnung

Der Vorsitzende Martin Sievers begrüßt die Teilnehmer der 63. Mitgliederversammlung von DANTE e.V. und stellt den örtlichen Veranstalter Thomas Hilarius Meyer vom »Landesinstitut für Pädagogik und Medien« in Saarbrücken vor. Thomas Meyer begrüßt alle Teilnehmer als Vertreter des Institutes, da alle Offiziellen entweder erkrankt oder verhindert sind. Neben seiner Begrüßung wiederholt er die Einladung nach Saarbrücken für den Herbst 2022 in Präsenz und leitet über zu Alexander König, Landesfachberater für medienbezogene Schul- und Unterrichtsentwicklung (und Vater der »Online-Schule Saarland«), der einen kurzen Einblick in die verwendete Meeting-Software gibt, die als Open-Source-Projekt des Landes verstanden werden kann.

Martin Sievers stellt die Tagesordnung vor:

1. Begrüßung und Tagesordnung
 - Vorstellung des Vorstands
2. Bericht des Vorstands
 - Büro/Infrastruktur
 - Tagungen und Messen
 - T_EX Collection 2021
 - Laufende Projekte
 - Abgeschlossene Projekte
3. Wahl eines/r neuen verantwortlichen Redakteurs/-in für „Die T_EXnische Komödie“
4. Verschiedenes

Die Tagesordnung wird ohne Einspruch akzeptiert.

TOP 1.1: Vorstellung des Vorstands

Die folgenden Vorstandsmitglieder sind »online« anwesend und werden von Martin Sievers vorgestellt: Dr. Uwe Ziegenhagen (stellvertretender Vorsitzender), Dr. Doris Behrendt (Schatzmeisterin), Volker RW Schaa (Schriftführer), Dr. Klaus Höppner (Beisitzer), Harald König (Beisitzer) und Dr. Herbert Voß (Beisitzer).

Stephan Lukasczyk (Beisitzer) ist durch Aufenthalt außerhalb des Mobilfunknetzes verhindert.

Der Verein unterhält in Heidelberg ein Büro, das von Karin Dornacher und Davina Engelman geleitet wird. Beide Damen sind nicht virtuell anwesend, ihnen wird ein Dank ausgesprochen, dass sie während der Corona-Pandemie das Büro und damit eine wesentliche Funktion des Vereins am Laufen gehalten haben.

TOP 1.2: Testabstimmung

Vor der Wahl eines/r neuen verantwortlichen Redakteurs/-in für „Die T_EXnische Komödie“ wird eine Testabstimmung zur Erprobung der Abstimmungssoftware des Anbieters „Vereins-Abstimmung.de“ durchgeführt. Alle Teilnehmer haben vorab entsprechende Zugangsdaten zu der Plattform erhalten. Die Handhabung der Abstimmungsseite wird nochmals erläutert und dann der Test durchgeführt. Die Testabstimmung verläuft einwandfrei.

TOP 2: Bericht des Vorstands

TOP 2.1: Büro – Neueinstellung

Dieses Thema schien erledigt, nachdem zum 1. Juni eine Nachfolgerin für Karin Dornacher eingestellt worden war. Die Einarbeitung hatte mit Unterstützung von Karin stattgefunden, es war vereinbart, dass sie zum 30. September ausscheiden könne. Nun hat sich für Frau Engelmann die unerwartete Situation ergeben, dass sie ein Studium mit Stipendium aufnehmen kann, was leider zu ihrer Kündigung bei DANTE e.V. führte.

Die Stellenausschreibung wurde bereits Ende August mit verkürztem Bewerbungsschluss veröffentlicht. Es gab mehr Bewerbungen als bei der ersten Ausschreibung und am 16. September hat bereits ein Vorgespräch mit einem aussichtsreichen Bewerber stattgefunden. In der kommenden Woche soll ein weiteres Gespräch im Büro folgen. Ein möglicher Arbeitsbeginn ist der 1. November. Es kann also sein, dass das Büro im Oktober nur eingeschränkt besetzt und arbeitsfähig ist.

TOP 2.2: Tagungen und Messen

- TUG-Tagung 2021, 5.–8. August 2021, virtuell. Die Vorträge wurden gestreamt und sind über Youtube verfügbar.
- INFOS 2021 (Informatik – Bildung von Lehrkräften), Wuppertal, 8.–10. September, virtuell
DANTE e.V. war mit einem Stand vertreten. Die Besucherzahl war gering, da die Hürde, sich aktiv in den virtuellen Raum zu begeben, viele wohl abschreckte. Das ist anders als bei Präsenzveranstaltungen, wo man an Ständen vorbeischlendern kann und bei Interesse zuhört. Es ist deshalb fraglich, ob sich dieses Format durchsetzen kann. Uwe Ziegenhagen berichtete von einer früheren Präsenzteilnahme, bei der gute Erfahrungen gemacht worden seien.
- 15. Internationales ConT_EXt-Meeting, Bassenge, Belgien, 20.–25. September wird dieses Jahr wieder als (einzige) T_EX-Präsenztagung stattfinden. Derzeit sind 13 Teilnehmer gemeldet, es wird aber für Mitglieder der ConT_EXt-Gruppe eine Online-Teilnahme geboten.
- GuIT-Meeting 2021, Florenz, 16. Oktober, virtuell

TOP 2.3: T_EX Collection 2021

Die T_EX Collection 2021 wurde erneut als Eigenproduktion mit Beteiligung anderer Usergroups durchgeführt. Im Einzelnen wurden folgende Stückzahlen geordert:

- CSTUG: 40
- Greek T_EX Friends: 10

- GuIT: 60
- GUST: 190
- GUTenberg: 100
- NTG: 110
- UK-TUG: 85
- TUG: 850
- Lehmanns: 20+x

Insgesamt wurden nur noch 4000 Exemplare geordert, da das Interesse abnimmt und insbesondere in Corona-Zeiten keine Veranstaltungen stattfinden, auf denen die DVD als Werbung für $\text{\TeX}$ abgegeben werden können. An die $\text{\TeX}$ -User-Groups wurden pro DVD 0,45 € zzgl. Porto berechnet.

TOP 2.4: Projektförderung

TOP 2.4.1: Laufende Projekte

Förderung durch Sachmittel – Stefan Kottwitz

- Übernahme von Serverkosten (insgesamt 1956 € jährlich)
- jährliche Verlängerung möglich
- Angebote umfassen Foren, Websites und Blogs, u. a.:
 - Foren: <https://latex.org/forum/>, <https://golatex.de/>, <https://texwelt.de/>
 - Websites: <https://texample.net/>, <https://pgfplots.net/>, <https://texdoc.org/index.html>
 - Blogs: <https://texwelt.de/blog/>, <https://tikz.de/>, <https://texblog.net/>, <https://latex.net/>, <https://latex-cookbook.net/>, <https://tex-talk.net/>, <https://texlive.net/>

Neu ist zum Beispiel der $\text{\LaTeX}$ -Online-Compiler, der es Nutzern und Entwicklern erlaubt, direkt Code-Schnipsel zu kompilieren. Da die Arbeit an dieser Vielzahl von Angeboten ständig wächst, freut sich Stefan über Verstärkung in allen Belangen, sei es die Mitarbeit an einem virtuellen Server, Hilfe bei Datenbank-Services oder Upgrades, E-Mail- oder Forenbetreuung.

TOP 2.4.2: Abgeschlossene Projekte

Fonterweiterung und -wartung – GUST e-Foundry

- Math Symbols Subsets,
- Realisierung der Math Symbols Subsets,

- Erweiterung der T_EXGyre Text Fonts,
- Fontwartung/Erweiterung bestehender Fonts.
- Fördersumme insgesamt 14.000 EUR (über drei Jahre)
- Die Abschlusspräsentation erfolgte auf der Tagung von DANTE e.V. im Frühjahr,
- der schriftliche Bericht erschien in »Die T_EXnische Komödie« Ausgabe 2/2021.

TOP 3: Wahl eines/r neuen verantwortlichen Redakteurs/-in für „Die TEXnische Komödie“

Luzia Dietsche ist seit DTK 1/2021 kommissarisch verantwortliche Redakteurin, gemäß Redaktionsstatut ist die Wahl durch die Mitglieder notwendig für die offizielle Ernennung.

Die Wahl eines/r neuen verantwortlichen Redakteurs/-in wird über die Software „Vereins-Abstimmung.de“ in öffentlicher Abstimmung durchgeführt. Martin Sievers öffnet die Abstimmungsseite mit den Optionen: Luzia Dietsche, Enthaltung, keine(r) der Genannten.

An der Wahl nehmen 32 Mitglieder per Abstimmungssoftware und zwei per Handzeichen teil. Für Luzia Dietsche stimmen 33 Mitglieder (97,1%), bei einer Enthaltung und keiner Gegenstimme. Damit ist sie zur verantwortlichen Redakteurin gewählt. Sie erklärt, dass sie die Wahl annimmt.

TOP 4: Verschiedenes

- Tagungsorte: Wie schon zu Beginn der Mitgliederversammlung erwähnt, sind wir im Herbst 2022 in Saarbrücken eingeladen. Für das Frühjahr 2022 und 2023 sucht DANTE e.V. Tagungsorte, wobei klar ist, dass wir gern die ausgefallene Tagung in Lübeck und die Online-Tagung im Frühjahr '21 in Magdeburg in Präsenz abhalten würden. Allerdings scheint es noch zu früh, einen endgültigen Tagungsort für den März oder April '22 zu benennen, deshalb sind Vorschläge und Angebote zur Ausrichtung willkommen.
- DANTE e.V. vor Ort: Es besteht die Hoffnung, im nächsten Jahr wieder bei Open-Source-Tagen an Schulen oder Universitäten, bei Messen, L^AT_EX-Fortbildungen, -Seminaren oder -Vorlesungen DANTE e.V. zu repräsentieren. Dafür können frühzeitig Materialien und Bücher vom Verein angefordert werden. DANTE e.V. ist dankbar, wenn sich Mitglieder bereit erklären, und unterstützt und fördert diese Events gerne.
- Frage von Marei Peischl: DANTE e.V. schreibt auf seiner Webseite, dass sämtliche Äußerungen auf „Social Media“-Plattformen Privatmeinungen sind. Dann

sollte auch gekennzeichnet werden, ob es sich um die Stellungnahme des Vereins handelt oder um eine private Meinung.

Martin Sievers erwidert, dass dies zu bejahen sei, es beträfe im Prinzip nur noch den Twitter-Account, wo er und Patrick Gundlach etwas schreiben, da die Facebook-Seite nicht mehr gepflegt wird und auf GitHub keine Meinungsäußerungen stattfinden. Martin wird sich dazu eine Vorgehensweise überlegen.

Doris Behrendt wendet ein, dass der Hinweis auf der Überlegung fußt, dass damit die DSGVO-Erklärung keine Hinweise zum Verein enthalten muss. Dies sei vornehmlich eine Arbeitersparnis bei der sonst notwendigen juristisch korrekten Ausarbeitung der Erklärung auf der Webseite von DANTE e.V. Wenn sich jemand fände, der diese Aufgabe übernimmt, könnte das entsprechend angepasst und geändert werden. Auf direkte Ansprache erklärt sich Marei bereit, den Sachverhalt zu untersuchen und dazu eine Erklärung abzugeben.

Martin Sievers schließt die Mitgliederversammlung um 10:37 Uhr.

Martin Sievers
(Versammlungsleiter)

Volker RW Schaa
(Protokollant)

Bericht zur Herbsttagung von DANTE e.V. 2021 in Neuland

Marcel Kapfer

Die Vorgeschichte

Es muss etwa Mitte August gewesen sein, als ein Gedanke in meinem Kopf anklopfte und mich daran erinnerte, dass bald mal wieder eine Tagung von DANTE e.V. bevorstehen sollte. Nach meinem ersten Besuch einer Tagung im Frühjahr 2017 in Zeuthen hatte ich leider nie die Zeit gefunden, zu einer weiteren zu fahren. Nachdem mir die virtuelle Frühjahrstagung recht gut gefiel, dachte ich mir, dass es mal wieder schön wäre, bei einer Tagung vor Ort vorbeizuschauen. Und Zeit hatte ich diesmal sogar.

Doch Ende August wurde den munteren Reiseplänen ein abruptes Ende gesetzt, als Martin Sievers über die vereinsinterne Mailingliste mitteilte, dass die Herbsttagung leider wieder virtuell stattfinden wird. Nichtsdestotrotz habe ich mich Anfang September angemeldet. Im Nachhinein vielleicht ein Fehler, wenn man bedenkt, dass ich nun diesen Bericht schreiben »darf«.

Als dann wenige Tage vor Beginn die notwendigen Vorabinformationen durch das Organisationsteam kamen, war die Vorfreude schon groß. Dies konnten auch gewisse technische Probleme mit der Online-Lernplattform des Saarlands, welche für die Tagung verwendet wurde, nicht ändern (insgesamt fünf E-Mails zur Kennwortänderung benötigte ich, bis es dann endlich klappte).

Der Vormittag

Mit einem frischen Kaffee bewaffnet, trat ich dann am Samstag, den 18. September, um 9:00 Uhr dem virtuellen Saal zur Begrüßung bei. Dort erwartete uns schon Thomas H. Meyer, der lokale Organisator, und unser Vorsitzender, Martin Sievers, welche uns herzlich willkommen hießen.

Nach den einleitenden Worten ging es sofort weiter mit der 63. Mitgliederversammlung. Dabei berichtete der Vorstand über die neuesten Entwicklungen im Verein. Im Anschluss an diesen Bericht wurde Luzia Dietsche offiziell als neue verantwortliche Redakteurin der Mitgliederzeitung gewählt. Abschließend wurde noch eine Diskussion über die Handhabung der *social media accounts* von DANTE e.V. begonnen, welche damit beendet wurde, dass man sich einigte, das Thema weiter zu verfolgen. Zum Ende der Mitgliederversammlung um 20 vor 11 waren dann noch 42 Teilnehmende anwesend.

Da der nächste Vortrag erst um 11:00 Uhr stattfand, blieb uns die Möglichkeit, in verschiedenen »Räumen« über diverse Themen zu reden. So fanden nicht nur nette Gespräche im eigentlichen Konferenzraum, sondern auch in der »Lästerstube« und dem DTK-Redaktionszimmer statt.

Alexander König begrüßte uns nach dieser kurzen Unterbrechung zu seinem Vortrag »Eine saarländische Lehr-Lernplattform macht sich auf den Weg«. Darin erklärte er, wie sich unter seiner Leitung über die letzten 14 Jahre in vielen einzelnen Schritten eine komplette Plattform entwickelt hat. Die neueste Iteration wurde zu Beginn der Corona-Beschränkungen im März letzten Jahres ausgerollt. Dabei wurden die Ressourcen immer weiter aufgestockt: So war die Plattform am Anfang für 5 000 Nutzende ausgelegt und im Dezember 2020 waren dann Kapazitäten für insgesamt 80 000 Nutzende vorhanden. Die Dienste, die in Teilen auch für die Tagung von DANTE e.V. verwendet wurden, beinhalten dabei eine Lernplattform, eine Videokonferenzsoftware, eine Cloud, einen E-Mail-Dienst und ein E-Portfolio. Seine Vortragsfolien veröffentlichte Alexander König auf seiner Webseite: <https://alexander-koenig.info/2021/09/18/dante-tagung-2021/>.

Der Nachmittag

Nach einem schnellen Mittagessen begrüßte uns Martin Sievers zurück zu seinem Vortrag »Der wissenschaftliche Textsatz mit L^AT_EX in den Digital Humanities – eine persönliche Bilanz der letzten elf Jahre«. Darin zeigt er auf, mit und an welchen Werkzeugen für wissenschaftlichen Textsatz er in den letzten elf Jahren arbeitete und vor allem auch, welche Probleme dabei aufkamen. Sowohl von Seiten des Projektmanagements als auch von den Anforderungen an das Ausgabeformat. Als Fazit nennt er unter anderem, dass L^AT_EX ein mächtiges Werkzeug für gedruckte wissenschaftliche Publikationen bleibt.

Im Anschluss hielt Ulrike Fischer ihren Vortrag »Color, xcolor und l3color: Farben mit L^AT_EX«. Darin stellte sie vor, welche Möglichkeiten es gibt, um in L^AT_EX mit Farben zu arbeiten. Begonnen mit dem color-Paket von 1994 über xcolor von 2003 bis hin zum neuen l3color. Letzteres soll vor allem fehlende Funktionen der älteren Pakete beinhalten, wie zum Beispiel das Definieren von Spotfarben. l3color soll früher oder später das Standard-Color-Paket werden, bis dahin sind allerdings noch einige To-dos offen, wie zum Beispiel globale Farben.

Nach einer kurzen Pause, in welcher sich wieder nette Gespräche in den Nebenzimmern entwickelten und man sich gemütlich einen weiteren Kaffee holen konnte, ging es weiter mit dem Vortrag »Futura PM: Eine geometrisch konstruierte Schriftart für (L^A)T_EX« von Philipp Conrad Müller. Dabei stellte dieser sein Projekt vor, die Futura-Schriftart mithilfe von Metatype1 parametrisch zu konstruieren. Dies eignet sich vor allem deshalb, weil die Schrift, welche ursprünglich von Paul Renner gestaltet wurde, hauptsächlich aus den geometrischen Grundformen Kreis, Dreieck und Rechteck besteht. Ziel dabei ist unter anderem auch, einen Mathesatz für Futura zu haben, was aktuell keine andere Futura-Variante beinhaltet. Neben der Fertigstellung von diesem möchte er unter anderem auch ein L^AT_EX-Paket veröffentlichen.

Den Abschluss des Vortragsprogramms machte Uwe Ziegenhagen mit dem Thema »Pocketmods mit L^AT_EX und Python«. Ein Pocketmod (siehe auch <https://pocketmod.com/howto>) ist dabei ein Stück Papier, welches so gefaltet wird, dass zum Beispiel aus einer DIN-A4-Seite ein DIN-A6-Dokument entsteht. Dies lässt sich unter anderem dafür nutzen, um einen Kalender oder eine To-do-Liste in einem handlichen Format stets bei sich zu haben. Uwes Ziel ist dabei, die Inhalte programmatisch mithilfe der Python-Programmiersprache zu erstellen. Als Beispiel zeigt er ein Python-Programm, welche einen Kalender-Pocketmod erstellt und dafür auf die wichtigen Einträge in seinem Google-Kalender (wie »Duschen« oder »Müll rausbringen«) zugreift. Weitere Informationen und der Code zu seinem Projekt lassen sich auf seiner Webseite finden: <https://github.com/UweZiegenhagen/TalksAndArticles/tree/master/2021-Dante-Herbst>

Das Ende

Um kurz nach 16.30 Uhr verabschiedeten sich Martin Sievers und Thomas H. Meyer bei allen Teilnehmenden. Uwe Ziegenhagen wies an der Stelle darauf hin, dass die Videos der Vorträge auch auf dem YouTube-Kanal von DANTE e.V. zur Verfügung gestellt werden. Die Links hierfür finden sich auf der Webseite von DANTE e.V.: <https://www.dante.de/veranstaltungen/herbst2021/programm/>.

Damit war allerdings nur der offizielle Teil vorbei: Einige kamen danach noch auf ein »Telebier« in einen der Konferenzräume. An dieser Stelle ist mir der Fehler passiert, dem falschen Raum beizutreten: Im DTK-Redaktionszimmer kam man dann recht schnell dahinter, dass diese Tagung zwar nicht meine erste war, ich allerdings noch nie einen Bericht schreiben durfte. Nachdem noch niemand dafür gefunden wurde (vielleicht auch, weil Martin Sievers mit den Worten »Ich habe auch so angefangen« eher Interessierte abschreckte als überzeugte) wurde mir daraufhin diese Aufgabe freudig aufgetragen.

Für mich war die Tagung schließlich etwa eine Stunde später leider recht abrupt vorbei, als das Videokonferenzsystem weder mein Mikrofon noch meine Webcam mehr erkennen wollte und auch ein Neuladen nicht weiterhalf.

Abschließend möchte ich mich bei dem lokalen Organisationsteam und dem Vorstand für die tolle Tagung bedanken! Ich hoffe, dass die nächsten wieder vor Ort sein werden und ich die Zeit dafür finde.

Bretter, die die Welt bedeuten

Ganz- und doppelseitige Gleitumgebungen

Herbert Voß

Das Paket `hvfloate` definiert ein Makro, das Objekte und Beschriftungen von Gleitumgebungen an verschiedenen Positionen mit unterschiedlichen Drehwinkeln für Objekt und Beschriftung platziert. Das Objekt kann eine komplette Spalte, Seite oder eine Doppelseite füllen, mit oder ohne Berücksichtigung von Rändern.

Einführung

Die bekannten Gleitumgebungen (floating environments), wie `figure` and `table`, sind einfach zu handhaben, wenn es in der Umgebung nur ein Objekt und eine Beschriftung gibt, die in das aktuelle Seitenlayout passen. Wenn man neben dem Objekt (Bild, Tabelle, ...) beispielsweise eine gedrehte Bildunterschrift wünscht, dann benötigt man etwas L^AT_EX-Kenntnisse oder ein Paket, welches die Drehung und die Überprüfung der aktuellen Seitenzahl übernimmt, falls die gedrehte Bildunterschrift beispielsweise in einem zweispaltigen Layout immer am äußeren Rand erscheinen soll. [2] All dies kann durch die Verwendung des Pakets `hvfloate` vereinfacht werden. Das Paket bietet eine Vielzahl von möglichen Optionen für das Gleitobjekt und die Beschriftung. Es wird wie gewohnt geladen:

```
\usepackage[<options>]{hvfloate}
```

Es gibt die Paketoptionen `hyperref`, `nostfloats` und `fbox`. Letztere wird nur verwendet, um einige Abstandsprobleme im Dokument zu lokalisieren; Objekt und Bildbeschriftungen werden umrahmt und unerwünschter Leerraum ist damit leicht zu erkennen. Mit `nostfloats` kann man das Laden des Pakets `stfloats` verhindern. Allerdings sind dann »Bottom«-Floats in einem zweispaltigen Dokument nicht mehr möglich. Diese Option sollte nur in einigen seltenen Fällen verwendet werden, in denen ein Paketkonflikt zwischen `stfloats` und einem anderen Paket besteht. Mit `hyperref` wird das gleichnamige Paket geladen.

Möchte man eine Voreinstellung für die Positionsparameter von Gleitumgebungen haben, die von L^AT_EX standardmäßig auf `tbp` gesetzt sind (*here*, *top* und *bottom*), so

kann man das Hilfspaket `hvfloa-fps` laden. Es kennt die Optionen `table`, `figure` und `all`, denen man entsprechende Positionsparameter zuweisen kann. Hat man ein Dokument mit einer großen Anzahl von Gleitumgebungen, aber relativ wenig Text, sollte man das Paket wie folgt laden:

```
\usepackage[all=!htb]{hvfloa-fps}
```

Das Ausrufezeichen erlaubt L^AT_EX, die internen Parametereinstellungen für die Gleitumgebungen zu ignorieren, z. B. die Maximalzahl der Gleitumgebungen auf einer Seite, Abstand zum normalen Text usw. [4, 10]

Abhängigkeiten

Folgende Pakete werden von `hvfloa` standardmäßig geladen: `caption`, `subcaption`, `picture`, `trimclip`, `expl3`, `multido`, `graphicx`, `xkeyval`, `stfloats`, `ifoddpage` und `afterpage`.

Möchte man eines dieser Pakete mit bestimmten Optionen laden, so empfiehlt sich zur Vermeidung des sogenannten »option clash« folgende Vorgehensweise, die hier beispielhaft für `caption` gezeigt wird:

```
\PassOptionsToPackage{margin=10pt,font=small,labelfont=bf}{caption}
\documentclass[...]{...}
\usepackage{hvfloa}
...
```

Die Makros und optionale Argumente

Die Syntax der wichtigsten drei Makros `\hvFloat`, `\hvFloatSetDefaults` und `\hvFloatSet` ist teilweise komplex. Optionale Argumente sind grau hinterlegt:

```
\hvFloatSetDefaults
\hvFloatSet{key=value list}
\hvFloat * [Options] + {float type}{floating object}%
           [short caption] {long caption}{label}
```

Die Sternvariante von `\hvFloat` wird im Abschnitt auf Seite 22 und das optionale Argument `+` im Abschnitt auf Seite 37 erklärt.

`\hvFloatSet` erlaubt das globale Setzen von Schlüsselwörtern, wohingegen `\hvFloatSetDefaults` alle Schlüsselwörter auf ihre, wie in Tabelle 2 auf Seite 19 gezeigten, Standardwerte zurücksetzt.

Hat `\hvFloat` einen leeren zweiten Parameter `<float type>`, dann wechselt `\hvFloat` automatisch zu einem nicht-gleitenden Objekt und setzt zusätzlich den Parameter

onlyText=true (siehe Tabelle 2 auf Seite 19). Der Parameter für die Beschriftung wird dann als reiner Text interpretiert. Ein Kurzeintrag bei der Beschriftung ist immer für eine Liste der Abbildungen (`\listoffigures`) oder Liste der Tabellen (`\listoftables`) gedacht. Alle anderen Parameter sind verpflichtend, selbst wenn sie als leeres Argument angegeben werden. `hvfloat` definiert intern weitere Makros, die man auch für eigene Bedürfnisse nutzen kann, wobei `\tabcaption` grundsätzlich die Tabellenbeschriftung oberhalb vorsieht:

```
\figcaption [short caption] {caption text}
\tabcaption [short caption] {caption text}
\tabcaptionbelow [short caption] {caption text}
```

Diese drei Makros werden für nicht-gleitende Umgebungen genutzt (Option `nonFloat`), bei denen der normale Befehl `\caption` nicht erlaubt ist. Dagegen kann beispielsweise `\tabcaption` an jeder beliebigen Stelle im Dokument benutzt werden, selbst ohne ein entsprechendes Tabellenobjekt:

Tab. 1: Eine Tabellenbeschriftung ohne wirklichen Sinn und ohne Tabelle.

Ein eventuelles Label muss allerdings *innerhalb* des Arguments `\tabcaption` gesetzt werden, damit eine Referenz auch auf die richtige Tabellennummer verweist, in diesem Beispiel auf Tabelle 1.

```
[...]
```

```
%
```

```
\tabcaption[Eine sinnlose Beschriftung \ldots]% für LOT
```

```
{Eine Tabellenbeschriftung ohne wirklichen Sinn und ohne Tabelle.\label{dummy}}
```

```
%
```

```
Ein eventuelles Label muss allerdings \emph{innerhalb} des Arguments
```

```
\Lcs{\tabcaption} gesetzt werden, damit eine Referenz auch auf die richtige
```

```
Tabellennummer verweist, in diesem Beispiel auf Tabelle~\vref{dummy}.
```

Mit dem Makro `\hvDefFloatStyle` kann man eigene Stile definieren, die intern unter dem Namen `\hv@<name>` gespeichert werden. Dies erspart wiederholtes Setzen von Schlüsselwörtern, da dann ein einfaches `style=meinStil` reicht.

```
\hvDefFloatStyle{<Name>}{<Settings>}
```

Alle möglichen Parameter sind in der ganzseitigen und im Querformat dargestellten Tabelle 2 auf Seite 19 aufgeführt. Diese Tabelle wurde zuerst in der definierten Box `hvOBox` als eine `tabularx` mit der Texthöhe als Tabellenbreite gespeichert. Die Um-

gebung tabularx kann selbst nicht als Argument von `\hvFloat` auftreten, weshalb hier der Weg über die interne Box `\hvOBox` gegangen werden muss:

```
\begin{lrbox}{\hvOBox}\small
\begin{tabularx}{\textheight}{@{} l>{\small\ttfamily}c X @{} }\toprule
[...]\end{tabularx}
\end{lrbox}
```

Die Tabelle ist zusammen mit der Beschriftung rotiert worden, was durch Festlegung von `rotAngle` möglich ist:

```
\hvFloat*[floatPos=p,rotAngle=90,capPos=top,capWidth=w,useOBox=true]%
{table}% Typ festlegen
{}% keine Angabe des Objekts, es wird \hvOBox benutzt
{Die möglichen optionalen Schlüsselwörter für das Makro \Lcs{hvFloat}.}%
{tab:options}
```

Erscheint im Folgenden eine Beschriftung einer Abbildung oder Tabelle zu reinen Demonstrationszwecken, so wird diese in roter Farbe gesetzt, um sie deutlich vom normalen Text abzuheben.

Die Position der Beschriftung

Standardmäßig wird die Beschriftung unterhalb (Abbildung) oder oberhalb (Tabelle) des Objekts gesetzt und das Makro `\hvFloat` verhält sich wie die bekannte Umgebung `figure` oder `table`. Mit dem Schlüsselwort `capPos` und dem Wert `before`, welcher bei kleinen Objekten (schmäler als eine Spalte oder Seite) identisch zu `left` ist, kann die Beschriftung neben dem Objekt platziert werden:

```
\hvFloat[capPos=left]{figure}{%
\includegraphics{monument0}}{Eine seitliche Beschriftung neben einer Abbildung,
die kein Label bekommt, weshalb das Argument dafür leer bleibt.}{}
```

Abb. 1: Eine seitliche Beschriftung neben einer Abbildung, die kein Label bekommt, weshalb das Argument dafür leer bleibt.



Ist die Bildunterschrift kürzer als die mögliche Breite, wird sie automatisch horizontal zentriert. Das Objekt wird dagegen standardmäßig rechtsbündig gesetzt, was für kurze Beschriftungen ungünstig sein kann. In diesen Fällen sollte die Beschriftung unterhalb des Objekts gesetzt werden.

Tab. 2: Die möglichen optionalen Schlüsselwörter für das Makro `\hvFloat`.

Keyword	Default	Description
floatPos	tbp	Dies ist die gleiche Standardplatzierungseinstellung wie bei Standard-L ^A T _E X, aber vielleicht nicht immer die beste Einstellung.
rotAngle	0	Der Wert für den Winkel, wenn sowohl das Objekt als auch die Bildunter-/überschrift zusammen gedreht werden sollen.
capwidth	n	Die Breite der Beschriftung. Kann »n« sein, wie eine natürliche Breite (gegeben durch die aktuelle Zeilenbreite), »w« wie die Breite des Objekts, »h« wie die Höhe des Objekts oder ein Skalierungswert für <code>\columnwidth</code> .
capAngle	0	Der ganzzahlige Wert für den Winkel, wenn die Beschriftung gedreht werden soll. Zählung gegen den Uhrzeigersinn.
capPos	bottom	Die Position der Beschriftung relativ zum Objekt. Mögliche Werte sind: before: immer vor vom Objekt; top: immer über dem Objekt; left: im <code>\twocolumn</code> -Modus immer vor vom Objekt und auf derselben Seite; after: immer nach dem Objekt; bottom: immer unterhalb des Objekts; right: immer nach dem Objekt und im <code>\twocolumn</code> -Modus auf derselben Seite; inner: Im <code>\twoside</code> -Modus immer am inneren Rand; outer: Im <code>\twoside</code> -Modus immer am äußeren Rand; evenPage: Im <code>\twoside</code> -Modus und ganzseitigen Objekten immer auf einer geraden (linken) Seite; oddPage: Im <code>\twoside</code> -Modus und ganzseitigen Objekten immer auf einer ungeraden (rechten) Seite.
capPos	bottom	Dies ist nur für <code>capPos=left right</code> wichtig. Dann kann die Beschriftung vertikal bottom, center oder top sein.
objectPos	center	Die horizontale Platzierung des Objekts relativ zum Dokument. Mögliche Werte sind <code>(l eft)(c enter)(r ight)</code> .
objectAngle	0	Der ganzzahlige Wert für den Winkel, um den das Objekt gedreht werden soll (gegen den Uhrzeigersinn).
floatCapSep	5pt	Der zusätzliche Abstand zwischen dem Objekt und einer links oder rechts platzierten Beschriftung.
useOBox	false	Anstatt das Objekt als Parameter zu übergeben, soll der Inhalt der Box <code>\hvOBox</code> benutzt werden.
onlyText	false	Die Beschriftung wird als normaler Text gedruckt und es gibt keinen Eintrag in der Liste der ...
nonFloat	false	Das Objekt wird in eine nicht-gleitende Umgebung gesetzt. Der entsprechende Zähler wird wie gewohnt erhöht und kann referenziert werden.
wide	false	Die Gleitumgebung kann <code>\textwidth+\marginparwidth</code> als horizontale Breite verwenden.
objectFrame	false	Setzen eines Rahmens ohne Abstand um das Objekt.
style	---	Verwendung des angegebenen Stils.
capFormat	---	Formatierungsoptionen für das Paket <code>\caption</code> (siehe Dokumentation des Pakets).
subcapFormat	---	Formatierungsoptionen für das Paket <code>\subcaption</code> (siehe Dokumentation des Pakets).
fullPage	false	Vollständigen Textbereich der Seite/Spalte für das Objekt nutzen.
FULLPAGE	false	Vollständiges Papierformat einer Seite für das Objekt nutzen.
FullPage	false	Vollständigen Textbereich für das Objekt nutzen (Seite oder beide Spalten).
doublePage	false	Benutze Textbereich einer Doppelseite mit zusätzlichem Text.
doublePAGE	false	Benutze Textbereich einer Doppelseite ohne zusätzlichen Text.
doubleFULLPAGE	false	Benutze gesamte Papierbreite und -höhe einer Doppelseite ohne zusätzlichen Text.
sameHeight	false	Bei doppelseitigen Objekten auf beiden Seiten gleiche Texthöhen.
vfill	false	Füge ein <code>\vfill</code> zwischen Teilobjekten bei <code>multifloat</code> oder <code>subfloat</code> ein.

Die vertikale Position ist standardmäßig ebenfalls zentriert. Dies kann durch das optionale Schlüsselwort `capVPos` geändert werden. Die Formatierung kann durch die optionalen Schlüsselwörter des standardmäßig geladenen Pakets `caption` geändert werden. Sie können von `\hvFloat` über das Schlüsselwort `capFormat` an `caption` übergeben werden (siehe Abbildung 2). Die Bildunterschrift wird individuell gedreht, indem `capAngle=90` gesetzt wird, was einer Drehung gegen den Uhrzeigersinn entspricht:

```
\hvFloat[capPos=right, capAngle=90, capWidth=h, capFormat={font=sf}]%
{figure}{\includegraphics[width=0.5\columnwidth]{beate}}%
{Eine etwas längere Bildunterschrift rechts neben dem Objekt in der Breite
der Höhe und zusätzlich um 90° gedreht.}{fig:1}
```



Abb. 2: Eine etwas längere Bildunterschrift rechts neben dem Objekt in der Breite der Höhe und zusätzlich um 90° gedreht.

Die vertikale Position wird durch das Schlüsselwort `capVPos` gesteuert, das die Werte `bottom`, `center` oder `top` akzeptiert. Aus typografischer Sicht sollte eine seitliche Beschriftung bei Abbildungen unten und bei Tabellen oben ausgerichtet werden.

```
\hvFloat[capPos=inner, capVPos=bottom, objectAngle=180]{figure}%
{\includegraphics[width=0.45\columnwidth]{usa3}}%
{Eine längere Beschriftung am Innenrand des zweiseitigen Dokuments und vertikal
am unteren Rand des Objekts, welches selbst um 180° gedreht wurde.}{fig:11}
```



Abb. 3: Eine längere Beschriftung am Innenrand des zweiseitigen Dokuments und vertikal am unteren Rand des Objekts, welches selbst um 180° gedreht wurde.

Breite der Beschriftung

Bei einer Bildunterschrift neben dem Objekt ist die horizontale Ausrichtung standardmäßig zentriert, wenn sowohl das Objekt als auch die Bildunterschrift kleiner als die aktuelle Spalten-/Zeilenbreite sind. Die Breite der Beschriftung selbst kann durch das Schlüsselwort `capWidth` gesteuert werden, das auf `n` (natural width), `w` (width of the object), `h` (height of the object) oder einen Faktor für `\columnwidth` gesetzt werden kann. Abbildung 2 auf der vorherigen Seite zeigt die Verwendung von `capWidth=h`, was für die gedrehte Beschriftung neben dem Objekt verwendet wird und Abbildungen 4 sowie 5 zeigen eine Beschriftung ober- und unterhalb und mit der gleichen Breite wie das Objekt.

```
\hvFloat[capWidth=w,capPos=top,objectAngle=90]{figure}%
{\includegraphics{greek0}}%
{Eine Beschriftung über dem um 90° gedrehten Objekt und mit der
gleichen Breite, die der ursprünglichen Höhe des Objekts entspricht.}{fig:1a}
```

Abb. 4: Eine Beschriftung über dem um 90° gedrehten Objekt und mit der gleichen Breite, die der ursprünglichen Höhe des Objekts entspricht.



Abb. 5: Eine Beschriftung unter dem Objekt und mit der gleichen Breite des Objekts.

Im Folgenden wird aus reinen Demonstrationsgründen der Text zweispaltig gesetzt, wozu das Makro `\twocolumn` ausgeführt wird. Das Paket `hvfloa` kann zwar mit dem Paket `multicol` benutzt werden, welches auch mehrspaltige Umgebungen erlaubt, bei spalten- oder seitenübergreifender Anwendung des Makros `\hvFloat` ist dann jedoch mit falschen Ergebnissen zu rechnen.



Abb. 6: Eine längere Beschriftung direkt neben dem Objekt und vertikal am unteren Rand des Objekts. Sie erstreckt sich über beide Spalten und befindet sich oben oder unten bezogen auf das Objekt.

Tab. 3: Zusätzliche Schlüsselwörter für das Makro `\includegraphics` aus dem Paket `graphicx`.

<i>name</i>	<i>width=</i>	<i>height=</i>	<i>keepaspectratio=</i>
<code>fullpage</code>	<code>\columnwidth</code>	<code>\textheight</code>	<code>false</code>
<code>FullPage</code>	<code>\textwidth</code>	<code>\textheight</code>	<code>false</code>
<code>FULLPAGE</code>	<code>\paperwidth</code>	<code>\paperheight</code>	<code>false</code>
<code>doubleFULLPAGE</code>	<code>2\paperwidth</code>	<code>\paperheight</code>	<code>false</code>
<code>doubleFULLPAGEbindCorr</code>	<code>2\paperwidth-2\bindCorrection</code>	<code>\paperheight</code>	<code>false</code>

Die Sternversion `\hvFloat*`

Im Zweispaltenmodus (`\twocolumn`) kann mit der Sternversion `\hvFloat*` die Gleitumgebung über beide Spalten gesetzt werden. Dies ist ähnlich zu den Umgebungen `table*` und `figure*`. Die Gleitumgebung wird, wenn möglich, nur oben auf der folgenden oder unten auf der aktuellen Seite angezeigt. Letzteres benötigt das Paket `stfloats`, das standardmäßig von `hvfloat` geladen wird. Das Platzieren der Gleitumgebung über beide Spalten *innerhalb* des Textbereiches ist nicht möglich. Der Code für Abbildung 6 lautet:

```
\hvFloat*[capFormat={font=sf},
    capPos=right,capVPos=bottom]%
{figure}%
```

```
{\includegraphics{greek1}}%
{Eine längere Beschriftung direkt
neben dem Objekt und vertikal am
unteren Rand des Objekts. Sie
erstreckt sich über beide Spalten
und befindet sich oben oder unten
bezogen auf das Objekt.}%
{fig:2}
```

Das Gleiche ist bei Tabelle 3 ersichtlich, die sich ebenfalls über zwei Spalten erstreckt. Intern wird die Anzahl der möglichen gleitenden Objekte oben auf der Seite durch die Parameter `\topnumber` (Einspaltenmodus) und `\dbltopnumber` (Zweispaltenmodus) gesteuert. Sie sind für die hier genutzte Dokumentenklasse (`dtk`) jeweils auf 2 voreingestellt und können für andere Dokumentenklassen abweichen. Eine Zusammenstellung aller Parameter

für Gleitumgebungen findet man in [4, 10].

Komplette Spalte oder Seite für ein Objekt

Wie bereits in Tabelle 2 auf Seite 19 erwähnt, gibt es drei Schlüsselwörter für ganzseitige Objekte:

- `fullpage` für eine komplette Textspalte oder -seite (einspaltig),
- `FullPage` für einen kompletten Textbereich einer Seite oder beider Spalten im zweispaltigen Modus und
- `FULLPAGE` für den vollständigen Papierbereich.

Diese Angaben beziehen sich auf den reservierten Seitenbereich, den `\hvFloat` beim Setzen des Objekts und der Beschriftung verwendet. Das Objekt selbst kann kleiner als eine ganze Spalte oder Seite sein. Das Paket `hvfloat` definiert fünf zusätzliche Schlüsselwörter für das Paket `graphicx`, die zusammen mit `\includegraphics` verwendet werden können, um den Code im Dokument etwas kürzer zu machen. Sie sind in Tabelle 3 auf der vorherigen Seite aufgeführt. Die sogenannte Bindekorrektur ist ein zusätzlicher Freiraum an den Innenrändern eines zweiseitigen Dokuments. [3]

Der einzige Unterschied zwischen der Nutzung des gesamten Textbereiches oder des gesamten Papierbereiches für das Gleitobjekt ist die Tatsache, dass es bei `FULLPAGE` weder eine Seitenzahl noch eine Kopf- und/oder Fußzeile gibt; der Seitenstil für diese Seite ist leer (`empty`).

Das Setzen von `keepaspectratio` auf `false` (nicht-beachten des Seitenverhältnisses) ist nur für Bilder sinnvoll, die nahezu das gleiche Verhältnis wie die aktuelle Papierbreite zu -höhe aufweisen. Die Verwendung einer vollständigen Spalte oder Seite für ein Objekt impliziert, dass die Beschriftung auf der vorhergehenden oder folgenden Spalte/Seite platziert wird. Bei einem zweispaltigen Dokument sollte dies immer die gegenüberliegende Spalte auf derselben Seite und bei zweiseitigen Dokumenten immer die gegenüberliegende Seite sein. Nur bei doppelseitigen Objekten (linke-rechte Seite) muss die Überschrift auf der vorhergehenden oder folgenden Spalte/Seite stehen, standardmäßig am Ende dieser Seite oder Spalte.

Es werden je nach Komplexität des Dokuments mehrere L^AT_EX-Durchläufe benötigt, bis das Makro weiß, ob es auf einer geraden oder ungeraden Seite ausgeführt wurde und daraus folgend alle Referenzen richtig sind. Die Logdatei sollte melden, dass sich Labels möglicherweise geändert haben, wenn ein weiterer Kompilierungslauf erforderlich ist.

Ein definiertes Label im Makro `\hvFloat` zeigt immer auf das Objekt und *nicht* auf die Beschriftung, was für die normalen Gleitumgebungen keinen Unterschied macht: Objekt und Beschriftung sind auf derselben Seite. Bei ganzseitigen oder doppelseitigen Objekten definiert das Makro intern zwei zusätzliche Marken (Label), die zum einen auf die Beschriftung (Marke `<label>-cap`) und zum anderen auf den zweiten (rechten) Teil des Objekts

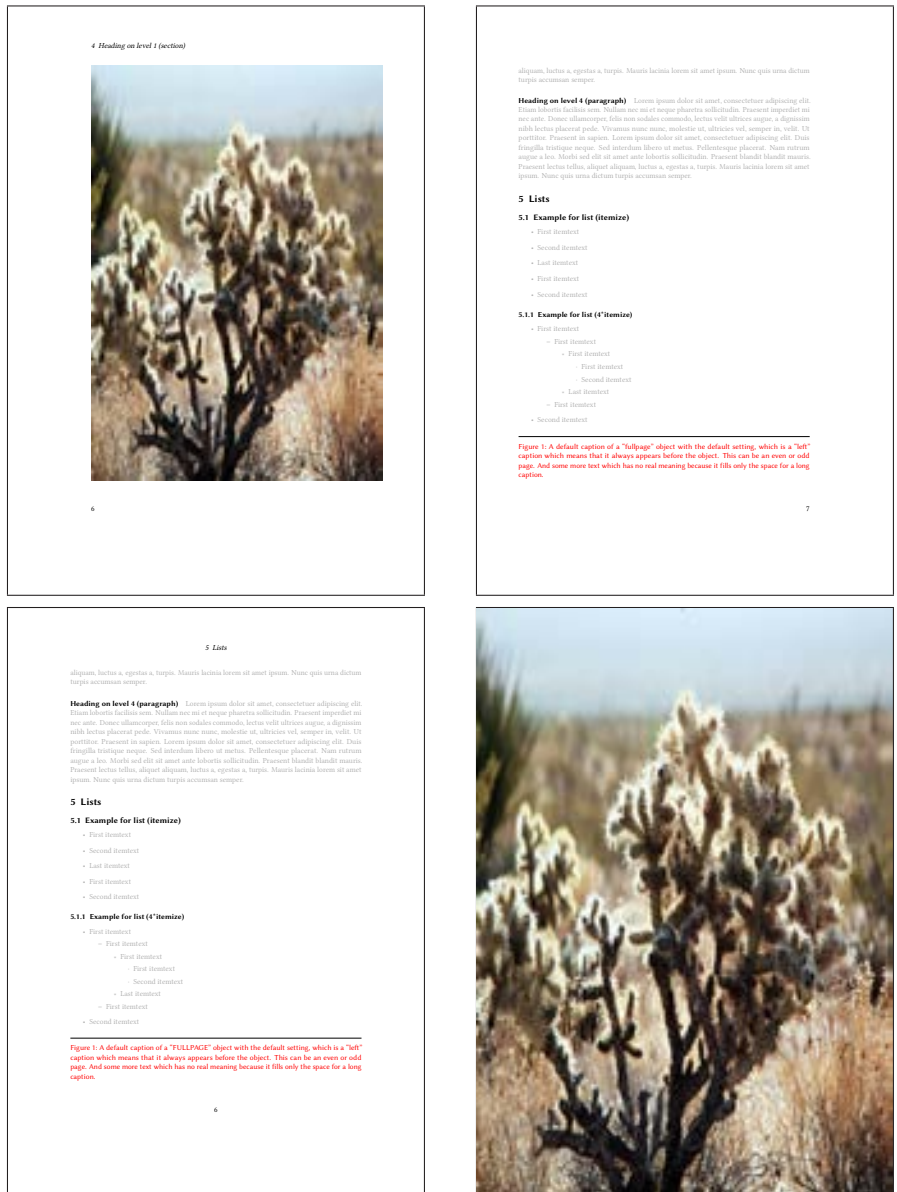


Abb. 7: Die Seiten 6 (links) und 7 (rechts) der Beispieldokumente `odd2s1c.tex` (Option `fullpage`) und `paper-even2s1c.tex` (Option `FULLPAGE`).

zeigen, wenn es sich um eine Doppelseite handelt (Marke `<label>-2`). Alle Marken, das angegebene `<label>` und die internen `<label>-cap` und `<label>-2`, zeigen auf denselben Objektzähler, aber auf unterschiedliche Seitenzahlen. Ein Beispiel wird im entsprechenden Abschnitt auf Seite 28 gezeigt, wo die Bildunterschrift auf Seite 33 und das Objekt auf den Seiten 32 und 33 zu sehen ist. Die folgende Tabelle zeigt dieses Verhalten für das benutzerdefinierte Label `dP`, das für Abbildung 14 verwendet wird:

	<i>dP</i>	<i>dP-cap</i>	<i>dP-2</i>
<code>\ref{dP}</code>	14	14	14
<code>\pageref{dP}</code>	32	33	33

Zweiseitig und einspaltig

Das ganzseitige Objekt und die entsprechende Beschriftung sollten in einem zweiseitigen Dokument gegenüber (links–rechts) stehen. Dies kann mit dem Schlüsselwort `capPos` und den Werten `evenPage` oder `oddPage` eingestellt werden (Abbildung 7 auf der vorherigen Seite). Um Platz zu sparen, werden nur die Ausgaben von zwei Beispieldokumenten gezeigt, wobei die ersten beiden das Makro mit der folgenden Festlegung verwenden:

```
\hvFloat[fullpage, capPos=evenPage]%
{figure}%
{\includegraphics[fullpage]{froese}}%
{Eine Beschriftung auf der geraden
(linken) Seite eines zweiseitigen
Dokuments. Und noch etwas Text,
der keine wirkliche Bedeutung hat,
weil er nur den Platz für eine
```

```
lange Bildunterschrift ausfüllt.}%
{fig:fullpage1}
```

Die unteren beiden Seiten in Abbildung 7 auf der vorherigen Seite sind die gleichen außer der Festlegung `capPos` to `evenPage`. Die Beschriftung ist in roter Farbe gesetzt, um sie besser sichtbar zu machen. Der vollständige Code des Beispiels ist unter CTAN verfügbar (<https://mirror.ctan.org/macros/latex/contrib/hvfloat/doc/examples>).

Zweiseitig und zweiseitig

Standardmäßig erscheint eine Beschriftung vor dem Ganzseiten- oder Ganzspalten-Objekt, unabhängig davon, ob es eine gerade oder ungerade Seite, beziehungsweise linke oder rechte Spalte ist. Die folgende Abbildung 8 auf der nächsten Seite zeigt die Ausgabe dieses Codes:

```
\hvFloat[fullpage, capPos=inner]%
{figure}%
{\includegraphics[fullpage]{buesum}}%
[Kurzer Titel für die LOF.]%
{Eine Beschriftung auf der
Innenseite eines zweiseitigen
Dokuments. Dies kann eine gerade
oder ungerade Seite sein. Und nun
noch etwas Text, der keine
wirkliche Bedeutung hat, weil er
nur den Platz für eine lange
Bildunterschrift ausfüllt.}%
{fig:full0}
```

Die Beschriftung sollte beim zweiseitigen und zweiseitigen Dokument in der inneren Spalte stehen, die bei einer geraden (linken) Seite die zweite und bei einer ungeraden (rechten) Seite die erste Spalte ist. Bei einem einspaltigen Dokument ist es auch sinnvoll, die Beschriftung auf der



geraden (linken) Seite am zweiten Rand und das Objekt auf der ungeraden (rechten) Seite am ersten Rand zu haben. Dies kann mit der Einstellung `capPos=inner` erreicht werden.

Wie bereits erwähnt, werden mehrere $\LaTeX$ -Durchläufe benötigt, bis das Makro weiß, ob es auf einer geraden oder ungeraden Seite ausgeführt ist. Entsprechende Informationen findet man in der Logdatei.

Man kann mit Problemen rechnen, wenn man die vollständige Spaltennutzung für ein Objekt haben möchte, aber auf der zweiseitigen Seite bereits oben eine Gleitumgebung als Sternvariante erscheint. In einem solchen Fall bleibt es dem Benutzer überlassen, seine Textstruktur zu ändern, um eine solche Situation zu vermeiden. Viele weitere Beispiele findet man auf CTAN (<https://mirror.ctan.org/macros/latex/contrib/hvfloat/doc/examples>) oder im Dokumentenverzeichnis der lokal installierten $\TeX$ -Distribution.

Ein vorgegebenes Label `<name>` zeigt auf das Objekt und das intern definierte Label `<name>-cap` zeigt auf die Beschriftung. Beide erhalten die gleiche Ausgabe für `\ref{<name>}`, können aber eine andere Ausgabe für `\pageref{<name>}` haben.

Abb. 8: Eine Beschriftung auf der Innenseite eines zweiseitigen Dokuments. Dies kann eine gerade oder ungerade Seite sein. Und nun noch etwas Text, der keine wirkliche Bedeutung hat, weil er nur den Platz für eine lange Bildunterschrift ausfüllt.

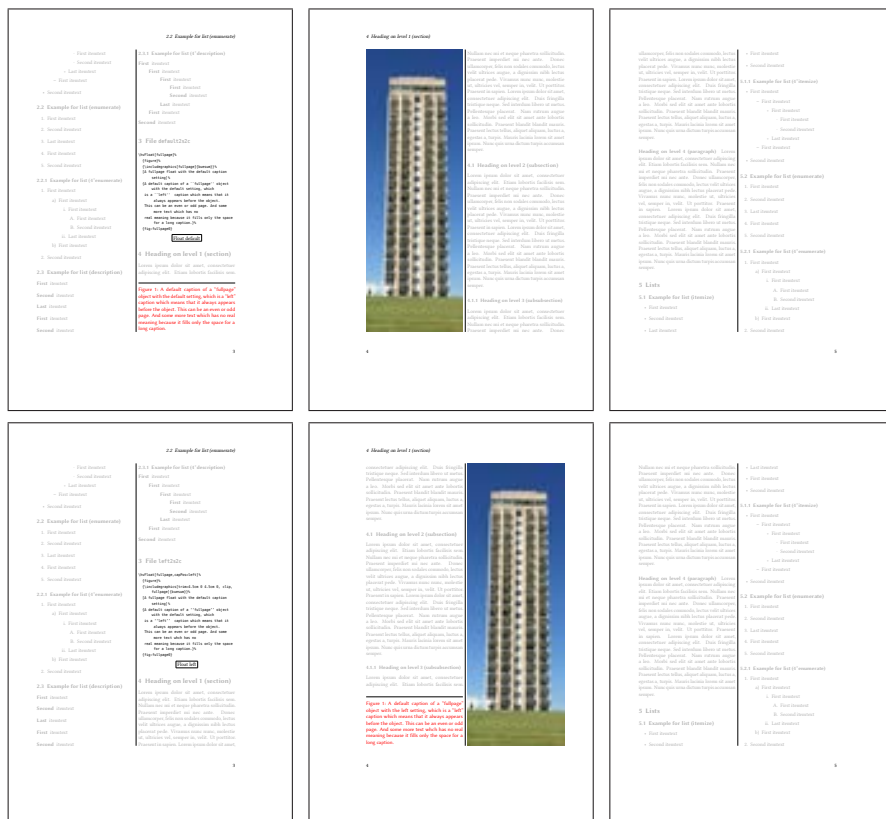


Abb. 9: Die Seiten 3–5 der Beispieldokumente `default2s2c.tex` (`capPos=before` – Standard) und `left2s2c.tex` (`capPos=left`).

Beispielsweise hat das Objekt in Abbildung 14 seine Beschriftung auf Seite 33 dieses Dokuments. Das Objekt selbst erscheint dagegen auf den zwei Seiten 32 und 33.

Beispielsweise hat das Objekt in Abbildung~\ref{dP} seine Beschriftung auf Seite~\pageref{dP-cap} dieses Dokuments. Das Objekt selbst erscheint dagegen auf den zwei

Seiten~\pageref{dP} und ~\pageref{dP-2}.

Im zweiseitigen und zweiseitigen Modus unterscheidet sich die Einstellung `capPos=left` von `capPos=before`. Für `capPos=before` ist es dem Makro egal, auf welcher Seite und in welcher Spalte die Beschriftung erscheinen wird; sie steht lediglich immer *vor* dem Objekt. Bei der Angabe `capPos=left` steht die

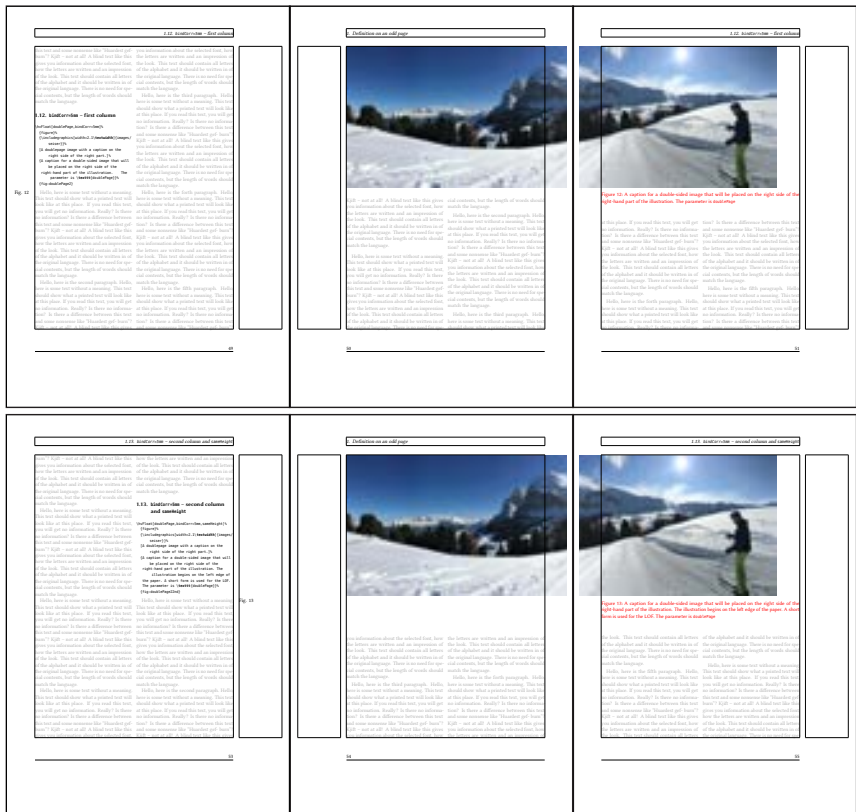


Abb. 10: Beispiel für eine doppelseitige Abbildung mit Bindekorrektur von 5 mm. Die obere Reihe zeigt die Seiten 49–51 und die untere Bildreihe die Seiten 53–56 und die Anwendung von `sameHeight` (Beispieldokument `doublepage2s2c.tex`).

Beschriftung ebenfalls nicht nur vor dem Objekt, sondern immer auch auf *derselben* Seite! Die untere Bildreihe der Abbildung 9 auf der vorherigen Seite zeigt dieses Verhalten. Die Definition des Makros `\hvFloat` steht für beide auf Seite 3 in der zweiten Spalte, mit einem kleinen Rahmen gekennzeichnet.

Doppelseitige Objekte

Doppelseitige Objekte machen prinzipiell nur bei zweiseitigen Dokumenten wirklich Sinn (Dokumentenklassensoption `twoside`). Dann kann das doppelseitige Objekt auf sichtbaren linken–rechten Seiten platziert werden und die Beschriftung vielleicht auf der rechten Seite. Falls

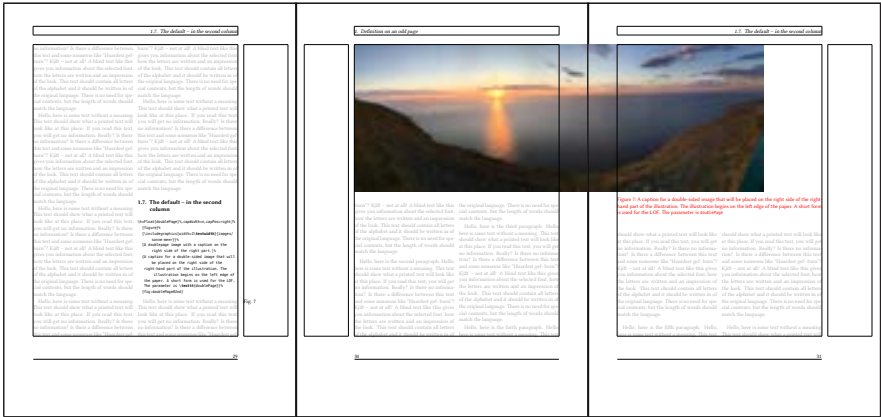


Abb. 11: Die Seiten 29–31 des Beispieldokuments doublePAGE2s2c.tex.

fast die gesamte Papierbreite verwendet wird, kann die Beschriftung auch unter dem rechten Teil des Bildes (wie in Abbildung 10 auf der vorherigen Seite zu sehen) oder auch unten auf der vorhergehenden oder nachfolgenden Seite gesetzt werden.

Ein Beispiel: Das zweiseitige Objekt verwendet die kompletten zwei Seiten (gegeben durch $2 \backslash \text{paperwidth} \times \backslash \text{paperheight}$) und erscheint auf den Seiten 80–81 (links–rechts). Dann kann die Beschriftung unten auf der Seite 79 (davor) oder unten auf der Seite 82 (danach) erscheinen. Abbildung 13 auf der nächsten Seite zeigt ein entsprechendes Verhalten. Es ist auch möglich, die Beschriftung über den rechten Teil des Objekts (Bild) unten zu drucken oder rechts gedreht daneben (siehe Abbildung 12 auf der nächsten Seite).

Nur mit dem Schlüsselwort doublePage kann unter dem zweiseitigen Objekt zusätzlicher Text erscheinen. Vorausgesetzt, dass Objekt und Beschriftung weniger als 90 % der Seite benutzen, anderenfalls erscheint kein weiterer Text. Die anderen beiden Möglichkeiten doublePAGE und doubleFULLPAGE erlauben keinen zusätzlichen Text auf den beiden Seiten, sind aber selbst dennoch Gleitumgebungen. Nachfolgender Text kann daher »vorgeholt« werden, sodass die Seite vor der belegten Doppelseite mit Text gefüllt werden kann.

Option doublePage

Dies ist dasselbe, als ob man zwei verschiedene Gleitumgebungen oben auf der linken und gegenüberliegenden rechten Seite platzieren würde. Das Paket hvfloat schneidet jeweils den Teil des Objekts ab, der für die Seite nicht sichtbar ist. Ist das

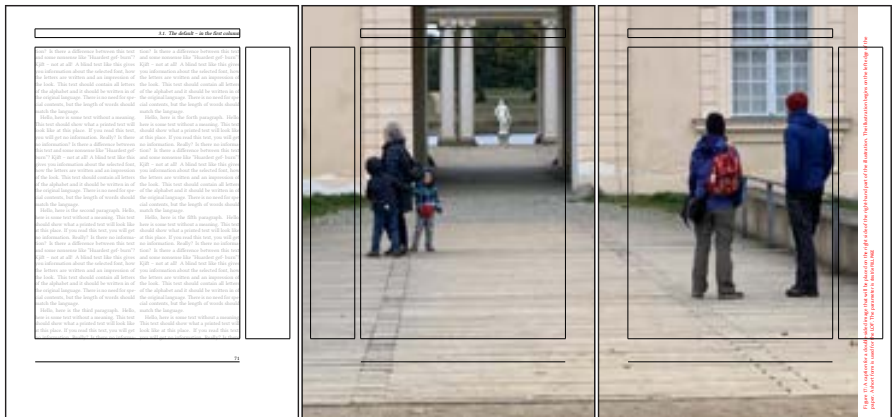


Abb. 12: Die Seiten 71–73 des Beispieldokuments doubleFULLPAGE2s2c.tex, wobei der Parameter `keepaspectratio` auf `true` gesetzt ist.

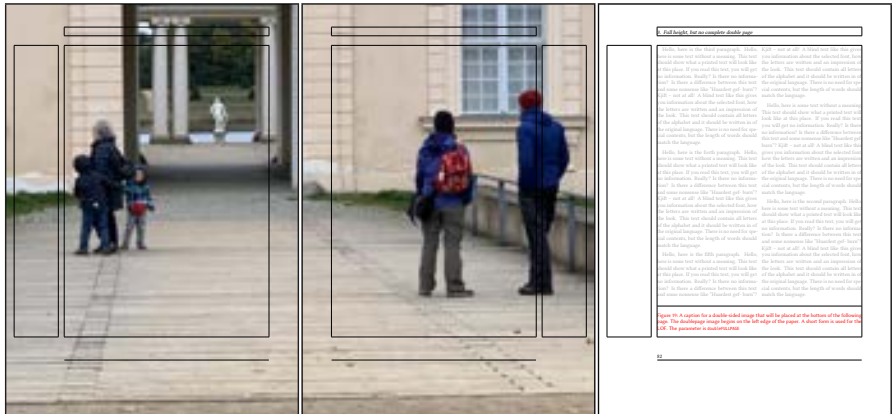


Abb. 13: Die Seiten 80–82 des Beispieldokuments doubleFULLPAGE2s2c.tex. `keepaspectratio` ist auf `false` gesetzt und die Beschriftung ist auf der folgenden Seite (in Rot ausgegeben).

Objekt in der Breite kleiner als die Papierbreite, macht es keinen Sinn, eine doppelseitige Gleitumgebung zu verwenden. Man sollte in diesem Fall auf eine normale Gleitumgebung zurückgreifen.

Für `doublePage` beginnt das Objekt oben links im Textbereich und endet in Abhängigkeit der Breite irgendwo auf der rechten Seite. Die Innenränder des zweiseitigen Dokuments werden hierbei ignoriert, aber eine Bindekorrektur kann eingestellt werden und wird dann berücksichtigt. Die Beschriftung befindet sich immer auf der rechten Seite (gedreht) neben oder unter dem Objekt. Für das Beispiel Abbildung 14 auf der nächsten Seite ist die Beschriftung rechts (`capPos=right`) und um 90° gedreht (`capAngle=90`). Für die folgende Information werden die intern erstellten Labels verwendet, ausgehend vom vorgegebenen Label `dP`.

Der linke Teil des Objekts befindet sich auf Seite 32, der rechte Teil auf Seite 33 und die Beschriftung auf Seite 33.

Der linke Teil des Objekts befindet sich auf Seite~\pageref{dP}, der rechte Teil auf Seite~\pageref{dP-2} und die Beschriftung auf Seite~\pageref{dP-cap}.

Der einzige Unterschied zu `doublePAGE` besteht darin, dass auf den beiden Objektseiten zusätzlicher Text erscheinen kann. Dies ist sinnvoll, wenn das Objekt nur eine geringe Höhe hat.

Mit der Festsetzung `doubleFULLPAGE` startet das Objekt in der linken oberen Papierecke, der eigentliche Textbereich hat in diesem Fall keine Bedeutung. Beschriftungen können wieder rechts daneben, darunter, darüber oder separat auf

eine vorhergehende oder folgende Seite gesetzt werden. Tabelle 3 auf Seite 22 zeigt die Bedeutung der entsprechenden Schlüsselwörter für `\includegraphics`. Das Objekt kann aber auch eine Tabelle oder etwas anderes sein. Abbildung 14 auf der nächsten Seite zeigt ein Beispiel mit folgendem Code:

```
\hvfFloat[doublePage,capWidth=n,
    capPos=right,capAngle=90]%
{figure}%
{\includegraphics[width=2\textwidth]
    {seiser}}%
[Eine Kurzbeschriftung für die LOF]%
{Eine Beschriftung einer
doppelseitigen Abbildung, die am
rechten Rand des rechten Teils
erscheint. Die Abbildung beginnt
im oberen linken Textbereich der
linken Seite. Eine Kurzform für
die LOF ist möglich.
Die Parameter lauten:
\texttt{doublePage},
\texttt{capWidth=n},
\texttt{capPos=right} und
\texttt{capAngle=90}. Das Foto zeigt
die Seiser Alm in den italienischen
Alpen.}%
{dP}
```

Mit `doublePage` kann zusätzlicher Text unter den beiden Teilen des Objekts erscheinen. Mit `capAngle=90` und `capPos=right` wird die Bildunterschrift gedreht und rechts vom Objekt mit einer Breite der Objekthöhe ausgegeben. Dies ist sinnvoll, wenn das Objekt kleiner als `\paperwidth+\textwidth` ist.

In manchen Fällen ist es erforderlich, einen Weißraum (Bindungskorrektur) zwischen den beiden Teilen des Objekts zu haben. Mit dem Schlüsselwort `bindCorr`



kann man einen Längenwert für den Zwischenraum festlegen, der rechts vom linken Teil und links vom rechten Bildteil hinzugefügt wird.

Option doublePAGE

Die Abbildung 14 mit einer Bindungskorrektur von 5 mm wird durch Abbildung 10 auf Seite 28 gezeigt, wogegen Abbildung 11 auf Seite 29 eine Bindungskorrektur von der Größe der Ränder zeigt. Die entsprechende »symbolische« Angabe lautet: `bindCorr=inner`.

Abgesehen davon, dass bei `doublePAGE` kein zusätzlicher Text auf der Doppelseite vorhanden ist, verhalten sich die beiden Schlüsselwörter `doublePage` und `doublePAGE` identisch. Wurde das Schlüsselwort `doublePage` gewählt, so wechselt `\hvFloat` automatisch zu `doublePAGE`, falls

Diese Variante empfiehlt sich für Abbildungen, die, bezogen auf die Dokumentenklasse, eine relativ große Höhe haben, aber schmaler als die doppelte Papierbreite sind. Das doppelseitige Objekt erscheint allein auf zwei gegenüberliegenden Seiten, auf denen kein zusätzlicher Text, außer einer Bildunterschrift, ausgegeben wird. Abbildung 11 auf Seite 29 zeigt drei Seiten eines Beispieldokuments. Die Beschriftung befindet sich unter dem Objekt in der ersten Spalte der rechten (ungeraden) Seite.

Abbildung 11 auf Seite 29 zeigt auch die Bedeutung des optionalen Schlüsselworts



Abb. 14: Eine Beschriftung einer doppelseitigen Abbildung, die am rechten Rand des rechten Teils erscheint. Die Abbildung beginnt im oberen linken Textbereich der linken Seite. Eine Kurzform für die LOF ist möglich. Die Parameter `lautendoublePage`, `capWidth=n`, `capPos=right` und `capAngle=90`. Das Foto zeigt die Seiser Alm in den italienischen Alpen.

`bindCorr` (Bindungskorrektur), das einen gewissen Leerraum zwischen dem geteilten Objekt erlaubt. Der Wert für `bindCorr` definiert den Leerraum für eine Seite; insgesamt ergibt sich daher der doppelte Wert für die beiden (links–rechts) Seiten. Es ist möglich, den inneren Rand für die Bindungskorrektur zu verwenden, um die beiden Bilder genau passend zur Textbreite zu erhalten. Der Wert für den inneren Rand kann durch die symbolische Angabe `bindCorr=inner` erreicht werden.

Die Definition der Gleitumgebung in Abbildung 11 auf Seite 29 wird als schwarzer Text in der ersten Spalte der ersten Seite angezeigt. Alle anderen Texte sind nur zu Demonstrationszwecken da und in grauer Farbe gedruckt.

Abgesehen davon, dass kein zusätzlicher Text vorhanden ist, verhalten sich die beiden Schlüsselwörter `doublePage` und `doublePAGE` identisch.

Option `doubleFULLPAGE`

In diesem Fall beginnt das Objekt der Gleitumgebung immer in der oberen linken Ecke der linken (geraden) Seite. Der definierte Textbereich hat keine Bedeutung, er wird bei diesen beiden Seiten komplett ignoriert. Die Beschriftung kann vor, nach, unter oder über dem Objekt gedruckt werden. Tabelle 3 auf Seite 22 erklärt die beiden optionalen Schlüsselwörter für `\includegraphics: doubleFULLPAGE` und `doubleFULLPAGEbindCorr` mit einer Voreinstellung von `keepaspectratio=false`.

Diese Schlüsselwörter machen nur den Code etwas lesbarer, haben aber ansonsten keine besondere Bedeutung für andere Objekte als Bilder, beispielsweise eine Tabelle oder Ähnliches.

Das Objekt selbst kann eine beliebige Breite und Höhe haben, sollte aber mindestens breiter als die angegebene `\paperwidth` und nicht weniger als 50 % von `\paperheight` hoch sein. Für kleinere Objekte nutzt man besser die anderen beiden Möglichkeiten `doublePage` oder `doublePAGE`.

Die Beschriftung des Objekts kann auch direkt über dieses gedruckt werden oder alternativ unten auf der Vor- oder Folgeseite des doppelseitigen Objekts (links–rechts). Bei einem zweispaltigen Dokument kann das Schlüsselwort `twoColCaption` verwendet werden, um beide Spalten für eine solche Beschriftung zu verwenden. Dies funktioniert nur für zweispaltige Dokumente, die den Spaltenmodus durch `\twocolumn` definieren, wie in diesem Beitrag für »Die $\TeX$ nische Komödie«. Das Paket `multicol` wird dagegen *nicht* unterstützt.

Abbildung 12 auf Seite 30 zeigt zwei Seiten mit einem Bild über einer Doppelseite, das klein genug ist, um rechts auf der Seite eine gedrehte Beschriftung zu erreichen, die hier zur Demonstration in Rot ausgegeben wurde. Das gezeichnete Seitenlayout hat nur den Zweck, die Funktionsweise des Makros `\hvFloat` und die Wirkung der einzelnen Schlüsselwörter besser zeigen zu können. Das Anzeigen des Seitenlayouts kann durch Laden des Pakets `showframe` erfolgen.

Der Code für Abbildung 12 auf Seite 30 lautet:

```
\hvFloat[doubleFULLPAGE,capPos=right]%
{figure}%
{\includegraphics
[height=\paperheight]%
{rheinsberg}}%
[Eine doppelseitige Abbildung mit
Beschriftung am rechten Rand.]%
{Eine Beschriftung einer
doppelseitigen Abbildung die am
rechten Rand des rechten Teils
erscheint. Die Parameter lauten:
\texttt{doubleFULLPAGE} und
\texttt{capPos=right}.}%
{fig:doubleFULLPAGE0n}
```

Wenn das Objekt fast das gleiche Seitenverhältnis wie die aktuelle Einstellung von `\paperwidth/\paperheight` hat, dann ist, wie in Abbildung 13 auf Seite 30 zu sehen, eine Bildunterschrift auf der folgenden Seite unten sinnvoller als eine Beschriftung über das Objekt zu setzen. Zur reinen Demonstration ist die Beschriftung in Rot ausgegeben worden. Die Beschriftung kann mit `capPos=before` auch unten auf die vorhergehende Seite gesetzt werden.

Der Code für Abbildung 13 auf Seite 30 lautet wie folgt:

```
\hvFloat[doubleFULLPAGE,capPos=after,
twoColumnCaption]{figure}%
{\includegraphics[doubleFULLPAGE]%
{rheinsberg}}%
{A caption for a double-sided image
that will be placed at the bottom
of the following page. The
doublepage image begins on the left
edge of the paper. A short form is
used for the LOF. The parameter is
\texttt{doubleFULLPAGE}}%
```

{fig:doubleFULLPAGE02ndnn}

Subfloats und Multifloats

Eine Gleitumgebung kann theoretisch einen beliebigen Inhalt haben, mit Ausnahme einer anderen Gleitumgebung. Der Name der Umgebung ist primär nur für den Befehl \caption von Interesse, um die richtige Bezeichnung auszugeben: Abbildung, Tabelle, ...

Die einzige Voraussetzung für den Inhalt ist, dass er kleiner als der Textbereich einer Seite sein muss. Der Inhalt selbst kann eine beliebige Kombination aus Text, Gleichungen, Tabellen und/oder Bildern sein. Hat man einen Inhalt mit mehreren gleichartigen Objekten, beispielsweise Abbildungen, die zueinander in irgendeiner inhaltlichen Beziehung stehen, so werden diese häufig zusammengefasst. Bekommt der gesamte Inhalt der Gleitumgebung einen Haupttitel und die einzelnen Objekte jeweils einen Untertitel, so bezeichnen wir diese als »subfloat«.

Wir bezeichnen sie als »multifloat«, wenn der Inhalt aus voneinander unabhängigen Objekten besteht, die keinen eigenen Haupttitel bekommen. Jedoch erhält jedes Teilobjekt seine eigene Bildunterschrift und die Zählung der Objekte erfolgt wie bei einzelnen Gleitumgebungen.

Tabelle 4 listet die beiden Schlüsselwörter auf, die solche speziellen Inhalte definieren. Diese können als standardmäßige Gleitumgebung, vollständige Spalte, vollständige Seite oder vollständige Doppelseite platziert werden.

Tab. 4: Zulässige Schlüsselwörter für mehrfache Objekte.

Name	Beschreibung
multiFloat	Für mehrere gleich- oder verschiedenartige Objekte mit eigener Hauptbeschriftung.
subFloat	Für mehrere gleichartige Objekte mit einer Haupt- und jeweiligen Teilbeschriftungen.

Die Syntax für das Makro, das solche Sub- oder Multifloats definiert, ist etwas komplex. Lediglich das Schlüsselwort definiert einen Typ multifloat oder subfloat. Die Syntax des Makros zeigt prinzipiell keinen Unterschied zwischen diesen Varianten. Mit dem Schlüsselwort vFill werden die Objekte einer Spalte (zweispaltiger Text) oder einer Seite (einspaltiger Text) über die vorgegebene Höhe \textheight vertikal gestreckt. Die Standardeinstellung ist kein Dehnen, sodass am unteren Rand der Spalte/Seite möglicherweise ein Leerraum angezeigt wird.

Subfloats

Eine Subfloat-Seite kann nur einen Objekttyp haben, der eine Hauptbeschriftung und einzelne Teilbeschriftungen hat. Werden keine Teilbeschriftungen definiert, so interessiert auch der Typ des Objekts nicht. Die Syntax ähnelt der für eine Multifloat-Seite, jedoch werden einige Argumente für einen Subfloat



(a) Eine »subcaption« A eines »fullpage« Teilobjekts.



(b) Eine »subcaption« B eines »fullpage« Teilobjekts, welches in der linken oder rechten Spalte erscheint.



(c) Eine »subcaption« C eines »fullpage« Teilobjekts.



(d) Eine »subcaption« D eines »fullpage« Teilobjekts.



(e) »subcaption« E eines »fullpage« Teilobjekts. Das Ganze hat die Teilbeschriftungen 15a...15e und die Hauptbeschriftung in der benachbarten Spalte.

ignoriert; man kann sie daher leer lassen. Die erste Zeile definiert nur den Gleittyp und die Hauptbeschriftung, der Objekteintrag wird ignoriert! Alle weiteren Zeilen haben den gleichen Gleittyp, weshalb dieser Eintrag bei folgenden Angaben der Teilobjekte ignoriert wird.

```
\hvFloat [Options]
+{float type}{}% the main type
[short caption] {long caption}
{label}
+{floating object}% sub object
[short caption] {long caption}
{label}
: ...
+{floating object}% sub object
[short caption] {long caption}
{label}
```

Das Symbol + definiert ein zusätzliches Objekt, das Teil derselben Gleitumgebung ist. Es liegt am Benutzer, sicherzustellen, dass eine Seite oder eine Spalte alle definierten Objekte platzmäßig aufnehmen kann.

Der Code für Abbildung 15, der die Teilfiguren 15a bis 15e zusammenfasst, lautet wie folgt:

```
\hvFloat[fullpage,capPos=after,subFloat,
vFill]%
+{figure}{}[Kurze Hauptbeschriftung des
\texttt{subfloats}]%
{Eine Haupt-Beschriftung für ein
```

Abb. 15: Eine Haupt-Beschriftung für ein »fullpage« Objekt, welches in der linken oder rechten Spalte erscheint. Dies kann eine gerade oder ungerade Seite sein.

```

»fullpage« Objekt, welches in der
linken oder rechten Spalte erscheint.
Dies kann eine gerade oder ungerade
Seite sein.}%
{sub:demo}%
+{{\includegraphics[trim=0 8cm 0 6cm,
clip, columnWidth]{usa9}}}%
[Short caption A]%
{Eine »subcaption« A eines »fullpage«
Teilobjekts.}%
{sub:demo0}%
+{{\includegraphics[trim=0 5cm 0 6cm,
clip, columnWidth]{usa5}}}%
{Eine »subcaption« B eines »fullpage«
Teilobjekts, welches in der linken
oder rechten Spalte erscheint.}%
{sub:demo1}%
+{{\includegraphics[trim=0 8cm 0 6cm,
clip, columnWidth]{usa6}}}%
{Eine »subcaption« C eines »fullpage«
Teilobjekts.}%
{sub:demo2}%
+{{\includegraphics[trim=0 8cm 0 6cm,
clip, columnWidth]{usa7}}}%
{Eine »subcaption« D eines »fullpage«
Teilobjekts.}%
{sub:demo3}%
+{{\includegraphics[trim=0 7cm 0 7cm,
clip, columnWidth]{usa8}}}%
{»subcaption« E eines
»fullpage« Teilobjekts. Das Ganze
hat die Teilbeschriftungen
\ref{sub:demo0}\ldots\ref{sub:demo5}
und die Hauptbeschriftung in der
benachbarten Spalte.}%
{sub:demo5}

```

Das Schlüsselwort `subFloat` definiert die Bilder oder Tabellen als Teilobjekte. Die Angabe des Gleittyps `figure` in der zweiten Zeile des Codes definiert den Gleittyp der Gleitumgebung; alle Unterobjekte müssen vom gleichen Typ sein. Aus die-

sem Grund sind alle folgenden Parameter, die dem Zeichen `+` folgen, leer: `+{}{...}`

Das Paket `subcaption` wird standardmäßig geladen und sollte mit

```
\captionsetup[sub]{singlelinecheck}.
```

aktiviert werden, damit Beschriftungen, die kürzer als eine Zeile sind, automatisch zentriert gesetzt werden.

Das Hauptlabel des Subfloats ist `sub:demo`, das auf die Objektspalte auf Seite 36 zeigt. In diesem Fall zeigt das interne Label `sub:demo-cap` auf dieselbe Seite 36, da Objekt und Bildunterschrift in unterschiedlichen Spalten liegen, aber auf derselben Seite. Beide beziehen sich auf dasselbe Objekt:

```
\ref{sub:demo} → 15 und \ref{sub:demo-cap} → 15.
```

Multifloats

```

\hvFloat [Options]
+{float type}{floating object}
[short caption] {long caption}
{label}
+{float type}{floating object}
[short caption] {long caption}
{label}
: ...
+{float type}{floating object}
[short caption] {long caption}
{label}

```

Jedes Objekt bekommt eine eigene Beschriftung, weshalb Abbildungen und Tabellen und andere Objekte gemischt werden können.

Alle individuellen Beschriftungen werden vor oder nach der vollen Spalte/Seite am Ende der Spalte/Seite aufgelistet (siehe Abbildung 16 und folgende). Das Symbol + definiert ein Folgeobjekt, das Teil derselben Gleitumgebung ist. Für ein Multifloat-Objekt sind alle Parameter gültig, da jedes als eigenständiges Objekt innerhalb der Gleitumgebung aufgefasst wird. Es liegt am Benutzer, sicherzustellen, dass eine Seite oder eine Spalte alle definierten Objekte aufnehmen kann.

Die Bildbeschriftungen der Abbildungen 16 bis 19 und der Tabellen 5 und 6 befinden sich auf Seite 38, während alle Objekte auf Seite 39 erscheinen. Abbildung 16, Tabelle 5 und 6, sowie die Abbildungen 18 bis 19 sind alle Teil desselben Multifloats.

Durch die Vorgabe `capPos=before` erscheint die Beschriftung vor dem Objekt, unabhängig davon, ob das günstig ist oder nicht. Für das Beispiel 16 auf der nächsten Seite (und folgende) wäre die Angabe `capPos=left` oder `capPos=right` sinnvoller; dann wären die Beschriftungen nicht auf dieser Seite und die Objekte auf der nächsten Seite. Aus reinen Demonstrationsgründen ist dies hier aber so gewollt. Der Code des Multifloat-Beispiels auf Seite 39 lautet:

```
\captionsetup{singlelinecheck=false}
\hvfFloat[fullpage, capPos=before,
  multiFloat, vFill]%
+{figure}%
{\includegraphics[columnWidth]{mv1}}%
[Kurzversion der Caption 1]%
{Die Beschriftung eines »fullpage«
  Objekts, welches in der linken oder
  rechten Spalte erscheint. Dies kann
```

```
eine gerade oder ungerade Seite sein.
Und noch etwas sinnloser Text zur
Demonstration dieses Beispiels.}%
{img:multidemo}%
+{table}{\small\tabcolsep=3pt
\begin{tabular}{@{}lrcp{3cm}@{}}
\hline
Left & Right & Center & Parbox\
\hline
L      & R      & C      & P\
left   & right   & center  & Text
      with possible linebreaks\
L      & R      & C      & P\
\multicolumn{4}{c}{Multicolumn over
all columns}\
\end{tabular}}%
[Kurzversion B1]%
```

Abb. 16: Die Beschriftung eines »fullpage« Objekts, welches in der linken oder rechten Spalte erscheint. Dies kann eine gerade oder ungerade Seite sein. Und noch etwas sinnloser Text zur Demonstration dieses Beispiels.

Tab. 5: Eine Beschriftung eines »fullpage« Objekts, welches in der linken oder rechten Spalte erscheint. Dies kann eine gerade oder ungerade Seite sein.

Abb. 17: Eine Beschriftung eines »fullpage« Objekts, welches in der linken oder rechten Spalte erscheint.

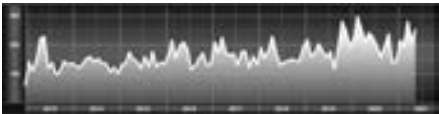
Abb. 18: Eine Beschriftung eines »fullpage« Objekts, welches in der linken oder rechten Spalte erscheint.

Abb. 19: Eine Beschriftung eines »fullpage« Objekts, welches in der linken oder rechten Spalte erscheint.

Tab. 6: Eine Beschriftung für ein »fullpage« Objekt, welches auf der linken oder rechten Spalte erscheint. Dies kann eine gerade oder ungerade Seite sein.



Left	Right	Center	Parbox
L	R	C	P
left	right	center	Text with possible line-breaks
L	R	C	P
Multicolumn over all columns			



Left	Right	Centered	Parbox
L	R	C	P
left	right	center	Text with possible line-breaks
Multicolumn over all columns			

```
{Eine Beschriftung eines »fullpage«
Objekts, welches in der linken oder
rechten Spalte erscheint. Dies kann
eine gerade oder ungerade Seite sein.}
{tab:multidem0}%
+{figure}%
{\includegraphics[columnWidth]{CTAN1}}
{Eine Beschriftung eines »fullpage«
Objekts, welches in der linken oder
rechten Spalte erscheint.}%
{img:multidem01}
+{figure}%
{\includegraphics[columnWidth]{mv0}}
{Eine Beschriftung eines »fullpage«
Objekts, welches in der linken oder
rechten Spalte erscheint.}%
{img:multidem02}
+{figure}%
{\includegraphics[columnWidth]{CTAN3}}
{Eine Beschriftung eines »fullpage«
Objekts, welches in der linken oder
rechten Spalte erscheint.}%
{img:multidem03}
+{table}{\small\tabcolsep=3pt
\begin{tabular}[b]{@{}c{}}\lrcp{3cm}@{}{}}
\hline
Left & Right & Centered & Parbox\\
\hline
L & R & C & P\\
left & right & center & Text
with possible linebreaks\\
\multicolumn{4}{c}{Multicolumn over
all columns}\\
\hline
\end{tabular}}%
[Kurzbeschriftung B2]%
{Eine Beschriftung für ein »fullpage«
Objekt, welches auf der linken oder
rechten Spalte erscheint. Dies kann
eine gerade oder ungerade Seite sein.}
{tab:multidem02}
```

Tabellen horizontal teilen

Standardmäßig kann eine Tabelle als sogenannte `longtable` nur in vertikaler Richtung gesplittet werden. Große Tabellen können zwar auf einer Seite gedreht werden (siehe Tabelle 2 auf Seite 19), ein automatisches Teilen in horizontaler Richtung, um sie auf zwei Seiten zu verteilen, ist jedoch mit Standard- $\text{\LaTeX}$ nicht möglich. Das Speichern einer Tabelle ohne Seitenumbrüche in einer Box ist kein Problem und eine solche Box kann wie eine Abbildung behandelt werden, welche letztlich auch nur eine Box ist. Das einzige Problem ist, dass die Tabelle horizontal sinnvollerweise *zwischen* zwei Spalten aufgeteilt werden muss, sonst macht es keinen Sinn: Eine geteilte Spalte kann unlesbar sein.

Das Paket `hvfloater` stellt die sogenannte `Savebox \hv0Box` bereit, in der man zuerst die Tabelle zwischenspeichern kann:

```
\savebox\hv0Box{%
  \begin{tabular}{@{} *{12}c @{}}
    ... the tabular ...
  \end{tabular}}
```

Um sie dann auf die gleiche Weise wie ein doppelseitiges Bild zu verwenden, kann die Tabelle in zwei Teile geteilt werden. Erfolgt die Aufteilung an einer ungünstigen Stelle in der Tabelle, z. B. in der Mitte einer Spalte, dann fügt man mit `@{\hspace{...}}`, einen horizontalen Abstand zwischen den beiden Spalten ein, beispielsweise wie folgt:

```
\begin{tabular}{111 @{\hspace{1cm}} 11}
  \hline
  1 & 2 & 3 & 4 & 5 \\
  1 & 2 & 3 & 4 & 5 \\
  1 & 2 & 3 & 4 & 5
```

```
1 & 2 & 3 & 4 & 5 \\ \hline
\end{tabular}
```

Das führt dann zur Tabelle 7.

Tab. 7: Erzeugen eines zusätzlichen Zwischenraums zwischen zwei Spalten.

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

Gleitumgebungen		Ganz- und doppel-			
upper	Schüler	Arbeitslos	Reimer	Student	S
11	8	12	71	200	
11	6	11	78	197	
11	4	11	80	165	
18	9	12	77	157	
14	9	12	79	148	
15	10	11	82	134	
11	7	11	89	126	
7	3	8	97	103	
9	4	10	99	96	
8	3	9	103	97	

Abb. 20: Die Tabelle hat eine ungünstige Spaltenanordnung zwischen den beiden Seiten.

Abbildung 20 zeigt die tabellarische Darstellung in der Mitte der Doppelseite. Zufälligerweise ist die Spaltenanordnung ungünstig; die Spalte *Arbeitslos* genau in der Mitte der Doppelseite und sollte daher nach rechts verschoben werden. Durch Ausprobieren erhält man einen sinnvollen Zwischenraum von 3 em, der *nach* der Spalte *Schüler* eingefügt wird.

```
\begin{tabular}{@{}*8c@{\hspace{6em}}*4c}
```

Es ergibt sich dann eine Anordnung entsprechend Abbildung 21 und mit 10 mm

Bindungskorrektur, was ebenfalls durch Ausprobieren gefunden wurde.

Zwischenraum von 20\,mm ergibt.}%
{tab:dP}

Gleitumgebungen		Ganz- und doppel	
upper	Schüler	Arbeitslos	Ren:
11	8	12	7
11	6	11	7
21	4	11	8
18	9	12	7
14	9	12	7
15	10	11	8
11	7	11	8
7	3	8	9
9	4	10	9

Abb. 21: Korrektur des Spaltenabstandes zwischen linker und rechter Seite.

Der Code für die geteilte Tabelle auf einer Doppelseite lautet:

```
\hvFloat[doublePage,capWidth=n,  
capPos=right,bindCorr=1cm,  
capVPos=top]%  
{table}%  
{\usebox\hvOBox}%  
[Eine doppelseitige Tabelle]%  
{Entwicklung der Mitgliederzahlen von  
DANTE \eV\ (Stand jeweils DTK Nr.\,1  
des angegebenen Jahres)\newline  
Eine Beschriftung für eine  
doppelseitige Tabelle, die auf der  
rechten Seite des  
rechten Teils der Tabelle platziert  
wird. Die Tabelle beginnt am linken  
Rand des Textbereich der linken Seite.  
Für das LOF wird eine Kurzform der  
Bildunterschrift verwendet. Der  
Parameter ist \texttt{doublePage} und  
der zusätzliche Abstand zwischen  
den Spalten \emph{Schüler} und  
\emph{Arbeitslos} beträgt 3\,em.  
Die Bindekorrektur ist auf 10\,mm  
eingestellt, was einen zusätzlichen
```

Diese Tabelle 8 ist auf den Seiten 42 und 43 zu sehen.

To-do-list

Das Makro \hvFloat überprüft nur die Position seines Aufrufs, ob es auf einer ungeraden oder geraden Seite definiert ist. Dies geschieht mit Hilfe des Makros \checkoddpage aus dem Paket \ifoddpage. [6]

Zusammen mit dem internen L^AT_EX-Makro \if@firstcolumn kennt es genau die Position seines Aufrufs im Quelltext des Dokuments: Linke oder rechte Seite, erste oder zweite Spalte. Es weiß aber nicht, ob die aktuelle Seite komplett leer ist, was der Fall ist, wenn \hvFloat zufällig der erste Befehl auf einer neuen Seite ist. Ist dies auch eine gerade Seite, könnte sofort ein doppelseitiges Objekt platziert werden.

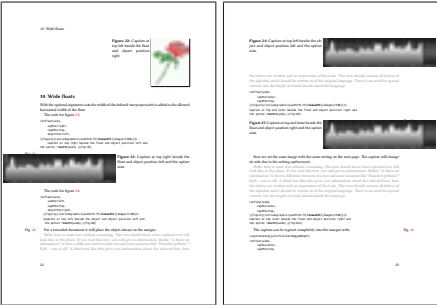


Abb. 22: Die Seiten 22 und 23 der Paketdokumentation, welche die Bedeutung des Schlüsselworts wide zeigen.

Jahr	Gesamtzahl	Privat	Institut	Firma	Ehren	Schnupper	Schüler
2007	2074	1626	113	25	8	11	8
2008	2120	1674	109	26	8	11	6
2009	2122	1696	108	29	8	21	4
2010	2098	1684	107	26	8	18	9
2011	2075	1673	106	26	8	14	9
2012	2075	1681	107	27	8	15	10
2013	2057	1680	99	27	7	11	7
2014	2025	1680	93	27	7	7	3
2015	2030	1686	90	27	9	9	4
2016	1996	1652	88	28	8	8	3
2017	1966	1638	83	26	8	3	2
2018	1925	1619	77	26	8	3	3
2019	1884	1589	76	25	8	6	1
2020	1858	1564	72	26	8	5	0

Der aktuelle Code verwendet jedoch immer die nächste gerade-ungerade Seitenkombination, da der Test auf einen Seitenumbruch in $\text{\TeX}$ prinzipiell nicht einfach ist.

Einige weitere Überprüfungen zur korrekten Verwendung der Parameter sind ebenfalls nützlich. Beispiel: Wenn man das Schlüsselwort `doubleFULLPAGE` mit einem Objekt verwendet, das kleiner als `\textwidth` ist, dann ist die Ausgabe unsinnig.

Das Schlüsselwort `wide`, wie in Abbildung 22 auf der vorherigen Seite gezeigt, funktioniert im ein- und zweiseitigen Modus nur dann korrekt, wenn man den einspaltigen Modus verwendet (siehe Abbildung 23 auf Seite 45). Für den Modus `twoside` und `twocolumn` existieren unterschiedliche Ränder für eine mögliche Gleitumgebung mit dem Parameter `wide` in der ersten und zweiten Spalte.

Dieser Umstand wird von `hvfloater` nicht erkannt. Wenn man jedoch in einem zweiseitigen und zweisepaltigen Modus überbreite Gleitumgebungen benötigt, so kann man das Makro `\hvfloater` einfach an Stellen im Dokument verschieben, an denen die Ausgabe immer in der äußeren Spalte erfolgt, die auch die `\marginpar`-Spalte enthält. Bei Verwendung des Arguments `nonFloat`, wie in Abbildung 23 auf Seite 45, erscheint das Objekt genau an der Stelle der Definition.

In manchen Fällen funktioniert das Schlüsselwort `useOBox` (siehe Seite 18) für eine vom Benutzer vordefinierte Box `\hvOBox` nicht korrekt. In solchen Fällen kann man stattdessen `\usebox\hvOBox` als normales Argument für das Objekt verwenden, was letztendlich den gleichen Effekt hat (siehe Beispielcode auf Seite 41). Die Anwendung des Schlüsselwortes `useOBox` kann dann entfallen.

Arbeitslos	Rentner	Student	Sonstiges
12	71	200	3
11	78	197	3
11	80	165	4
12	77	157	4
12	79	148	6
11	82	134	6
11	89	126	6
8	97	103	8
10	99	96	8
9	103	97	9
8	111	87	9
6	114	69	8
6	111	62	9
7	109	67	9

Tab. 8: Entwicklung der Mitgliederzahlen von DANTE e.V. (Stand jeweils DTK Nr. 1 des angegebenen Jahres)
Eine Beschriftung für eine doppelseitige Tabelle, die auf der rechten Seite des rechten Teils der Tabelle platziert wird. Die Tabelle beginnt am linken Rand des Textbereich der linken Seite. Für das LOF wird eine Kurzform der Bildunterschrift verwendet. Der Parameter ist doublePage und der zusätzliche Abstand zwischen den Spalten Schüler und Arbeitslos beträgt 3 em. Die Bindekorrektur ist auf 10 mm eingestellt, was einen zusätzlichen Zwischenraum von 20 mm ergibt.

Zusammenfassung

Das Paket hvfloat sollte mit allen Dokumentenklassen funktionieren, einseitig im ein- oder zweispaltigen Modus, zwei-seitig im ein- oder zweispaltigen Modus. Bei einem einspaltigen Dokument ist es viel einfacher, doppelseitige Objekte zu platzieren als bei einem zweispaltigen Do-kument. Intern fügt T_EX zwei einzelne Spalten zu einer Seite mit zwei Spalten zusammen. Über diese „zwei“ Seiten wer-den nur die optionalen Kopf- und Fußzei-len gedruckt.

Das Paket hvfloat verwendet intensiv das Makro \afterpage. [1] Definiert man in der ersten Spalte einer linken (gera-den) Seite ein zweiseitiges Gleitobjekt, dann benötigt \hvFloat drei verschach-telte \afterpage-Aufrufe bis es auf einer leeren linken (geraden) Seite erscheinen kann. Für jede Spalte eine Anweisung, damit ein Objekt oder eine Überschrift

auf der nächsten linken Seite beginnen kann. Bis L^AT_EX diese Seite für das Objekt oder die Beschriftung erreicht, müssen bis zu zwei Seiten mit Text oder anderen Objekten gefüllt werden, die erst *nach* dem Makro \hvFloat im Quelltext er-scheinen. Insbesondere im Zweispalten-Modus ist mit Problemen zu rechnen, wenn das Dokument zu wenig Text oder andere einfache nicht-gleitende Objekte aufweist, um diese beiden Seiten zu fül-len, bis das Objekt auf eine Doppelseite gesetzt werden kann.

Solche Probleme können nur gelöst wer-den, indem Text hinzugefügt oder das Ma-kro \hvFloat in eine andere Spalte des Dokuments verschoben wird, was im All-gemeinen kein Problem sein sollte. Dies kann dennoch sehr zeitaufwändig sein, wenn man beispielsweise wie in diesem Beitrag *sehr* viele Abbildungen hat, eini-ge davon über die ganze Spalte oder auch

Doppelseiten gehend und dabei eben relativ wenig Text zwischen den einzelnen Gleitumgebungen existiert.

Wie bei den standardmäßigen Gleitumgebungen `figure` und `table` bleibt es dem Benutzer überlassen, sicherzustellen, dass der Inhalt der Umgebung auf die Seite passt. Wenn ein Objekt breiter als `2\paperwidth` und/oder höher als `\paperheight` ist, kann es nicht auf einer Doppelseite platziert werden und die Ausgabe ist möglicherweise nutzlos.

Einfache und nicht-gleitende Objekte können auch mithilfe der Umgebung `hvFloatEnv` platziert werden. Diese hat nur ein optionales Argument für die Breite der Umgebung:

```
\begin{hvFloatEnv}[0.5\columnwidth]
\caption{Nichtgl. Tabelle}
\begin{tabular}{@{} l c r @{}}\hline
left & center & right \\
L & C & R \\
\end{tabular}
\end{hvFloatEnv}
```

Tab. 9: Nichtgl. Tabelle

left	center	right
L	C	R

Das Abbildungs- und Tabellenverzeichnis wird vom Paket `hvfloat` nicht beeinflusst und sollte wie gewohnt funktionieren. Hier zum Beispiel die Liste der Tabellen dieses Artikels:

Tabellenverzeichnis

1	Eine sinnlose Beschriftung ...	17
---	--------------------------------	----

2	Die möglichen optionalen Schlüsselwörter für das Makro <code>\hvFloat</code>	19
3	Erweiterung von <code>graphicx</code>	22
4	Zulässige Schlüsselwörter für mehrfache Objekte.	35
5	Kurzversion B1	38
6	Kurzbeschriftung B2 . . .	38
7	Zusätzlicher Soaltenzwischenraum	40
8	Eine doppelseitige Tabelle	43
9	Nichtgl. Tabelle	44

Bevor man eine Doppelseite für ein Objekt verwendet, sollte man vorher testen, ob es vielleicht schon ausreicht, den Rand zu nutzen, um zusätzlichen Platz für ein größeres Objekt zu bekommen. `\hvFloat` kennt das Schlüsselwort `wide`, welches den horizontalen Platz um die Breite von `\marginparwidth` vergrößert, sodass nur noch `\marginparsep` als Abstand zum Papierrand bleibt. Die Beschriftung kann wie gewohnt über/unter oder links/rechts relativ zum Objekt platziert werden. Auch die Verwendung der Innen-/Außenposition für zweiseitige Dokumente ist möglich. Abbildung 22 zeigt einige Beispiele für ein einspaltiges Dokument mit der folgenden Verwendung von `\hvFloat`.

```
\hvFloat[wide, capPos=inner,
capVPos=top]{figure}%
{\includegraphics[width=0.75\linewidth]{CTAN}}%
{Beschriftung oberhalb und neben dem
Objekt am inneren Rand mit der Option
\texttt{wide}.}{fig:wide}
```

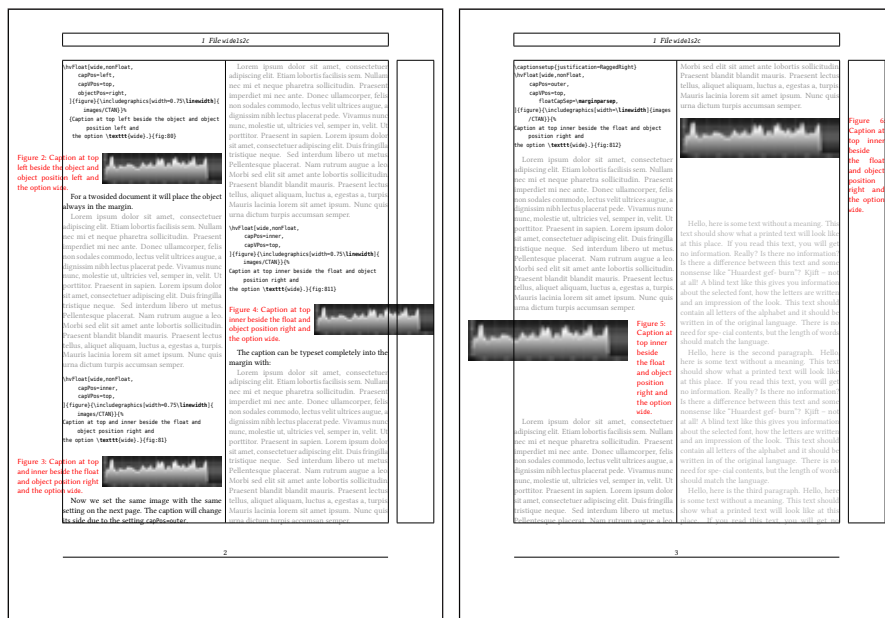


Abb. 23: Die Seiten 2 und 3 des Beispieldokuments `wide1s2c.tex` (einseitig und zweiseitig mit `wide=true`).

Literatur

- [1] David Carlisle, The L^AT_EX Team: The afterpage package, Execute command after the next page break, Version 1.08, 2014, CTAN: /macros/latex/required/tools (besucht am 1. 6. 2021).
- [2] Hubert Gäßlein, Rolf Niepraschk: The sidecap package, Typeset captions sideways, Version 1.6f, 2003, CTAN: /macros/latex/contrib/sidecap (besucht am 8. 6. 2021).
- [3] Markus Kohm: The KOMA-Script package, KOMA-Script >book< class, Version 3.33, 2021, CTAN: /macros/latex/contrib/koma-script (besucht am 9. 6. 2021).
- [4] F. Mittelbach u. a.: The L^AT_EX Companion, 2. Aufl., Tools and Techniques for Computer Typesetting, Pearson Education, 2004, ISBN: 9780133387643.
- [5] Martin Scharrer: The adjustbox package, Graphics package-alike macros for »general« boxes, Version 1.3, 2020, CTAN: /macros/latex/contrib/adjustbox (besucht am 1. 6. 2021).

- [6] Martin Scharrer: The `ifoddpage` package, Determine if the current page is odd or even, Version 1.1, CTAN: `/macros/latex/contrib/ifoddpage` (besucht am 1. 6. 2021).
- [7] Axel Sommerfeldt: The `caption` package, Customising captions in floating environments, 2020, CTAN: `/macros/latex/contrib/caption` (besucht am 1. 6. 2021).
- [8] — The `subcaption` package, Support for sub-captions, Version 1.3i, 2020, CTAN: `/macros/latex/contrib/caption` (besucht am 1. 6. 2021).
- [9] Sigitas Tolušis: The `stfloats` package, Commands to control the presentation of floats, Version 3.3, 2017, CTAN: `/macros/latex/contrib/sttools` (besucht am 1. 6. 2021).
- [10] Herbert Voß: Die wissenschaftliche Arbeit mit $\text{\LaTeX}$ – unter Verwendung von $\text{\LuaTeX}$, KOMA-Script und Biber/Bib $\text{\LaTeX}$, 2. Aufl., DANTE Edition, Lehmanns Media, Berlin, 2021, S. 448, ISBN: 978-3-96543-217-8, <https://www.lehmanns.de/shop/mathematik-informatik/56594229-9783965432178-die-wissenschaftliche-arbeit-mit-latex>.
- [11] — The `hvfloating` package, Rotating caption and object of floats independently, Version 2.29, 2021, CTAN: `/macros/latex/contrib/hvfloating` (besucht am 1. 6. 2021).

Dymo-Aufkleber mit $\text{\LaTeX}$ gestalten

Uwe Ziegenhagen

In Ergänzung des Artikels zum Paket `ticket` in »Die $\text{\TeX}$ nische Komödie« 1/2021 soll in diesem Artikel gezeigt werden, wie man mit $\text{\LaTeX}$ auch Adressaufkleber für Label-Drucker erzeugen kann.

Label-Definition und $\text{\LaTeX}$ -Dokument

Seit einigen Jahren besitze ich einen »Dymo LabelWriter 450«, der Adressetiketten mittels Thermotransferdruck direkt von der Rolle druckt. Er ist auch in die Windows-Druckerverwaltung eingebunden, sodass er aus dem Adobe Reader und anderen PDF-Viewern angesprochen werden kann.

Zur Spezifikation der von mir genutzten 57×32 Millimeter großen Etiketten habe ich eine Textdatei `Dymo5732.tdf` angelegt, die den Inhalt aus Listing 1 enthält. Die Einstellungen wurden dabei so gewählt, dass die durch das Paket `ticket` erzeugte

Box genau in die Seite passt. Die Offset-Werte wurden durch wiederholtes Probieren ermittelt.

Listing 1: Ticketdefinition

```
\unitlength=1mm
\hoffset=-25.4mm
\voffset=-29mm
\ticketNumbers{1}{1} % Anzahl der Labels pro Bogen
\ticketSize{57}{32} % Breite und Hoehe der Labels in mm
\ticketDistance{0}{0} % Abstand der Labels
```

In der L^AT_EX-Datei, siehe Listing 2, erfolgte dann die Definition der Seitengröße und das Setzen der Ränder. Die Option `landscape` erwies sich dabei als notwendig; ohne diese druckte der Dymo die Label um 90 Grad gedreht über die einzelnen Aufkleber hinaus.

Für die Definition der einzelnen Tickets wurde dann der Befehl `\myticketml` definiert, der nebst dem Parameter für die Skalierung drei weitere Parameter für die Inhalte der drei einzelnen Zeilen einer Adresse bietet. Reichen diese drei Zeilen aus dem Listing nicht, so lassen sich problemlos noch weitere Zeilen über zusätzliche Parameter hinzufügen.

Listing 2: L^AT_EX-Datei

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{graphicx}
\usepackage[landscape,paperheight=57mm,paperwidth=32mm,%
left=0mm,top=0mm,right=0mm,bottom=0mm]{geometry}
\usepackage{palatino}

\usepackage[Dymo5732]{ticket}
\renewcommand{\ticketdefault}{} % Reset des Tickets

\newcommand{\myticketml}[4]{
\ticket{%
\put(5,20){\scalebox{#1}{\bfseries #2}}
\put(5,15){\scalebox{#1}{\bfseries #3}}
\put(5,10){\scalebox{#1}{\bfseries #4}}
}}

\begin{document}
\myticketml{1.25}{Dr. Max Mustermann}{Musterweg 123}{54321~Musterstadt}
\end{document}
```

Das erzeugte Label ist in Abbildung 1 abgedruckt. Anpassungen an andere Label-Größen oder an andere Drucker, die nach ähnlichem Prinzip arbeiten, sind einfach zu bewerkstelligen.



Abb. 1: Das vom $\text{\LaTeX}$ -Code erzeugte Label.

Ein letzter Hinweis: Nach dem Artikel in »Die $\text{\TeX}$ nische Komödie« 1/2021 bekam ich von Bernd Raichle noch den Tipp, die $\text{\TeX}$ -Code Hacks zwischen `\makeatletter` und `\makeatother` besser durch die entsprechenden Paket-Optionen zu ersetzen. Dies möchte ich hiermit gern auch weitergeben.

Einbetten von Statistik R-Code in $\text{\LaTeX}$

Walter Entenmann

Die Programmiersprache R ist die am häufigsten verwendete Open-Source-Software in der Statistik zur Verarbeitung und grafischen Darstellung von empirischen Daten. Mit den Zusatzpaketen `Sweave` und `knitr` können die R-Code-Sequenzen direkt in ein $\text{\LaTeX}$ -Dokument als Chunks eingebettet werden. Die Bearbeitung erfolgt in zwei Schritten mit `Rscript` und `lualatex`. Als Ergebnis erhält man eine pdf-Datei. Der Vorteil dieser Einbettung besteht darin, dass nur ein Dokument erstellt und bearbeitet werden muss, das sowohl den Bericht als auch die R-Code-Sequenzen umfasst. Bei der Bearbeitung werden die R-Anweisungen ausgeführt und die Ergebnisse und Diagramme in das Dokument eingebunden. Die Arbeit beschreibt die Vorgehensweise mit `Sweave` und `knitr`, das mehr Bedienungskomfort und zusätzliche grafische Möglichkeiten bietet. Schließlich wird noch gezeigt, wie man mit Hilfe des Pakets `tikzDevice` die Grafiken mit $\text{\LaTeX}$ beschriften kann.

Kurze Einführung in R

Das R-Projekt besteht im Kern aus der Programmiersprache R als Skriptsprache mit Interpreter und sehr vielen statistischen Funktionen, einer Bibliothek und einem Archiv (CRAN) mit unzähligen weiteren Paketen für spezielle statistische Anwendungen, vergleichbar mit T_EX, L^AT_EX, seinen Standardpaketen und dem CTAN-Archiv. Wir gehen im Folgenden davon aus, dass das Softwarepaket R installiert ist, das für Windows, Linux und Mac von der Homepage des R-Projekts <https://www.r-project.org/> heruntergeladen werden kann. Dieser Artikel wurde unter MacOS erstellt.

Historisch ist R aus dem Statistikprogramm S entstanden, das 1976 von John M. Chambers in den Bell Laboratories entwickelt wurde. Das heutige Open-Source-Programm R wird von Ross Ihaka und Robert Gentleman und dem R-Core-Team betreut und gewartet. Das Einbettungsproblem wurde 2002 von Friedrich Leisch, Institut für Statistik, LMU München mit dem Paket Sweave gelöst. Das Paket knitr von Yihui Xie (2013) ist eine Weiterentwicklung, inspiriert von Sweave, mit vielen Optionen zur komfortablen Gestaltung der Grafik. Eine ausführliche Dokumentation findet man auf der Home-Page von <https://www.r-project.org/> [8].

Interaktives Arbeiten mit R

R wird am einfachsten interaktiv verwendet, indem man im Terminal das Kommando 'R' eintippt und mit der Enter-Taste abschickt. Daraufhin öffnet sich das Fenster des Work-Space von R mit dem Prompt '>'. Nach dem Prompt erwartet R die Eingabe eines R-Kommandos, abgeschlossen (abgeschickt) mit der Enter-Taste. Mit dem Zeichen '<-' (assign) kann man einer Variablen einen Wert zuweisen. Das Zeichen '#' beginnt einen Kommentar bis zum Zeilenende, der von R ignoriert wird. Zum Beenden der interaktiven R-Sitzung muss man das Kommando 'q()' eingeben und die Frage mit 'y' beantworten. Dann befindet man sich wieder im Terminal mit dem Prompt '\$' der bash-Shell. R funktioniert im Prinzip wie ein komfortabler Taschenrechner.

```
$ R
> # mein erster Versuch mit R
> 2*3
[1] 6
> q()
save workspace image? (y/n/c): y
$
```

R-Quelldatei

Bei fortgeschrittener Benutzung von R wird man die R-Kommandos in eine R-Quelldatei `file.R` schreiben und interaktiv mit

```
> source("file.R")
```

oder im Terminal mit dem Kommando

```
R CMD BATCH -q file.R
```

bearbeiten.

Beispiel aus der Statistik

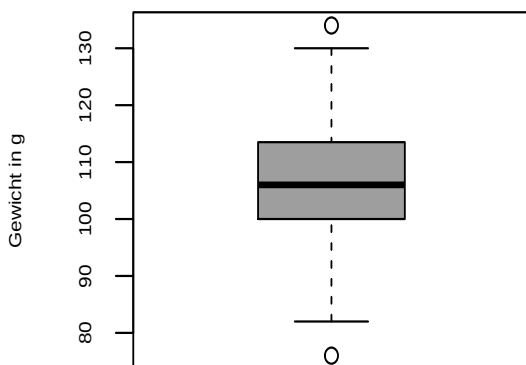
Als kleine statistische Anwendung messen wir von 100 zweiwöchigen Küken deren Gewicht in Gramm [5]. Diese Stichprobe ist eine Liste von 100 Zahlenwerten, die wir mit R typisch in einem Vektor `x` zusammenfassen. Nun können wir von dieser Stichprobe einige statistische Kenngrößen berechnen lassen, z. B. den kleinsten und größten Wert, den Mittelwert, den Median und die Standardabweichung. Hierfür stellt R leistungsfähige Funktionen bereit.

```
x <- c(107, 108, 111, 101, 97, 113, 109, 105, 116, 122, 117, 98, 114, 82, 92,
102, 100, 112, 115, 106, 105, 104, 121, 111, 112, 103, 76, 92, 96, 109, 106,
126, 114, 108, 105, 100, 101, 99, 125, 116, 114, 102, 104, 100, 112, 94, 95,
118, 97, 103, 105, 100, 94, 109, 100, 102, 96, 100, 114, 134, 113, 120, 101,
92, 108, 104, 118, 130, 111, 86, 88, 121, 94, 96, 105, 110, 91, 112, 101, 124,
119, 87, 95, 108, 97, 127, 118, 110, 101, 107, 116, 110, 114, 108, 119, 102,
107, 103, 90, 107)
N <- length(x)
min(x)
[1] 76
max(x)
[1] 134
mean(x)
[1] 106.19
median(x)
[1] 106
sd(x)
[1] 10.63698
```

Grafische Darstellung

Um auch von den grafischen Möglichkeiten von R einen Eindruck zu vermitteln, wollen wir die Eigenschaften der Stichprobe durch einen Boxplot veranschaulichen, der den kleinsten und größten Wert zeigt, den Median (mittlerer Wert) Q_2 und die Quartile Q_1 und Q_3 für 25 % bzw. 75 % sowie die Whisker im Abstand von 1.5 mal IQR ober- und unterhalb der Box. Der Interquartilabstand IQR ist gleich $Q_3 - Q_1$.

```
boxplot(x,range=1.5,col=8,xlab="2-wöchige Küken",
ylab="Gewicht in g",cex.lab=0.55,cex.main=0.55,cex.sub=0.55,cex.axis=0.55)
```



2-wöchige Küken

Abb. 1: Boxplot: Gewicht der Küken.

Es gibt in R mehrere Grafikfunktionen und ca. 40 Grafikparameter, siehe [8]. Wir haben hier einige davon verwendet.

Funktion/Option	Beschreibung
<code>boxplot(x,range=1.5, ...)</code>	Plotfunktion zum Zeichnen eines Boxplots
<code>xlab=, ylab=</code>	Beschriftung der x - und y -Achse
<code>main=, sub=</code>	Titel, Untertitel
<code>cex.main=, cex.sub=</code>	relative Schriftgröße (default 1) für Titel und Untertitel

<code>cex.axis=</code>	relative Schriftgröße der Achsenbezeichnung
<code>cex.lab=</code>	relative Schriftgröße der Achsenbeschriftung
<code>col=</code>	Farbwert (8=grau)
<code>par(opt1=wert1, ...)</code>	Funktion zum globalen Setzen von R-Grafik-Parametern in einem Chunk, die dann auch für alle folgenden Chunks gelten.

Einbetten von R-Code in ein L^AT_EX-Dokument

Mit den Paketen `Sweave` und `knitr` kann man R-Code-Sequenzen in ein L^AT_EX-Dokument einbetten. Das Prinzip ist bei beiden Paketen das Gleiche. Die R-Code-Sequenzen werden jeweils in eine spezielle Umgebung, Chunks genannt, eingefügt gemäß

```
<< Chunk-Optionen >>=
# R-Code
@
```

Die Chunk-Kopfzeile muss in eine Zeile ohne Zeilenumbruch geschrieben werden. Zusätzlich zu den Grafik-Parametern von R gibt es jetzt auch Chunk-Optionen (Liste von »Option=Wert«-Paaren) mit denen man beeinflussen kann, ob der R-Code nur ausgeführt wird (`eval=TRUE`) oder ob der Code und die Ergebnisse auch im Dokument erscheinen sollen (`echo=TRUE`). Pro Chunk können ein Label (`label`) vergeben und weitere Chunk-Optionen gesetzt werden. Je nach Art der Chunk-Option kommen als Werte in Frage: `TRUE/FALSE`, ein String oder ein Zahlenwert. Ein String ist eine Zeichenfolge zwischen Einfach- oder Doppelapostrophen. Die L^AT_EX-Quelldatei mit den Chunks erhält die Dateierweiterung `.Rnw` (R-Noweb-Datei). Die Bearbeitung kann interaktiv oder im Terminal erfolgen. In jedem Fall erhält man als Ergebnis eine T_EX-Datei mit der Endung `.tex`, die man in einem zweiten Schritt wie üblich mit `lualatex` bearbeitet.

Das Paket Sweave

Das Paket `Sweave` wurde 2002 von Friedrich Leisch entwickelt [4] und ist integraler Bestandteil jeder R-Distribution. Die Dokumentation zu `Sweave`, das `Sweave-Manual`, ist z. B. unter [3] zu finden. `Sweave` ist eine L^AT_EX-Stildatei, die mit `\usepackage{Sweave}` in der Präambel des Dokuments geladen werden muss.

Im folgenden ist ein Template für eine minimale Quelldatei `file.Rnw` zur Bearbeitung mit `Sweave` und `lualatex` zu sehen:

```
\documentclass{article}
\usepackage{Sweave}
\begin{document}
Text
<<>>=
# R-Code
@
Text
...
\end{document}
```

Bearbeitung interaktiv in einer R-Sitzung:

```
$ R
> Sweave("file.Rnw")
> q()
$
```

Bearbeitung auf der Kommandozeile im Terminal:

```
Rscript -e "Sweave('file.Rnw')"
```

Als Ergebnis erhält man in beiden Fällen im Arbeitsverzeichnis eine T_EX-Datei `file.tex`, die wie üblich mit `lualatex` bearbeitet wird. Eventuell sind mehrere Läufe erforderlich, um Querverweise und Inhaltsverzeichnis zu aktualisieren, Literatur und Index einzubinden, etc. Das Endergebnis ist eine Datei `file.pdf`, die das vollständig formatierte Dokument enthält.

Die von R erzeugten Grafiken stehen auch als einzelne pdf-Dateien `file-xxx.pdf|eps` abhängig von der Art der Bearbeitung (interaktiv oder im Terminal) im Arbeitsverzeichnis bzw. im Unterverzeichnis `figure/` zur Verfügung.

Wenn ein Chunk eine Grafik erzeugen soll, muss die Chunk-Option `fig=TRUE` gesetzt werden. Eine Abbildung kann zentriert werden, indem man den Chunk in eine `center`-Umgebung einschließt.

Innerhalb eines Chunks, der eine Grafik erzeugt, verwendet Sweave die gleichen Grafik-Parameter wie R zur detaillierten Gestaltung der Diagramme. Eine globale Einstellung dieser Parameter mit `par()`, gültig für alle Grafiken, ist jedoch in Sweave nicht möglich, da jeder Chunk mit Grafik ein Device öffnet und nach der Ausgabe der Grafik wieder schließt, wobei die Parameter gelöscht werden. Die Parameter müssen daher in jedem Chunk, der Grafik erzeugt, erneut angegeben werden.

Weitere wichtige Chunk-Optionen von Sweave sind im Folgenden tabellarisch zusammengestellt. Im Allgemeinen kommt man mit den voreingestellten Werten zurecht, sodass keine expliziten Angaben erforderlich sind, lediglich die Option `fig=TRUE` muss bei allen Chunks, die Grafik erzeugen sollen, gesetzt werden. Sweave erzeugt standardmäßig pdf-Grafiken.

Option	Beschreibung
<code>echo=</code>	Ausgabe des R-Codes in das Dokument, default TRUE
<code>eval=</code>	Auswertung des R-Codes, default TRUE
<code>pdf=</code>	Ausgabe der Grafikdatei im pdf-Format, default TRUE
<code>eps=</code>	Ausgabe der Grafikdatei im eps-Format, default FALSE
<code>width=</code>	Breite der Grafik in Inch, default 6
<code>height=</code>	Höhe der Grafik in Inch, default 6
<code>fig=</code>	Wenn der Chunk eine Grafik erzeugen soll, muss die Option <code>fig=TRUE</code> gesetzt werden, default FALSE.
<code>label=</code>	Jeden Chunk kann man benennen, z.B. <code>label=hello</code> oder einfach <code><<hello, ...>>=</code> , aber ausnahmsweise nur, wenn das Label an erster Position steht. Alle Label müssen eindeutig sein.

Chunk-Optionen kann man auch *global* in der Präambel setzen mit

```
\SweaveOpts{opt1=value1, ...}
```

Die eingestellten Werte gelten dann für alle Chunks.

Mit `width` und `height` kann man nur die Größe der ausgegebenen pdf-Grafiken festlegen, nicht jedoch die Größe der Grafiken im Dokument. Deren Breite ist auf `0.8\textwidth` voreingestellt. Eine Änderung ist nach `\begin{document}` mit der Anweisung

```
\setkeys{Gin}{width=0.8\textwidth}
```

möglich. Hier ist `width` eine Option für den `\includegraphics`-Befehl.

Dateiorganisation:

Die Stildatei `Sweave.sty` befindet sich auf dem Mac in

`/Library/Frameworks/R.framework/Resources/share/texmf/tex/latex/Sweave.sty`

Da sie dort von L^AT_EX nicht gefunden wird, habe ich sie auch in mein `texmf`-Directory `~/Library/texmf/tex/latex/` kopiert.

Anmerkungen:

1. Vorsicht, Sweave beachtet bei der Bearbeitung alle Chunk-Indikatoren im Dokument, selbst innerhalb von `verbatim`-Umgebungen!

2. In L^AT_EX wird ein Dokument oft in mehrere Teile aufgeteilt, die dann mit `\input{}` in das Hauptdokument eingefügt werden. Dabei ist aber zu beachten, dass `\input{}` ein L^AT_EX-Befehl ist, der beim Einlesen das Teildokument entsprechend der L^AT_EX-Syntax interpretiert und die Sweave-Chunks ignoriert. Teildokumente, die Sweave-Chunks enthalten, müssen mit `\SweaveInput{}` eingelesen werden. Die Teildokumente müssen die gleiche Sweave-Syntax und die gleiche Kodierung wie das Hauptdokument verwenden.
3. Beim Schreiben einer Sweave-Quelldatei `file.Rnw` muss man darauf achten, dass der Computer (Betriebssystem), der Editor und das Dokument die gleiche Kodierung verwenden. Das gilt insbesondere, wenn auch extern erstellte Dateien verwendet werden. Standard ist heute die UTF-8-Kodierung. Das Problem besteht selbst dann, wenn man sich strikt an den ASCII-Zeichensatz hält, weil z. B. Anführungszeichen unterschiedlich kodiert werden.

Das Paket knitr

Das Paket `knitr` stammt von Yihui Xie (2013) und ist eine Weiterentwicklung des Einbettungsproblems inspiriert von Sweave. Es bietet erweiterte grafische Darstellungsmöglichkeiten und mehr Bedienungskomfort [9, 11].

`knitr` ist keine L^AT_EX-Stildatei, sondern eine Bibliothek auf CRAN. Das Paket muss von dort einmalig in einer R-Sitzung installiert werden mit

```
install.packages('knitr', dependencies=TRUE)
```

Die Installation erfolgt interaktiv. Danach sollte man in der gleichen R-Sitzung noch mit `library(knitr)` testen, ob `knitr` korrekt installiert ist. Dies ist der Fall, wenn dabei keine Fehlermeldung auftritt.

Auf dem Mac befinden sich danach `knitr` und die Dokumentation in

```
/Library/Frameworks/R.framework/Resources/library/knitr/bin/knit  
/Library/Frameworks/R.framework/Resources/library/knitr/doc/...
```

Dort findet man `knitr-refcard.pdf` [10] und `knitr-intro.html`. Siehe auch [10, 11, 12] für diese und weitere Manuals und man-pages.

Da wir Sweave bereits kennengelernt haben, ist der Umstieg auf `knitr` nicht schwierig. Man muss nur alles Sweave-spezifische entfernen, nämlich

- `\usepackage{Sweave}`
- `\SweaveOpts{...}`
- `\setkeys{Gin}{...}`
- eigene Sweave.sty-Anweisungen
- in Grafik-Chunks `fig=TRUE`, `eps=TRUE`

- L^AT_EX-Umgebungen um Grafik-Chunks herum, wie center, figure inklusive caption
- \SweaveInput{}

knitr verwendet einen Großteil der grafischen Parameter von R, kennt aber z. T. auch andere und vor allem wesentlich mehr Chunk-Optionen, siehe <https://yihui.org/knitr/options/> [13]. Unter <https://yihui.org/knitr/> findet man auch einen Übersichtsartikel mit vielen Verweisen auf das knitr-Buch, Manuals, Demos/Beispiele, FAQ's, Options, hooks, u.v.a.m.

Ein Teildokument, z. B. vorwort.Rnw wird in knitr mit der Chunk-Option child= als Chunk in das Hauptdokument file.Rnw eingefügt gemäß

```
<<preface,child='vorwort.Rnw'>>=
@
```

Die Bearbeitung einer Quelldatei file.Rnw erfolgt in knitr mit der Funktion knit() interaktiv in einer R-Sitzung oder auf der Kommandozeile im Terminal (purl() extrahiert nur den R-Code, ähnlich wie Stangle()).

interaktiv:

```
library(knitr)
knit('file.Rnw')
```

Terminal:

```
Rscript -e "library(knitr); knit('file.Rnw')"
```

In beiden Fällen erhält man eine T_EX-Datei file.tex, die man wie üblich mit lualatex bearbeiten muss. Grafiken werden auch als einzelne pdf-Dateien im Unterverzeichnis figure/ abgelegt.

Template für die Quelldatei file.Rnw zur Bearbeitung mit knitr und lualatex:

```
\documentclass{article}
\begin{document}
Text
% Setup:
<<setup,include=FALSE>>=
knitr::opts_chunk$set(comment='',fig.width=3.5,fig.height=3.5,
fig.align='center',out.width='0.6\textwidth',
out.height='0.6\textwidth',highlight=FALSE,
background='#F7F7F7',size='normalsize')
knitr::opts_knit@set(global.par=TRUE)
@
```

```
<< knitr-Chunk-Optionen >>=
#   R-Code
@
Text
...
\end{document}
```

Zum praktischen Arbeiten mit knitr hat Josh Davis [1] einen sehr lesenswerten Artikel geschrieben. »Als Statistiker bin ich es gewohnt, mit R und L^AT_EX zu arbeiten. Aber ich suchte nach einer einfacheren Möglichkeit, meine Berichte zu schreiben, stieß auf knitr und war auf Anhieb restlos begeistert.« Auch dieser Beitrag ist mit knitr geschrieben und wir sind der gleichen Meinung. Deshalb wollen wir noch etwas genauer auf die wichtigsten Eigenschaften und Optionen von knitr eingehen.

Die Chunk-Optionen von knitr stehen wie bei Sweave in der Chunk-Kopfzeile

```
<<Chunk-Optionen>>=
```

als Liste von »Option=Wert«-Paaren, die in einer Zeile ohne Zeilenumbruch geschrieben werden muss, z. B.

```
<<abb1,fig.lp='f1:',fig.cap='Boxplot: Gewicht der Küken'>>=
```

Option	Beschreibung
abb1	ist ein Label an erster Position ohne label=
fig.lp=	ist ein Label-Prefix zur Referenzierung der Grafik im Text mit <code>\ref{f1:abb1}</code> . (Das entsprechende L ^A T _E X-Label wird automatisch generiert.)
fig.cap=	ist eine Bildunterschrift

Chunk-Optionen kann man auch global festlegen in einem ersten Setup-Chunk, wie z. B. in dem obigen Template angegeben.

Option	Beschreibung
setup	ist ein Label
include=FALSE	Chunk erscheint nicht im Dokument, wird aber ausgeführt.
knitr::opts_chunk\$set()	Funktion zum globalen Setzen von Chunk-Optionen:
comment=''	Kommentare werden nicht ausgegeben
fig.align=	Grafik im Dokument horizontal ausrichten. Werte: 'default', 'center', 'left', 'right'

<code>fig.pos=</code>	Positionierung der Grafik im Dokument. Werte: 'h', 't', 'b', 'p', '!', 'H', entspricht in L ^A T _E X der figure-environment-Option. Die Option 'H' erfordert das Paket float.
Die Parameter <code>width</code> und <code>height</code> gibt es in knitr nicht, stattdessen:	
<code>fig.width=4</code>	Breite und Höhe in Inch der pdf-Grafiken
<code>fig.height=4</code>	
<code>out.width='10cm'</code>	Breite und Höhe der Grafiken im Dokument
<code>out.height='10cm'</code>	
<code>highlight=</code>	Syntax-Highlighting, default TRUE. Für s/w-Druck FALSE setzen
<code>background='#F7F7F7'</code>	Hintergrundfarbe für die Ausgabe des R-Codes (sehr helles grau), hexadezimal, rgb-Farbmodell im Bereich 00 bis FF
<code>size='normalsize'</code>	Schriftgröße für die Ausgabe des R-Codes gemäß dem Wert in der Class-Option im Hauptdokument.
<code>knitr::opts_knit@set()</code>	Funktion zum globalen Setzen von knitr-Paketoptionen
<code>global.par=</code>	Wenn TRUE, bleiben die mit <code>par()</code> gesetzten R-Grafikparameter erhalten, default FALSE.

Mit den Parametern für die Abmessungen der Grafik kann man gegebenenfalls auch Verlagsvorgaben berücksichtigen. Der default-Font für die Grafikbeschriftung ist Helvetica. Die Fontgröße kann mit dem R-Grafik-Parameter `cex` beeinflusst werden. Dieser Beitrag ist mit knitr geschrieben und möge als Anschauungsbeispiel dienen.

`tikzDevice` zur Beschriftung von R-Grafiken mit L^AT_EX

Texte, Ziffern und mathematische Formeln in Grafiken sollten wie im Hauptdokument im L^AT_EX-Text- bzw. Mathematikmodus formatiert werden, damit die Fonts im Dokument und in den Grafiken übereinstimmen. Dies ermöglicht das Paket `tikzDevice` [6, 7].

Die Library `tikzDevice` schafft eine Verbindung zwischen R und dem Grafikprogramm TikZ.

Lösung mit L^AT_EX

Beispiel:

Wir schreiben zuerst ein R-Programm, z. B. `simpler.R` der Form

```
library(tikzDevice)
tikz('simplexpl.tex',width=3.5,height=3.5)
plot(1,1)
text(1,0.8,"$\sigma^2=\frac{1}{n-1}\sum_{i=1}^n(x_i-\bar{x})^2$")
dev.off()
```

Die Funktion `tikz()` aus dem `tikzDevice`-Paket öffnet ein Grafikfenster mit den angegebenen Maßen (`width` und `height` sind `tikz`-Parameter, Vielfache von 1 in), interpretiert die nachfolgenden R-Anweisungen und L^AT_EX-Texte als `tikz`-Anweisungen in einer `tikzpicture`-Umgebung und schreibt den erzeugten T_EX-Code in die Ausgabedatei `simplexpl.tex`. Dazu bearbeiten wir in einem ersten Schritt die R-Datei mit

```
Rscript -e "source('simpler.R')"
```

und erhalten die T_EX-Datei `simplexpl.tex`.

In einem zweiten Schritt binden wir diese Datei mit `\input{}` in eine dummy-L^AT_EX-Datei `simplifiedummy.tex` ein, laden das Paket `tikz` und bearbeiten die Datei wie üblich mit `lualatex`.

```
\documentclass{article}
\usepackage{tikz}
\parindent0pt
\pagestyle{empty}
\begin{document}
\input{simplexpl}
\end{document}
```

Als Ergebnis erhalten wir die Datei `simplifiedummy.pdf`. Das Diagramm zeigt die Abbildung 2.

Lösung mit Sweave

Unser Ziel ist es natürlich, diesen Mechanismus in ein einziges *.Rnw-Dokument zu integrieren und mit Sweave zu bearbeiten.

Eine mögliche Lösung mit Sweave verwendet den `tikz`-Parameter `console=TRUE` anstelle der expliziten Angabe einer Ausgabedatei [2]. Dadurch wird die Ausgabe auf STDOUT geschrieben und von Sweave direkt wieder eingelesen. Dies ergibt folgende Programmstruktur für die Datei `mytikz.Rnw`.

Beispiel 1 mit Sweave und `lualatex`.

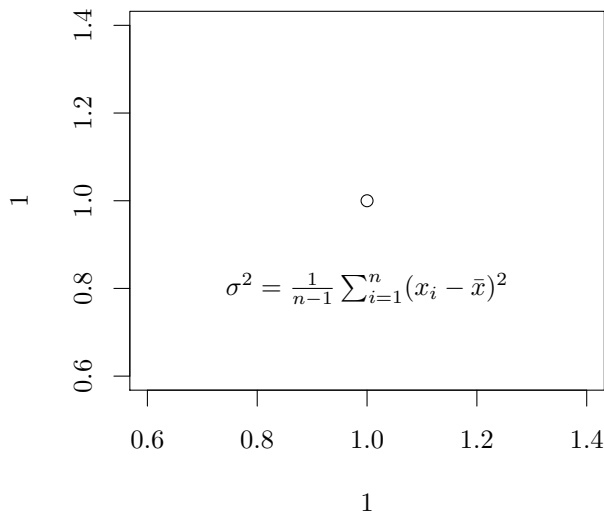


Abb. 2: Grafik mit tikzDevice.

```

\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage{Sweave}
\usepackage{tikz}
\parindent0pt
\pagestyle{empty}
\begin{document}
<<echo=FALSE,results=tex>>=
library(tikzDevice)
tikz(console=TRUE,width=3.5,height=3.5)
Chol <- read.table("chol.dat")
alter <- Chol[,1]
chol <- Chol[,2]
plot(alter,chol,xlab="Alter",ylab="Cholesterin")
LM <- lm(chol ~ alter)
abline(LM)
mtext(text="$\\hat{\\beta}=(X^{\\prime}X)^{-1}X^{\\prime}y$",side=3,line=1)
tmp <- dev.off()
@
end{document}

```

Bearbeiten mit Rscript -e "Sweave{'mytikz.Rnw'}", lualatex mytikz und pdfcrop mytikz.pdf. Das Ergebnis zeigt die Abbildung 3.

$$\hat{\beta} = (X'X)^{-1}X'y$$

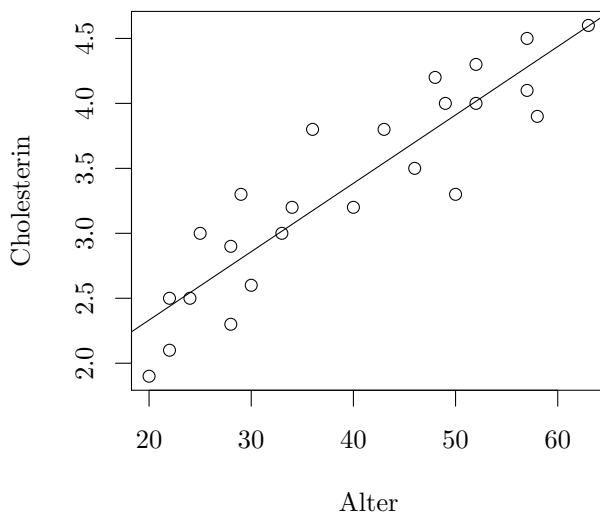


Abb. 3: Grafik mit tikzDevice und Sweave.

Lösung mit knitr

Zur Bearbeitung mit knitr ändert sich nur die Chunk-Option results=, die in knitr andere Werte hat, nämlich results='asis'. Außerdem darf in der L^AT_EX-Präambel das Paket Sweave nicht geladen werden.

Beispiel 2 mit knitr.

Unsere Beispieldatei mytikz-knitr.Rnw lautet dann wie folgt.

```
\documentclass[ngerman]{dtk}
\usepackage{tikz}
\parindent0pt
\begin{document}
\begin{figure}[htb]
\begin{center}
\footnotesize\fontspec{TeX Gyre Heros}[Scale=MatchLowercase]
<<plot2,fig.lp='p2:',echo=FALSE,results='asis'>>=
```

```

library(tikzDevice)
tikz(console=TRUE,width=3.5,height=3.5)
Chol <- read.table("chol.dat")
alter <- Chol[,1]
chol <- Chol[,2]
plot(alter,chol,xlab="Alter",ylab="Cholesterin")
LM <- lm(chol ~ alter)
abline(LM)
mtext(text='$\\hat{\\beta}=(X^{\\prime}X)^{-1}X^{\\prime}y$',side=3,line=1)
tmp <- dev.off()
@
{\\normalfont\\normalsize
\\caption{Grafik mit \\texttt{tikzDevice} und \\texttt{knitr}.\\label{bild}}
\\end{center}
\\end{figure}
\\end{document}

```

Bearbeiten mit `Rscript -e "library(knitr); knitr('mytikz-knitr.Rnw')"` und `lualatex mytikz-knitr`. Das Ergebnis zeigt die Abbildung 4.

Der bevorzugte Font zur Beschriftung von Grafiken ist Helvetica mit einem um 2pt kleineren Schriftgrad, deshalb wurde die Schrift lokal vor dem Chunk entsprechend umgeschaltet. Außerdem wurde der Chunk in ein `figure`-Environment eingefügt, mit der Option `'htb'` positioniert, zentriert und mit einer Caption und einem Label versehen.

Die typographische Perfektion hat aber auch ihren Preis. Mehrere nützliche Eigenschaften von `knitr` sind verloren gegangen.

1. Die Ausgabe der Grafik als externe pdf-Datei fehlt.
2. Die Ausgabe des R-Code im Dokument fehlt.
3. Die Optionen `crop`, `fig.cap`, `label`, `fig.lp`, `fig.align`, `fig.pos` sind wirkungslos. Diese können aber, wie gezeigt, ersatzweise auch in L^AT_EX formuliert werden.

Vielleicht kann man die gezeigte Lösung diesbezüglich noch verbessern. Für Hinweise wäre ich sehr dankbar.

Literatur

- [1] Josh Davis: A Beginners Tutorial for knitr, <https://joshldavis.com/2014/04/12/beginners-tutorial-for-knitr/>.
- [2] Frederike Fuhlbrück: Sweave: Grafiken mit tikzDevice, TikZ und PGF, <https://k-baeumchen.fuhlbrueck.net/R-und-LaTeX.html>.

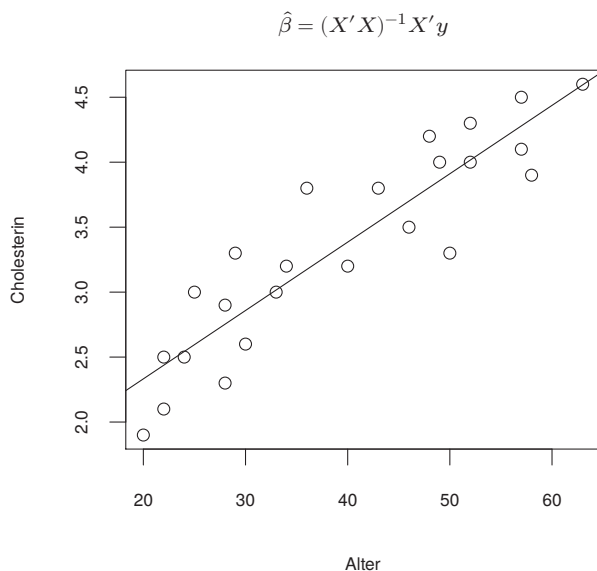


Abb. 4: Grafik mit tikzDevice und knitr.

- [3] Friedrich Leisch u. a.: Sweave User Manual, <https://stat.ethz.ch/R-manual/R-devel/library/utils/doc/Sweave.pdf>, 2020.
- [4] Friedrich Leisch: Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis, COMSTAT 2002 Proceedings Computational Statistics, 69, pp. 575–580, Physica Verlag, Heidelberg.
- [5] Arthur Linder: Statistische Methoden, Birkhäuser Verlag, Basel, 1960.
- [6] Charlie Sharpsteen, Cameron Bracken: TikZ-Device: L^AT_EX Graphics for R, <https://github.com/yihui/tikzDevice>.
- [7] The R-project.org Team: tikzDevice: man pages, <https://cran.r-project.org/web/packages/tikzDevice/index.html>.
- [8] W. N. Venables, D. M. Smith: An Introduction to R, <https://r-project.org/Documentation> → R-intro.pdf.
- [9] Yihui Xie: Dynamic Documents with R and knitr, Chapman & Hall, 2013.

- [10] Yihui Xie: knitr Reference Card, <https://yihui.org/knitr/>, 2013.
- [11] — knitr: A General-Purpose Tool for Dynamic Report Generation in R, <https://yihui.org/knitr/demo/manual/knitr-manual.pdf>, 2013.
- [12] — knitr: Graphics Manual, <https://yihui.org/knitr/demo/manual/knitr-graphics-manual.pdf>, 2013.
- [13] — knitr: Options, <https://yihui.org/knitr/options>, 2013.

Von fremden Bühnen

Der erweiterte Orbit

Henning Hraban Ramm

Das 15. ConT_EXt-Meeting stand unter dem Motto »expanding orbits« und fand vom 20. bis 25. September 2021 im »Schlösschen« ('s Sjetootje) von Bassenge-Boirs (Belgien) statt, einem Dorf zwischen Maastricht und Liège.

On- und offline

Dies war das erste ConT_EXt-Meeting, das »hybrid« vor Ort und online stattfand. Eigentlich wollte jemand das Streaming der Vorträge über seinen geschäftlichen Zoom-Zugang abwickeln, wie er das schon im Vorjahr für die zwei durch Reisebeschränkungen verhinderten Vorstandsmitglieder gemacht hatte. Vor Ort fiel ihm dann aber ein, dass er diesen Account gar nicht mehr nutzen konnte. Da ich im Vorfeld vorgeschlagen hatte, senfcall.de (ein Angebot des Computerwerk Darmstadt e. V.) zu nutzen, hatte ich spontan eine zusätzliche Aufgabe.

Harald König richtete seinen Bastel-Raspi als dauerhaften Kamera- und Mikrofon-Account ein, und für die externen Referenten schalteten wir Lautsprecher ein. Zum Glück funktionierte das BigBlueButton-System sehr zuverlässig; Probleme machte dagegen der unzuverlässige DHCP-Server vor Ort, nicht jedes Bildschirmfenster ließ sich zur Präsentation freigeben, und der Raspi war vielleicht doch etwas zu schwach. Insgesamt hat sich das Konzept aber bewährt. Bis zu 14 externe und 11 körperlich anwesende Teilnehmer nutzten die Gelegenheit, den wie üblich vielfältigen und interessanten Vorträgen zu lauschen und sich über T_EX und die Welt auszutauschen.

Ideen und Pläne

Und wie immer wurden Ideen entwickelt: Ein ConT_EXt-Kochbuch soll als Corporate-Identity-Handbuch der fiktiven »Aardvark Airlines« entstehen (wobei unser Präsident Pavneet Arora nicht an den Kampffjet F-111 alias Aardvark dachte). Zu Ehren von Donald Knuth kann man Maße in ConT_EXt jetzt auch in Potrziebie angeben (1 dk = 2,263348517438173216473 mm). Die *Context Group* möchte mit Entwicklern von



Abb. 1: Gruppenbild der körperlich Anwesenden.¹

Open-Source-PDF-Viewern zusammenarbeiten, um einerseits PDF 2.0, andererseits diverse fehlende Features zu unterstützen.

Zum letzten Punkt: Ich hatte alle PDF-Viewer für Linux und MacOS getestet, die ich finden (bzw. installieren) konnte und dabei festgestellt, dass die meisten Open-Source-Projekte sehr zu wünschen übrig lassen. Die meisten Programme können noch nicht einmal die erweiterten Eigenschaften anzeigen, in denen die erzeugende $\text{\TeX}$ -Version hinterlegt ist. Natürlich kann kein Programm die unendlichen Weiten der PDF-Standards vollständig abbilden, aber es scheint beispielsweise kein Open-Source-Programm für Linux zu geben, mit dem man PDF-Formulare ausfüllen und ausdrucken kann, von Präsentationen mit JavaScript-Aktionen ganz zu schweigen. Für eine professionelle Druckausgabe ist es wichtig, CropBox, BleedBox und TrimBox anzeigen und ändern zu können; das geht nur mit den kommerziellen Programmen Acrobat Pro oder PDF Studio Pro – eine freie Alternative habe ich nicht gefunden. Die kommerziellen PDF-Editoren, die auch für Linux verfügbar sind (PDF Master Editor und PDF Studio) und in ihren Demoversionen als Viewer brauchbar wären, haben dann wieder keinen Präsentationsmodus.

Ähnlich wie OpenType-Schriften gehören PDF-Viewer heute zum unverzichtbaren Werkzeug zur Arbeit mit $\text{\TeX}$, ihre Entwicklung darf daher nicht stehen bleiben. Der Prager Informatikstudent Michal Vlasák schrieb seine Bachelorarbeit² über die verschiedenen Möglichkeiten der Einbindung von Multimedia-Daten (Audio, Video und 3D) in PDF, die Unterstützung der Anzeigeprogramme dafür und die Ansteuerung mit $\text{\TeX}$. Eine Kurzfassung hat er uns vorgestellt und bereits vorab im



¹ Fotos: Harald König, Hraban Ramm und Frans Goddijn.

² <https://dspace.cvut.cz/handle/10467/95065> (tschechisch)



Abb. 2: Diskussion in einer Pause.

aktuellen *CG Journal* veröffentlicht. Für OpTeX und LaTeX hat er das Paket pdfextra erstellt. Im Wesentlichen diskutierten Hans und Taco mit ihm, wie ConTeXt die Multimedia-Einbindung in Zukunft behandeln soll. Da Michal nicht zuletzt beim Debugging von Lua(Meta)TeX bewiesen hat, dass er programmieren kann, sehen wir Chancen, wertvollen Code zu Viewer-Projekten beisteuern zu können.

Information Overload

Mehrere Beiträge berichteten vom ConTeXt -Einsatz in der Praxis und dem dabei Gelernten, sei es bei einem Finanzdienstleister in Singapur (Ramkumar K. B.), bei einem Filmfestival in Tschechien (Jano Kula), im akademischen Fundraising in Kanada (Aditya Mahajan) oder in meinem Kleinverlag.



Abb. 3: Konzentration bei den Vorträgen.

Ansonsten erfuhren wir mehr, als irgendjemand wirklich aufnehmen könnte, über die internen Mechanismen von ConTeXt und MetaPost, über Lua, LPEG und XML, Schriften und ein paar Themen, die nur sehr entfernt mit ConTeXt zu tun hatten. Den aktuellen Stand der Entwicklung von LuaMetaTeX stelle ich in einem eigenen Artikel vor.

Den größten Teil der Vorträge bestritt wie immer Hans Hagen; auch Taco Hoekwater, Wolfgang Schuster, Tomáš Hála, Willi Egger und ich waren mit mehr als einem Beitrag vertreten.

Die obligatorische Exkursion führte uns schließlich in die Sternwarte von Heerlen, wo diejenigen, die niederländisch verstanden, etwas über unser Sonnensystem erfahren konnten. Der Referent machte allerdings einen etwas konfuse Eindruck und brachte wohl auch das eine oder andere durcheinander. Die Infotafeln ließen typografisch sehr zu wünschen übrig – mit TeX wäre das nicht passiert –, was aber nichts über ihren Inhalt aussagen soll. Draußen konnten wir dann unter Anleitung von Hobbyastronomen trotz starker Bewölkung den Jupiter, seine Monde und einige Sternbilder erkennen, mit und ohne Teleskop.



Abb. 4: Mit TeX wäre das nicht passiert.



Abb. 5: Hans Hagen im Erklärmodus.

Nötige Formalitäten

Bei der Mitgliederversammlung der *Context Group* wurde die Satzung vereinfacht. Der Vorstand besteht nun nach kleinen Veränderungen aus Pavneet Arora (Kanada), Mojca Miklavc (Slowenien), Arthur Rosendahl (Schweden), Alan Braslau (USA),

Taco Hoekwater (Niederlande), Peter Münster (Frankreich) und Henning Hraban Ramm (Deutschland).

Bis bald!

Das nächste Con_TE_Xt-Meeting³ wird in der Woche 12.–18. 9. 2022 in Dreifelden im Westerwald stattfinden, wobei ein Teil der Themen schon absehbar ist. Die Anmeldung wird ab Mai möglich sein.

Ein neuer Motor für Con_TE_Xt

Henning Hraban Ramm

Seit über drei Jahren konzentriert sich die Con_TE_Xt-Entwicklung auf die neue »Engine« LuaMeta_TE_X.

Zeitplan

Vor fast vier Jahren entstand die Idee zu LuaMeta_TE_X und fand breite Zustimmung unter den Con_TE_Xt-Anwendern, denen sie beim Con_TE_Xt Meeting 2018 präsentiert wurde. Anfang 2019 stellte Hans Hagen die erste Testversion vor, und beim Meeting 2019 die erste offizielle Version.

Schon beim Meeting 2020 hatte er mehr oder weniger erreicht, was er sich vorgestellt hatte, und seit Herbst 2021 kann der Code als stabil gelten. Der offizielle »Umzug« von der Lua_TE_X-basierten Version MkIV zu LMTX mit der Veröffentlichung des Quellcodes ist für 2022 geplant, obwohl schon viele Anwender produktiv mit LuaMeta_TE_X arbeiten.

Das Konzept

In LuaMeta_TE_X werden Satzmaschine (_TE_X) und Grafikroutinen (MetaPost) von Lua-Code zusammengehalten und gesteuert.

_TE_X wird dabei einerseits auf die Knuthsche Satzmaschine reduziert, ohne all die Anbauten für Dateiformate, Schriften und Grafik, andererseits aber zeitgemäßer und flexibler umgesetzt, ohne Rücksicht auf die technischen Beschränkungen der

³ <https://meeting.contextgarden.net>



1980er Jahre. Aus MetaPost wurden die veraltete Schriftverarbeitung sowie Ein- und Ausgabefunktionen entfernt.

Die Verarbeitung von Dateien, Schriften, Grafiken und Farben ist schon in ConT_EXt MkIV komplett in Lua geschrieben, so dass große Teile von LuaT_EX gar nicht mehr verwendet werden. Diesen Weg geht LuaMetaT_EX konsequent weiter.

In der Abkürzung LMTX steht das X für XML, das als Eingabeformat für ConT_EXt eine bedeutende Rolle spielt.

Einzelheiten

LuaMetaT_EX ist gleichzeitig ein vollwertiger Lua-Interpreter (5.4), erweitert um ein paar praktische Bibliotheken (z. B. `lpeg` und `pplib`), aber ohne JIT (*Just In Time*) und FFI (*Foreign Function Interface*). Die Maschine soll so schlank wie möglich sein und kommt ohne Abhängigkeiten aus. (Die wenigen notwendigen Bibliotheken sind klein, alt und stabil, z. B. `zlib`.) Der vollständige Code wird mit der Distribution verteilt werden, so dass sich das System wahrscheinlich auch auf zukünftigen Systemarchitekturen compilieren lässt.

Der CWEB-Code von LuaT_EX wurde zunächst in C-Code konvertiert, der ohne komplexe Infrastruktur auskommt. Artefakte von Pascal und Web wurden entfernt. Die Lesbarkeit des Codes wurde damit deutlich verbessert. Mit Hilfe der Compile-Farm wird laufend getestet, ob sich der Code auf allen Plattformen übersetzen lässt. Dass das schnell und einfach möglich ist, ist für das Projekt essenziell.

Alle betriebssystemabhängigen Funktionen wurden abstrahiert. Alle Dateioperationen laufen über Lua, einschließlich der T_EX-Primitiven, die sich auf das Lesen und Schreiben beziehen. Das Gleiche gilt für die Konsolenausgabe.

Die Speicherverwaltung wurde stark verändert – selbst Kleinstrechner haben heute mehr Speicher zur Verfügung als die Computer von damals, wir rechnen nicht mehr mit 8 bit (oder sogar weniger), sondern mit 64 bit, so dass viele Tricks zum Speichersparen heute keinen Sinn mehr ergeben. Das Programm ist trotzdem kleiner als 3 MB und läuft auch auf einem Raspberry Pi 4 recht ordentlich.

Sprachen, Schriften, Marken usw. sind nicht mehr Register-basiert. Schriftinformationen werden nicht mehr im String-Pool abgelegt, was viel ausmacht.

Der weitgehend ungenutzte Code für Richtungen wurde aus dem Kern entfernt und der Rest ins Backend verlagert.

Die Einstellungen für Sprachen wurden effizienter und flexibler implementiert; es wird nur noch wenig in der Formatdatei festgelegt. Die Formatdatei ist damit auch kleiner geworden, obwohl sie nicht mehr komprimiert wird. Das Erstellen des Formats ist stabiler und schneller geworden.

Manche Mathe-Konzepte wurden erweitert, so dass es mehr Kontrolle über die Darstellung gibt. (In diesem Bereich ist im Original-T_EX sehr viel hartcodiert, was dann nicht zu allen Schriften passt.)

T_EX braucht wenig über Schriften zu »wissen« und wird bei Bedarf mit virtuellen Fonts »gefüttert«, die ihren Namen jetzt wirklich verdienen, denn sie werden nur bei Bedarf im Speicher erzeugt.

Diese Darstellung ist natürlich unvollständig – wer sich für die Einzelheiten interessiert, möge sich die Dokumentation in der ConT_EXt-Distribution ansehen.

Makros

Es gibt ein paar Erweiterungen für die Behandlung von Makro-Argumenten und zusätzliche Verzweigungsbefehle, was eine deutlich elegantere Makroprogrammierung ermöglicht.

Eine größere konzeptuelle Änderung sind die Schutzmechanismen: Makros können nicht nur `\protected`, sondern auch `\frozen` sein. Damit werden alle Primitiven und viele grundlegende Makros vor dem versehentlichen Überschreiben geschützt.

Die Versionen `\long` und `\outer` sind dafür verschwunden – was keine große Änderung bedeutet, denn in ConT_EXt waren Makros schon immer `\long` und niemals `\outer`, und schon in MkII waren die wenigsten Befehle expandierbar.

In Nodes (vom Zeichen bis zum Absatz) sind mehr Einzelheiten gespeichert und zugänglich, was auch dem Debugging zugute kommt.

Und wie geht es weiter mit LuaT_EX?

Natürlich wird LuaT_EX weiterhin gepflegt. Schließlich setzt auch ConT_EXt MkIV darauf auf, und es ist die Referenz für die Ausgabe von LuaMetaT_EX. (L^AT_EX setzt mit LuaHB_TE_X auf einen Ableger mit eingebauter Schriftverarbeitung.)

Vor allem die LuaT_EX-Anwender außerhalb von ConT_EXt drängten auf eine stabile Version, es sollte keine Erweiterungen und vor allem keine Änderungen an den Schnittstellen mehr geben.

Hans war zunehmend frustriert von negativen Kommentaren: LuaT_EX habe zu viele Bugs oder sei zu langsam und werde nie in den Distributionen landen, das Handbuch sei zu schlecht, oder auch, das Programm würde in einer kommerziellen Umgebung entwickelt und finanziert. Diese Kommentare sagen wahrscheinlich mehr über ihre Verfasser und ihre Einstellung als über LuaT_EX.

Auf der anderen Seite erliegen auch Nicht-ConT_EXt-Anwender dem Charme von LuaT_EX, und je mehr sie programmieren, desto wichtiger ist es, die Codebasis einzufrieren.

Weil LuaTeX bereits weit verbreitet und fest in TeXLive integriert ist, kann nicht mehr viel geändert werden. Es wäre schlechte Werbung für TeX, wenn Low-level-Anwender mit konzeptuellen Änderungen konfrontiert würden.

Für ConTeXt soll die Entwicklung aber nicht stehen bleiben.

Während möglicherweise einmal eine Korrektur oder Erweiterung ihren Weg zu LuaTeX findet, werden die meisten Änderungen in LuaMetaTeX nicht portiert werden; schon jetzt ist die Codebasis zu unterschiedlich. Dafür können die LuaTeX-Benutzer ziemlich sicher sein, dass sich keine Fehler mehr einschleichen werden.

L^AT_EX News – Issue 33, Juni 2021¹

Frank Mittelbach, L^AT_EX Project

Einleitung

Der Fokus des L^AT_EX-Releases vom Juni 2021² liegt darin, weitere wichtige Bausteine für die zukünftige Produktion einer zuverlässigen Tagged-PDF-Ausgabe zu liefern (siehe [3]) – diese Verbesserungen werden in den nächsten zwei Abschnitten besprochen.

In den dann folgenden Abschnitten wird eine Reihe aktueller kleinerer Verbesserungen und Korrekturen beschrieben. Wie üblich finden Sie mehr Details zu den einzelnen Änderungen in der Datei `changes.txt` der Distribution und natürlich in den dokumentierten Quellen [1].

Erweiterung des Hook-Konzepts auf Absätze

Größtenteils ausgelöst durch die Anforderung, die Verarbeitung von Absatztext besser steuern zu können – insbesondere beim Erstellen von Tagged PDF – haben wir L^AT_EX so geändert, dass der Kernel sowohl zu Beginn als auch am Ende jedes Absatzes die Kontrolle erhält. Das geschieht transparent für Pakete und Dokumente (zumindest sollte es so sein).

Neben dem Ergänzen interner Kontrollpunkte für die exklusive Verwendung durch den L^AT_EX-Kernel haben wir auch vier öffentliche Hooks implementiert, die in Paketen oder Dokumenten genutzt werden können (über die normalen Hook-Management-Deklarationen), um beispielsweise besondere Effekte zu erzielen. Bisher

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

² Um besser auf die TeXLive-Releases abgestimmt zu sein, wird L^AT_EX ab sofort jeweils im Mai/Juni und im November in einer neuen Version verteilt.

war für solche Erweiterungen die Neudefinition von `\everypar` oder `\par` erforderlich, was die üblichen Probleme verursachte, da solche Änderungen leicht zu Konflikten mit Änderungen durch andere Pakete führen konnten.

Die Dokumentation dieser neuen »Absatz-Hooks« findet sich zusammen mit ein paar Beispielen in `ltpara-doc.pdf` und den (ziemlich interessanten) Code kann man – wenn man denn Interesse daran hat – in `ltpara-code.pdf` betrachten. Zudem sind beide Dateien als Teil der vollständigen Kernel-Dokumentation in `source2e.pdf` enthalten.

Erweiterung des Hook-Konzepts auf Befehle

Bisher hat das Hook-Management nur Hooks für ein paar wenige Kernbereiche abgedeckt, wie zum Beispiel die Hooks für den `\shipout`-Prozess oder die in der document-Umgebung, sowie ein paar »generische« Hooks zum Laden von Dateien (hilfreich beim Patchen solcher Dateien) und für eigene Umgebungen (die Hooks, die innerhalb von `\begin` und `\end` ausgeführt wurden). Dieses Konzept der »generischen Hooks« wurde nun erweitert, um `/before`- und `/after`-Hooks für sämtliche Befehle auf Dokumenten-Ebene bereitzustellen – zumindest in der Theorie.

In der Praxis können diese neuen generischen `cmd`-Hooks (insbesondere die `cmd/.../after`-Hooks) bei Befehlen fehlschlagen, die zu komplex sind, um sie automatisch zu patchen, und die dann kaputtgehen, wenn der Hook Code enthält. Diese Einschränkungen sind in `ltxcmdhooks-doc.pdf` dokumentiert. Aber da diese Hooks vor allem für Entwicklerinnen und Entwickler gedacht sind, die eine bessere Interoperabilität zwischen unterschiedlichen Paketen sowie zwischen Paketen und dem L^AT_EX-Kernel ermöglichen wollen, sind dies keine zu großen Einschränkungen. Für Befehle, bei denen dieser Mechanismus nicht zum Einsatz kommen kann, verändert sich tatsächlich nichts – für alle anderen wird es eine merkbare Verbesserung geben.

Diese Hooks werden besonders wichtig für unser aktuelles Projekt sein, mit dem wir eine barrierefreie und getaggte PDF-Ausgabe [3] ermöglichen wollen, weil wir letztendlich (zumindest temporär) viele Pakete von dritter Seite patchen müssen – was kontrolliert und standardisiert über die Bühne gehen muss.

Andere Hook-Veränderungen

Eine Seite ausgeben und dabei Hooks umgehen

Im Release vom Oktober 2020 wurde eine Reihe von Hooks hinzugefügt, um den Prozess des Erstellens und Ausgebens einer Seitenbox zu steuern: Diese unterstützen beispielsweise das Hinzufügen von Hintergrund- und Vordergrundmaterial für einige oder alle Seiten.

Wir haben nun einen Befehl `\RawShipout` hinzugefügt, der die Seitenbox nicht neu baut und damit einen Großteil dieser Hooks nicht ausführt. Beim Einsatz dieses neuen Befehls werden die wichtigsten internen Verwaltungsaktivitäten trotzdem ausgeführt, wie zum Beispiel das Aktualisieren des `totalpages`-Zählers oder das Hinzufügen von `shipout/firstpage`- oder `shipout/lastpage`-Material (sofern sinnvoll).

Ein neuer Lua-Callback in `ltshipout` für eigene Attribute

Für die Verwendung direkt vor dem Ausgeben einer Seite gibt es nun einen neuen Lua_T_EX-Callback `pre_shipout_filter` für abschließende Anpassungen an der auszugebenden Box. Das ist besonders nützlich für Lua_T_EX-Pakete, die Elemente auf einer Seite markieren (zum Beispiel mit Attributen oder Eigenschaften), um bei der Ausgabe Effekte auf diese Elemente anzuwenden (wie zum Beispiel das Einfügen von »Farbbefehlen«).

Verbesserter Umgang mit Dateinamen

Dateinamen mit Leerzeichen, mehreren Punkten oder UTF-8-Zeichen

In einem der letzten L^AT_EX-Releases haben wir die Schnittstelle für die Angabe von Dateinamen verbessert, so dass diese nun problemlos Dateinamen mit Leerzeichen (wie es heutzutage üblich ist), mehr als einen Punkt und auch UTF-8-Zeichen außerhalb des ASCII-Bereichs enthalten können. In der Vergangenheit war das nur durch eine spezielle Syntax für Leerzeichen möglich, während Dateinamen mit mehreren Punkten häufig nicht funktionierten – ebenso wenig wie die meisten UTF-8-Zeichen.

Konsequenzen für Dateinamen in `\include`

T_EX besitzt eine eingebaute Regel, die besagt, dass Sie normalerweise die Erweiterung weglassen können, wenn sie `.tex` lautet. Daher führen `\input{file}` und `\input{file.tex}` beide dazu, `file.tex` zu laden (sofern vorhanden). Das ist zwar meistens sehr praktisch, aber in manchen Szenarien schlecht (wenn beispielsweise sowohl `file` als auch `file.tex` vorhanden sind) und auch knifflig, wenn Sie versuchen, die Regel manuell zu implementieren.

L^AT_EX besitzt daher eine spezielle Syntax für `\include` und `\includeonly`: Es wird immer davon ausgegangen, dass die Argumente einen Dateinamen ohne Erweiterung enthalten,³ so dass diese `.tex` sein muss. Haben Sie also unabsicht-

³ Bei `\includeonly` eine kommaseparierte Liste solcher Namen.

lich `\include{mychap.tex}` geschrieben (weil Sie beispielsweise von `\input` zu `\include` gewechselt haben), hat L^AT_EX in der Vergangenheit versucht, die Datei `mychap.tex.tex` einzubinden und die Datei `mychap.tex.aux` für interne Informationen verwendet. Der Grund lag darin, dass `\include` beide Dateinamen aus dem angegebenen Argument erstellen musste und sich nicht darum geschert hat, dass die angegebene Erweiterung `.tex` gesondert zu behandeln ist.

Das hat sich mit der neuen Implementierung geändert: Die Erweiterung `.tex` wird nun entfernt/ignoriert, wenn sie angegeben wird. Daher sucht `\include{mychap.tex}` nun nicht mehr länger nach `mychap.tex.tex`, sondern lädt `mychap.tex` und nutzt `mychap.aux`. (GitHub Issue 486)

Normalisieren robuster Befehle in Dateinamen

Der Umgang mit Dateinamen wurde so angepasst, dass `\string` auf normalisierte robuste Befehle innerhalb des Dateinamens angewendet wird. Zuvor hätte beispielsweise `\input{\sqrt{2}}` zu einer Endlosschleife in L^AT_EX geführt, während durch die neue Normalisierung nun nach der Datei mit dem Namen `sqrt_2.tex` gesucht wird (und daher sehr wahrscheinlich »file not found«, aber eben keine Endlosschleife liefert). (GitHub Issue 481)

Fix für filecontents mit UTF-8-Zeichen im Dateinamen

Seit ein paar Releases erlaubt die Umgebung `filecontents` UTF-8-Zeichen im Dateinamen. Es gab aber einen Bug, durch den es nicht möglich war, eine Datei mit UTF-8-Zeichen im Namen zu überschreiben. Das wurde behoben und jetzt ermöglicht `filecontents` beliebige Zeichen im Dateinamen. (GitHub Issue 415)

Aktualisierungen für das Schriftauswahl-Schema

Ein neuer Hook in `\selectfont`

Nachdem `\selectfont` die Schrift geändert hat, führen wir nun einen Hook (`selectfont`) aus, so dass Pakete abschließende Anpassungen vornehmen können. Diese Funktionalität wurde ursprünglich durch das Paket `everyysel` implementiert, unsere Implementierung ist aber ein wenig anders und nutzt das Standard-Hook-Management. (GitHub Issue 444)

Änderung von Schriftstärke/-form verzögert bis `\selectfont`

Mit den 2020 eingeführten NFSS-Erweiterungen können die Einstellungen von Schriftstärke und -form durch Änderungen an der Schriftfamilie beeinflusst werden. Das Festlegen dieser beiden wird daher nun verzögert, bis `\selectfont` ausgeführt wurde – damit werden unnötige oder falsche Substitutionen vermieden, die ansonsten aufgrund der Reihenfolge der Deklarationen erfolgen könnten.

(GitHub Issue 444)

Glyphen, Zeichen & Kodierung

Verbessertes Copy & Paste für pdf_TE_X-Dokumente

Beim Kompilieren mit pdf_TE_X werden jetzt automatisch zusätzliche Informationen (aus der Datei `glyphtounicode.tex`) in die PDF-Datei übernommen, um das Kopieren von Text aus der Datei und das Suchen in ihr zu verbessern.

Das ermöglicht insbesondere, die gebräuchlichsten Ligaturen aus allen erzeugten PDF-Dateien wie erwartet zu kopieren, ohne explizit das Paket `cmap` laden zu müssen.

(GitHub Issue 465)

Unterstützung für mehr Unicode-Zeichen

L^AT_EX kann zwar ziemlich gut Zeichen wie »ṁ« (man beachte den Punkt) setzen, aber bisher konnte es bestimmte Unicode-Zeichen aus dem Block Latin Extended Additional nicht interpretieren. So gab es beispielsweise kein Unicode-Mapping für Zeichen, die genutzt werden, um Sanskrit-Wörter in lateinischer Transkription zu schreiben (wie es in Büchern über Yoga, buddhistische Philosophie und so weiter vorkommt). Mappings für diese Zeichen wurden nun hinzugefügt, so dass sie direkt eingegeben werden können, statt `\d{m}` und ähnliches nutzen zu müssen.

(GitHub Issue 484)

Mehr »Striche« in den Kodierungen OT1, T1 und TU

Beim Einfügen von Text aus externen Quelle kann man auf die drei Unicode-Zeichen "2011 (Non-Breaking Hyphen), "2012 (Figure Dash) und "2015 (Horizontal Bar) stoßen – neben den häufiger vorkommenden "2013 (En-Dash) und "2014 (Em-Dash). Bisher führten die ersten drei Zeichen zu einer Fehlermeldung beim Einsatz von pdf_TE_X (da sie in mit OT1 oder T1 kodierten Fonts nicht verfügbar waren). Nun wird eine Annäherung der Glyphe gesetzt – der »Figure Dash« wird beispielsweise durch einen En-Dash abgebildet.

Mit Unicode-Engines funktionieren sie entweder (wenn die Glyphe im ausgewählten Unicode-Font enthalten ist) oder es wird nichts gesetzt und eine Warnung »Missing character« in der Log-Datei ausgegeben.

Bei allen Engines können diese Zeichen nun zudem über die Befehlsnamen `\textnonbreakinghyphen`, `\textfiguredash` und `\texthorizontalbar` angesprochen werden. *(GitHub Issue 404)*

Poor Man's `\textasteriskcentered`

Beim Symbol `\textasteriskcentered`, das zu den Fußnotensymbolen in L^AT_EX gehört, wird von L^AT_EX davon ausgegangen, dass es in jedem Font mit der Kodierung TS1 (wenn pdfT_EX zum Einsatz kommt) oder mit der Kodierung TU bei Unicode-Engines implementiert ist. Diese Annahme ist leider nicht für alle Fonts korrekt, da beispielsweise die stix2-Fonts diese Glyphe nicht enthalten. Das führt dann zu Meldungen über fehlende Glyphen beim Einsatz von `\thanks` und so weiter.

Daher prüft `\textasteriskcentered` nun, ob es solch eine Glyphe gibt, und nutzt bei ihrem Fehlen ein normales »*«, allerdings leicht vergrößert und nach unten verschoben. Das mag nicht in allen Fällen perfekt sein, ist aber sicherlich besser als gar kein Symbol und eine Warnung, die man leicht übersieht. *(GitHub Issue 502)*

Die Zeichen aus `textcomp` sind im Kernel

Vor ein paar Releases wurde die Funktionalität des Pakets `textcomp` in den L^AT_EX-Kernel integriert. Damit ist es nicht mehr länger erforderlich, dieses Paket zu laden, um auf Glyphen wie `\textcopyright`, `\texteuro` oder `\textyen` zugreifen zu können.

Dabei wurde die Gelegenheit genutzt, ein wenig das Chaos rund um die Frage zu lichten, welche Glyphen aus der TS1-Kodierung in einem gegebenen Font verfügbar sind. Das geschah durch einen Ansatz, der auf Schriftfamilien und -Collections basiert und bei dem die unterschiedliche Glyphen-Abdeckung des »Textsymbols« durch das Zuweisen einer »Sub-Encoding Number« zu einer Schriftfamilie angegeben wurde, die anzeigt, welche Glyphen aus der TS1-Kodierung auf jeden Fall verfügbar sind, wenn man einen Font aus dieser Familie oder Collection nutzt. Diese Zuweisung stellt sicher, dass L^AT_EX bei Fehlern immer zur sicheren Seite hin arbeitet und vorsichtshalber davon ausgeht, dass eine Glyphe nicht verfügbar ist, auch wenn es sie durchaus gibt.

Eine Anmerkung zur Geschichte der »Textsymbole« und der TS1-Kodierung

Die »Textsymbol-Kodierung« (TS1) wurde ursprünglich auf der Cork Conference (1990) als Ergänzung zur T1-Kodierung entworfen. In ihr wurden diverse Symbole zusammengefasst, die bei der Silbentrennung kein Thema sind, und das Paket `textcomp` wurde entwickelt, um sie verfügbar zu machen. Leider war die T_EX-Community ein wenig zu enthusiastisch und sie nahm eine Reihe von Symbolen auf, die nur in einigen wenigen T_EX-Fonts verfügbar sind, und manche, wie die Akzente von Großbuchstaben, die es noch gar nicht gab, aber als Teil der Referenz-Font-Implementierung entwickelt wurden.

Rückblickend war das eine sehr schlechte Idee, weil dadurch (damals) andere existierende Fonts und später hinzukommende neue Fonts nicht den vollständigen Satz an Glyphen bereitstellen konnten, den die TS1-Kodierung enthält. Für bestehende freie PostScript-Fonts haben sich Leute den Aufwand gemacht, virtuelle Fonts zu erzeugen, die (manche) der fehlenden Glyphen nachgebaut haben. Aber das war und ist zeitaufwändig, daher geschah das nur für ein paar Basis-Fonts. Und selbst so enthielten nur ein paar Fonts alle Glyphen aus TS1, daher enthielt `textcomp` schon damals eine lange Liste, in der Fonts in fünf Kategorien unterteilt wurden – abhängig davon, welche Glyphen implementiert wurden und welche fehlten.

Mittlerweile sind viele neue freie Fonts verfügbar. Als wir nun vor kurzem die Funktionalität von `textcomp` in den L^AT_EX-Kernel integrierten, mussten wir leider feststellen, dass mit ihnen auch das Chaos um die Frage gewachsen war, welche Glyphen der TS1-Kodierung von welchen Fonts implementiert werden. Nicht nur wurden viele Lücken gefunden, es war auch nahezu unmöglich, die Fonts sinnvoll zu Unter-Kodierungen zusammenzufassen, die in einer einfachen Abfolge ineinander enthalten sind, das heißt, in der jede Unter-Kodierung neue, aber auch alle Glyphen des Vorgängers enthält.

Letztendlich entschieden wir uns für neun oder zehn Unter-Kodierungen mit jeweils einer vernünftigen Zahl von Fonts, so dass alle Fonts alle Glyphen der Unter-Kodierung implementierten, auf die sie abgebildet waren. Beim Setzen mit einem Font konnte man nun sicher sein, dass ein Befehl wie `\textcopyleft` das angeforderte Zeichen entweder setzen wird (wenn die Glyphe Teil der Unter-Kodierung war, zu der der Font gehört) oder man einen Fehler erhält, dass die Glyphe in diesem Font nicht zur Verfügung steht. Das Mapping würde sicherstellen, dass L^AT_EX immer nur auf der Seite der Vorsicht ist und eventuell eine Glyphe als nicht verfügbar vermeldet, auch wenn diese vorhanden ist, aber nie versucht eine Glyphe zu setzen, welche es im Font nicht gibt.

So ist beispielsweise der alte Font `pcr` (PostScript Courier) wie die meisten anderen alten PS-Fonts auf der Unter-Kodierung 5 abgebildet und meldet damit, dass

`\textasciigrave` nicht zur Verfügung steht, obwohl das für Courier nicht stimmt. Wird solch ein Font genutzt und ist dies ein Problem, gibt es eine Reihe (suboptimaler) Lösungen. Eine ist, das Mapping von Courier zu ändern und so zu tun, als ob es zu einer umfangreicheren Unter-Kodierung gehört, zum Beispiel über

```
\DeclareEncodingSubset{TS1}{pcr}{2}
```

Nachteilig daran ist, dass L^AT_EX dann glaubt, dass andere Glyphen ebenfalls vorhanden sind, die es aber in der Tat nicht gibt. Daher ist es wichtig, das so erzeugte Dokument auf fehlende Glyphen zu überprüfen.

Eine Alternative ist, vorzugeben, dass `\textasciigrave` immer aus der TS1-Kodierung genommen werden kann (ohne weitere Fragen zu stellen):

```
\DeclareTextSymbolDefault{\textasciigrave}{TS1}
```

Auch das ist gefährlich, denn bei einem anderen Font muss das nicht mehr stimmen und man würde eine fehlende Glyphe erhalten.

Die möglicherweise beste Lösung ist daher – wenn es nicht aus anderen Gründen unmöglich ist – einfach einen anderen Font zu verwenden, zum Beispiel den T_EX Gyre Cursor Font (eine Reimplementierung von Courier mit einem umfangreicheren Glyphenset).

Neue oder verbesserte Befehle

`itemize`-Label mit `\labelitemfont` anpassen

Der Befehl `\labelitemfont` wurde schon mit dem L^AT_EX-Release 2020-02-02 eingeführt, aber damals haben wir vergessen, ihn zu beschreiben, was wir hiermit nachholen wollen. Sein Zweck ist, ein paar gravierende Formatierungsprobleme bei der Umgebung `itemize` zu lösen und es leichter zu machen, das Layout bei Bedarf anzupassen. In der Vergangenheit haben `itemize`-Label (zum Beispiel •) gelegentlich auf Font-Änderungen in der Umgebung reagiert und plötzlich ihre Form geändert, zum Beispiel nach •.

Der neue Befehl `\labelitemfont`, der standardmäßig `\normalfont` entspricht, kann genutzt werden, um beim Setzen jedes Labels zusätzliche Steuerungsmöglichkeiten zu haben. Durch die Wahl unterschiedlicher Einstellungen können daher auch andere Effekte erreicht werden. Hier zwei Beispiele:

```
\renewcommand\labelitemfont{\normalfont\fontfamily{lmss}\selectfont}
\renewcommand\labelitemfont{\rmfamily\normalshape}
```

Die erste Definition nutzt die Symbole aus der Schrift Latin Modern Sans, so dass Sie ■, −, * und · erhalten, während die zweite Variante die Schriftfamilie und die Stärke einfriert, die Serie aber variabel hält, so dass ein `itemize` in einem fetten Umfeld

auch fettere Symbole zeigen wird. Setzen Sie `\labelitemfont` auf einen leeren Wert, erhalten Sie das alte, fehlerhafte Verhalten zurück. *(GitHub Issue 497)*

Mehrere Fußnotenzeichen für eine Fußnote erzeugen

Manchmal ist es notwendig, die gleiche Fußnote mehrfach zu referenzieren, also mehrere Fußnotenzeichen mit der gleichen Zahl oder dem gleichen Symbol zu erzeugen. Das lässt sich jetzt einfach erreichen, indem man ein `\label` in der referenzierten `\footnote` aufnimmt und sich auf dieses Label mit dem neuen Befehl `\footref` bezieht. So können Fußnotenzeichen erzeugt werden, die sich auf beliebige Fußnote beziehen (einschließlich derer in `minipages`).

Dieser Befehl `\footref` stand schon zuvor zur Verfügung, allerdings nur in bestimmten Klassen oder über das Paket `footmisc`. *(GitHub Issue 482)*

`\nocite` in der Präambel zulassen

Ein natürlicher Platz für `\nocite{*}` wäre eigentlich die Präambel des Dokuments, aber aus historischen Gründen hat L^AT_EX einen Fehler gemeldet, wenn man es dort untergebracht hat. Dieser Befehl ist nun in der Präambel erlaubt. *(GitHub Issue 424)*

`\\` allgemein robuster machen

Im Jahr 2018 wurden die meisten L^AT_EX-Befehle auf Benutzerebene robust gemacht, auch der Befehl `\\`. Aber `\\` wird in einer Reihe von Umgebungen neu definiert und nicht all diese Fälle wurden abgedeckt – zum Beispiel insbesondere der Einsatz als Zeilentrenner in `tabular`-Strukturen. Das wurde korrigiert, so dass `\\` jetzt unter allen Umständen robust sein sollte.

Diese Änderung hat zudem eine bisher bestehende Anomalie behoben: In einer `tabular`-Präambel der Form

```
{1>{raggedright}p{10cm}r}
```

hätte ein `\\` in der zweiten Spalte die Definition, die innerhalb von `\raggedright` genutzt wird, so dass es nicht für das (vorzeitige) Ende von `tabular` stehen würde. Daher wurde beispielsweise

```
a & b1 \\ b2 & c \\
```

als eine Zeile der `tabular` interpretiert (wie gedacht), während

```
a &    \\ b2 & c \\
```

zu zwei Zeilen führte! Das lag daran, dass `\` direkt hinter `&` immer noch die Bedeutung »Zeilenende« hatte und nicht »Beginne eine neue Zeile in der zweiten Spalte«.

Indem `\` nun robust ist, endet der durch `&` ausgelöste spezielle Scan-Modus sofort, sobald der Befehl beginnt: Die zweite Spalte wird begonnen, was dazu führt, dass `\` als innerhalb dieser Spalte liegend betrachtet wird und daher seine erwartete Bedeutung in der Spalte besitzt.

Wir haben hier wieder für Konsistenz gesorgt: Jetzt erzeugen beide Zeilen eine einzelne tabular-Zeile. Wie zuvor können Sie nun `\raggedright\arraybackslash` in die Präambel von tabular für eine Spalte aufnehmen, um sicherzustellen, dass `\` immer als Zeilentrenner für die Tabelle betrachtet wird, auch wenn dieser in solch einer Spalte vorkommt. Und Sie können `\tabularnewline` verwenden, um explizit für eine neue Tabellenzeile zu sorgen, auch wenn `\` in der aktuellen Spalte eine andere Bedeutung besitzt.

(GitHub Issue 548)

Zusätzlichen Platz zwischen Name und Adresse in der Klasse letter ermöglichen

Der Befehl `\opening` in der Klasse letter erwartet, dass Name und Adresse durch `\` getrennt sind, aber er erlaubt kein optionales Argument für zusätzlichen Platz unter dem Namen. Der Code wurde nun ein wenig angepasst, um das zu ermöglichen.

(GitHub Issue 427)

Ergänzungen zu `\tracingall`

Im Juli 2020 hat David Jones eine Ergänzung für T_EX-Engines vorgeschlagen, mit der es möglich sein sollte, `\tracinglostchars=3` zu setzen, um eine Fehlermeldung zu erzeugen, wenn verwendete Zeichen in einem Font fehlen. Zuvor wurde lediglich eine Warnung über ein fehlendes Zeichen in die .log-Datei (falls `\tracinglostchars > 0`) und zudem am Terminal (falls `> 1`) ausgegeben. Diese Erweiterung wurde für T_EX Live und MiK_TE_X vorgenommen (außer natürlich in Knuths T_EX), so dass Sie nun mit `\tracinglostchars > 2` auch eine Fehlermeldung für jede fehlende Glyphe erhalten.

Im Januar 2021 hat Petr Olšák dann eine zusätzliche Ergänzung vorgeschlagen: Ein neuer Primitiv-Parameter `\tracingstacklevels`, der – wenn sowohl er wie auch `\tracingmacros` positiv sind – für die Tracing-Informationen für jedes Makro visuell dessen Verschachtelungstiefe (mit Hilfe von Punkten) im Makro-Expansion-Stack ausgibt.

Diese Änderungen wurden nun beide in die Debugging-Makros `\tracingall` und `\tracingnone` von L^AT_EX übernommen, so dass diese beiden neuen Erweiterungen

nach Bedarf aktiviert/deaktiviert werden können, sofern die T_EX-Engine sie unterstützt. Ein Beispieldokument, das diese Parameter demonstriert, findet sich im GitHub-Issue.

(GitHub Issue 524)

Code-Verbesserungen

`\par` am Ende von `\marginpar` ausführen

Bisher beendete L^AT_EX ein `\marginpar`, ohne jemals explizit `\par` aufzurufen. Dieser Befehl wurde nun explizit hinzugefügt, weil es für die korrekte Funktionsweise der Absatz-Hooks essenziell ist.

Ein weiterer Fall, bei dem dies zu Problemen führte, war das Paket `lineno`, bei dem die letzte Zeile nicht nummeriert wurde, wenn `\marginpar` ohne explizites `\par` endete.

(GitHub Issue 489)

`\AtEndDocument`-Hook im vertikalen Modus ausführen

Bisher hat `\end{document}` den Code aus dem `\AtEndDocument`-Hook als seine erste Aktion ausgeführt. Das bedeutete, dass dieser Hook im horizontalen Modus ausgeführt wurde, wenn der Anwender oder die Anwenderin keine leere Zeile nach dem letzten Absatz angefügt hatte. So konnte ein störender Abstand hinzugefügt werden, wenn der Code beispielsweise eine `\write`-Anweisung enthielt. Das wurde nun behoben und `\end{document}` gibt als erstes ein `\par` aus, um sicherzustellen, dass es immer in den vertikalen Modus wechselt.

(GitHub Issue 385)

Farbgruppen immer implementieren

Die Verwendung von Farben in bestimmten L^AT_EX-Konstruktionen, insbesondere in Boxen, erforderte eine zusätzliche Gruppierung, um sicherzustellen, dass die Farbeinstellungen nicht *ausbrechen* und außerhalb der Box weiter aktiv sind, obwohl sie das nicht sollen. Um dabei zu helfen, definiert der L^AT_EX-Kernel eine Reihe von Befehlen, zum Beispiel `\color@begingroup`, die hier genutzt werden können.

Bisher wurden diese Befehle initial als No-Ops gesetzt und nur die Farbpakete haben sie neu definiert, damit sie echte Gruppen werden – diese Vorgehensweise verkompliziert das Coding, da man mit der An- und Abwesenheit einer Gruppe rechnen muss (abhängig davon, was im Dokument geladen wird). Der Kernel fügt diese »Farbgruppen« nun immer hinzu.

(GitHub Issue 488)

Für Key/Value-Handler die unbearbeitete Optionsliste bereitstellen

Vor einer ersten Verarbeitung der Optionsliste wird nun die ursprüngliche (nicht-normalisierte, unbearbeitete und unveränderte) Liste der Paket- oder Klassenoptionen als `\@raw@opt@...` gesichert – diese Liste wird nicht vom normalen Code zum Verarbeiten der Optionen genutzt, steht aber für die Verwendung durch erweiterte Key/Value-Handler in Klassen und Paketen zur Verfügung. Beachten Sie, dass sich der normale Code zum Verarbeiten der Optionen aus Gründen der Kompatibilität nicht geändert hat.

Ein Aspekt dieser Änderung betrifft die Standardverarbeitung: Alle Token rechts eines `=`-Zeichens werden bei Aufbau der »Liste mit ungenutzten Optionen« entfernt. So tragen beispielsweise in diesem Release `clip=true` und `clip=false` beide `clip` in die Liste der verwendeten Optionen ein. (GitHub Issue 85)

Neu für `latexrelease`: `\NewModuleRelease`

Um zu erklären, warum dieses neue Feature benötigt wird, sollten wir uns folgendes Beispiel anschauen: Im Release 2020-10-01 wurde das neue Hook-Management-System in den L^AT_EX-Kernel aufgenommen (siehe [4]) und wie alle Änderungen am Kernel wurde es zu `latexrelease` hinzugefügt. Damit wurde es möglich, zu einem Datum zurückzukehren, an dem das Modul noch nicht vorhanden ist, oder von einem älteren L^AT_EX-Release nach vorne zu springen, um das Hook-Management-System zu erhalten (durch das Laden des Pakets `latexrelease`). Aber dieses Zurückrollen von einem neueren Release zum Release 2020-10-01 hat nicht richtig funktioniert, weil versucht wurde, alle Befehle aus `lthooks` erneut zu definieren – was natürlich zu den zu erwartenden Fehlern geführt hat, die Befehle erzeugen, welche mit `\newcommand` oder (wie in `lthooks`) `\cs_new:Npn` definiert wurden.

Um solche Probleme zu beheben, haben wir jetzt `\NewModuleRelease` bereitgestellt, so dass vollständig neue Module mit Hilfe von `latexrelease` so definiert werden können, dass das System beim Zurück- oder Vorwärtsrollen weiß, ob der Code des neuen Moduls gelesen oder vollständig ignoriert werden soll. Mehr Details dazu finden Sie in der Dokumentation zu `latexrelease` (über `texdoc latexrelease`). (GitHub Issue 479)

Kleine Korrektur für ein Zurückrollen vor 2020-02-02

Während das Paket `latexrelease` normalerweise einen älteren L^AT_EX-Kernel ohne große Probleme emulieren kann, hat ein Zurückrollen vor das Release 2020-02-02 nicht richtig funktioniert: Das lag daran, dass das Management des

`\ExplSyntaxOn/Off`-Status für Pakete (nachdem ein auf `expl3` basierendes Paket geladen wurde) nicht mehr durch den Rollback zurückgenommen werden kann, ohne die Catcodes durcheinanderzubringen. Dies wurde behoben, so dass der Rollback nun sorgfältiger darauf achtet, nach dem Ende eines Pakets kein `\ExplSyntaxOn` zu hinterlassen. *(GitHub Issue 504)*

Änderungen an Paketen in der graphics-Kategorie

Warnung beim Laden von Grafikdateien entfernt

Ein voriges Release hat manchmal unbeabsichtigt eine (falsche) Warnung ausgegeben, wenn eine generische Grafikregel zum Finden und Laden einer Grafikdatei mit einer unbekannten Erweiterung verwendet wurde. Diese Warnung besagte fälschlicherweise, dass die Datei nicht gefunden wurde, obwohl sie tatsächlich korrekt geladen wurde. Die Warnung ist nun in diesem Fall verschwunden. *(GitHub Issue 516)*

Laden von gzippten PostScript-Dateien gefixt

In einem vorigen Release wurde unbeabsichtigt der Mechanismus zum Suchen von Dateien verändert, so dass komprimierte PostScript-Dateien beim Laden mit `\includegraphics` zu einem Fehler führten. Das wurde behoben, so dass gzippte Grafikdateien jetzt wieder korrekt geladen werden. *(GitHub Issue 519)*

Änderungen an Paketen in der tools-Kategorie

layout: Sprachoptionen ergänzt

Dieses Paket erkennt jetzt `japanese` und `romanian` als Sprachoptionen. *(GitHub Issues 353 und 529)*

array und longtable: `\` allgemein robuster gemacht

Der Fix für dieses Problem wurde auch auf diese Pakete angewendet – siehe oben. *(GitHub Issue 548)*

longtable: Update mit Fehlerkorrekturen

Das ist ein kleineres Update für das Paket `longtable`, das eine Reihe gemeldeter Fehler behebt – vor allem die Möglichkeit falscher Seitenumbrüche, wenn Gleitobjekte auf der Seite erscheinen, auf der eine `longtable` beginnt. Da dies den Seitenumbruch bestehender Dokumente beeinflussen kann, wird ein Rollback zu `longtable 4.13` (`longtable-2020-01-07.sty`) unterstützt. *(Gnats Issue tools/2914 3396 3512)*

(GitHub Issue 133 137 183 464 561)

trace: Ergänzungen für `\traceon`

Die beiden Erweiterungen `\tracingstacklevels` und `\tracinglostchars` für `\tracingall` (siehe oben) wurden auch für `\traceon` im Paket `trace` vorgenommen, so dass man bei seiner Anwendung auch von diesen neuen Debugging-Möglichkeiten profitieren kann. (GitHub Issue 524)

bm: Bessere Unterstützung für Befehle mit optionalen Argumenten

Die Verwendung mancher optionaler Argumente in `\bm` wurden (seit 2004) nicht mehr unterstützt, wenn `\kernel@ifnextchar` statt `\@ifnextchar` intern vom Format verwendet wurde. Dieses Update handhabt nun beide Versionen dieses Befehls und stellt das ursprüngliche Verhalten wieder her.

Außerdem wurden Paketoptionen für die Unterstützung von »Poor Man’s Bold« in Fallback-Situationen hinzugefügt. (GitHub Issue 554)

Änderungen an Paketen in der `amsmath`-Kategorie

Der Fix für Issue 548 wurde auch in `amsmath` umgesetzt – siehe oben.

(GitHub Issue 548)

Literatur

- [1] L^AT_EX-Dokumentation auf der Website des L^AT_EX Project, <https://latex-project.org/help/documentation/>.
- [2] The L^AT_EX Projekt: »L^AT_EX News – Issue 33«, *TUGboat*, 42.2 (2021), 180–185.
- [3] Frank Mittelbach, Chris Rowley: L^AT_EX Tagged PDF—A blueprint for a large project, <https://latex-project.org/publications/indexbyyear/2020/>.
- [4] L^AT_EX Project Team: L^AT_EX 2_ε News 32, <https://latex-project.org/news/latex2e-news/ltnews32.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.

handoutwithnotes von *Marei Peischl* erweitert beamer um die Möglichkeit, Handouts einer Präsentation samt Notizen zu erstellen.

CTAN:macros/latex/contrib/handoutwithnotes

datestamp von *Niranjana Vikas Tambe* ermöglicht es, Zeitstempel mit fest vorgegebenen Daten in ein Dokument einzufügen.

CTAN:macros/luatex/latex/datestamp

unicodefonttable von *Frank Mittelbach* ist ein Generator für Unicode-Font-Tabellen.

CTAN:macros/latex/contrib/unicodefonttable

bboldx von *Michael Sharpe* erweitert das Paket *bbold* von *Alan Jeffrey* um einen normalen und einen fetten Schriftschnitt.

CTAN:fonts/bboldx

tikz-bagua von *Xu Wang* stellt Befehle bereit, mit denen man Symbole des Yijing (I Ching) oder Zhouyi mithilfe von pgf/TikZ zeichnen kann.

CTAN:graphics/pgf/contrib/tikz-bagua

zx-calculus von *Léo Colisson* stellt ZX-Calculus-Grafiken mithilfe von pgf/TikZ dar. Es geht um Netzwerkdiagramme, die der Penroseschen graphischen Notation ähneln.

CTAN:graphics/pgf/contrib/zx-calculus

lutabulartools von *Kale Ewasiuk* stellt einige neue Befehle für den Umgang mit Tabellen bereit, die mit Lua^AT_EX funktionieren.

CTAN:macros/luatex/generic/lutabulartools

cdcmd von *Longaster* ermöglicht es, expandierbare bedingte Befehle zu definieren.

CTAN:macros/latex/contrib/cdcm

phfextendedabstract von *Philippe Faist* setzt auf *revtex* auf und stellt ein Paket bereit, mit dem man sehr knapp gefasste Zusammenfassungen (*extended abstracts*) setzen kann, wie sie bei manchen Tagungen üblich sind.

CTAN:macros/latex/contrib/phfextendedabstract

phfcc von *Philippe Faist* ermöglicht freie Inline-Kommentare in einem Text, der kollaborativ geschrieben wird. Die Anmerkungen werden farblich markiert, um die Reviewer voneinander zu unterscheiden. Auch längere »Tipps« können in das Dokument eingefügt werden.

CTAN:macros/latex/contrib/phfcc

lua-widow-control von *Max Chernoff* überlässt Lua die Suche und die Bekämpfung von »Hurenkinder« und »Schusterjungen« in einem Dokument, indem die Seitenlänge immer wieder an den Text angepasst wird. Der Verfasser warnt davor, dass dadurch in seltenen Fällen Textteile verlorengehen können. An dem Problem werde gearbeitet.

CTAN:macros/luatex/generic/lua-widow-control

nahuatl von *Roberto Alvarez Zavala* ermöglicht es, klassische Texte in der in Mexiko heimischen indigenen Sprache Nahuatl (Aztekisch) mit L^AT_EX zu setzen. Ziel ist es, die Aztekencodices, die in Form von Bildhandschriften vorliegen, zu setzen. Bisher liegen dazu einige Glyphen vor.

CTAN:fonts/nahuatl

tex-vpat von *Boris Veytsman* enthält den Bericht für T_EX-Live 2021 nach der *Voluntary Product Accessibility Template* (VPAT) zur Barrierefreiheit.

CTAN:documentation/tex-vpat

truthtable von *Dominic Flück* ist ein Paket für LuaL^AT_EX, mit dem man automatisiert Wahrheitstabellen generieren und setzen kann.

CTAN:macros/luatex/latex/truthtable

linenoamsmath von *Karl Wette* geht zurück auf eine Diskussion auf *Stack Exchange* und ermöglicht es, das Paket *lineno* für die Zeilennummerierung zusammen mit *amsmath* zu benutzen.

CTAN:macros/latex/contrib/linenoamsmath

uwa-pcf und *uwa-pif* von *Anthony di Pietro* sind Vorlagen (1) für die Einwilligungserklärung für Teilnehmer an einer medizinischen Studie an der *University of Western Australia* und (2) das dazugehörige Informationsblatt.

CTAN:macros/unicodetex/latex/uwa-pcf (1)

CTAN:macros/unicodetex/latex/uwa-pif (2)

debate von *Yegor Bugayenko* ermöglicht es Korrekturlesern, Kommentare in einen Text einzufügen. Es gibt eine Option, um den Dialog der Korrektoren in der finalen Version auszublenden.

CTAN:macros/latex/contrib/debate

yamlvars von *Kale Ewasiuk* ist ein YAML-Parser für LuaL^AT_EX.

CTAN:macros/luatex/latex/yamlvars

zitie von *Longaster* dient zum Erstellen von Vorlagen für kalligrafische Übungen mit CJK-Schriftzeichen.

CTAN:macros/xetex/latex/zitie

hanzibox von *Nan Geng* erleichtert die Eingabe chinesischer Schriftzeichen mit Hanyu Pinyin und der entsprechenden Übersetzung.

CTAN:macros/xetex/latex/hanzibox

penlight von *Kale Ewasiuk* macht die Penlight-Library Lua^AT_EX-Benutzern zugänglich.

CTAN:macros/luatex/generic/penlight

luapstricks von *Marcel Krüger* ermöglicht es, PSTricks mit Lua^AT_EX zu verwenden, ohne spezielle Befehle oder externe Aufrufe einzusetzen. Das Paket wird automatisch von PSTricks geladen, wenn ein Dokument mit Lua^AT_EX kompiliert wird.

CTAN:graphics/pstricks/contrib/luapstricks

clicks von *Yegor Bugayenko* erzeugt »Klick«-Effekte, die in einer Präsentation ähnlich funktionieren wie in einer Präsentation, die man mit Microsoft Powerpoint erstellt hätte.

CTAN:macros/latex/contrib/clicks

njuthesis von *FengChendian* ist eine Vorlage für Abschlussarbeiten der chinesischen Universität Nanjing.

CTAN:macros/unicodetex/latex/njuthesis

notocondensed von *Bob Tennent* enthält die L^AT_EX-Unterstützung für die schmale Dichte der Schriftart *Noto*, die *Steve Matteson* für Google entworfen hatte.

CTAN:fonts/notocondensed

uwa-letterhead von *Anthony di Pietro* enthält den Briefkopf für Schreiben der *University of Western Australia*.

CTAN:macros/unicodetex/latex/uwa-letterhead

uwa-colours von *Anthony di Pietro* definiert die Farben für die Corporate Identity der *University of Western Australia* mithilfe des Pakets *xcolor*.

CTAN:macros/latex/contrib/uwa-colours

crumbs von *Yegor Bugayenko* setzt eine Brotkrumen-Navigation an den Seitenkopf eines Dokuments, was vor allem für Präsentationen sinnvoll sein kann.

CTAN:macros/latex/contrib/crumbs

styledcmd von *Paolo de Donato* ermöglicht es, verschiedene Versionen desselben Befehls zu definieren, damit man die passende Fassung für das jeweilige Dokument aufrufen kann.

CTAN:macros/latex/contrib/styledcmd

mecaso von *Youssef Derrazi* stellt Formeln aus der Starrkörpersimulation zur Verfügung, die meist lang und daher schwierig zu schreiben sind. Sie stehen in handlichen Befehlen bereit.

CTAN:macros/latex/contrib/mecaso

etl von *Jonathan P. Spratte* ist eine Variante von *l3tl* aus *expl3*, das aber expandierbare token lists verarbeiten kann. Es richtet sich nur an Programmierer.

CTAN:macros/latex/contrib/etl

biblatex-lncs von *Merlin Göttlinger* ist ein Biblatex-Stil für die Springer Lecture Notes in Computer Science.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-lncs

visualfaq-fr von *Jérémy Just* ist die französische Übersetzung der *visualfaq* von *Scott Pakin*, einer neuartigen Suchschnittstelle zur FAQ, die erweitert ist um Links zur französischen L^AT_EX-FAQ, die man auf <https://faq.gutenberg.eu.org/> findet.

CTAN:info/visualfaq-fr

gamebooklib von *Robert J. Lee* dient zum Setzen von »Spielbüchern«, in denen Abschnitte zufällig aufeinander folgen.

CTAN:macros/latex/contrib/gamebooklib

lambdax von *Erwann Rogard* erstellt Lambda-Ausdrücke in L^AT_EX.

CTAN:macros/latex/contrib/lambdax

keyparse von *Erwann Rogard* definiert Ersetzungsregeln auf der Grundlage von Schlüsseln.

CTAN:macros/latex/contrib/keyparse

makelabels von *Markus Kohm* erweitert die Briefklassen von KOMA-Script um eine *makelabels.lco*, die, ähnlich wie bei den Standardklassen, dazu dient, Adressaufkleber zu drucken. Wer nach mehr Gestaltungsmöglichkeiten sucht, möge das Paket *enlab* ergänzend ausprobieren.

CTAN:macros/latex/contrib/makelabels

uni-titlepage von *Markus Kohm* beschäftigt sich mit der Gestaltung der Titelseite eines Dokuments. Tatsächlich verwendete Beispiele von Titelseiten wurden mit wohldefinierter Schnittstelle neu implementiert und mit Befehlen versehen, um diese an eigene Bedürfnisse anzupassen. Dieses Paket stellt diese als Muster bereit.

CTAN:macros/latex/contrib/uni-titlepage

geradwp von *Benoit Hamel* ist eine Dokumentenklasse für die Reihe *Cahiers du GERAD* der *Groupe d'études et de recherche en analyse des décisions* aus Montreal.

CTAN:macros/latex/contrib/geradwp

tikzbricks von *samcarter* stellt Makros bereit, mit denen man Lego-Bausteine und Figuren, die daraus aufgebaut sind, zeichnen kann.

CTAN:graphics/pgf/contrib/tikzbricks

nchairx von *Marvin Dippell* enthält einige mathematische Makros, die am Lehrstuhl für mathematische Physik an der Universität Würzburg verwendet werden.

CTAN:macros/latex/contrib/nchairx

smart-eqn von *Ziyue Xiang* ermöglicht es, mathematische Symbole einheitlich zu formatieren, um sie, je nach Kontext, besser unterscheiden zu können.

CTAN:macros/latex/contrib/smart-eqn

nwafuthesis von *Nan Geng* ist eine Dokumentenklasse für Abschlussarbeiten an der chinesischen *Northwest A&F Universität*.

CTAN:macros/unicodetex/latex/nwafuthesis

zref-check von *Gustavo Barros* prüft, ob Querverweise in einem Dokument zur Laufzeit konsistent sind.

CTAN:macros/latex/contrib/zref-check

graphicscache von *Max Schwarz* beschleunigt und optimiert das Einfügen und Einpassen von Grafiken, indem es die Ausgabe von `\includegraphics` zwischenspeichert.

CTAN:macros/latex/contrib/graphicscache

scrambledenvs von *Dennis Chen* dient dazu, zufällig gestaltete Umgebungen für Hinweise und Lösungen zu erzeugen.

CTAN:macros/latex/contrib/scrambledenvs

hvglogos von *Herbert Voß* ist eine Erweiterung des Pakets *hologo* von *Heiko Oberdiek*. Es definiert die typischen T_EX-Schriftzüge als Befehle und setzt sie als Logos.

CTAN:macros/latex/contrib/hvglogos

palette von *Dennis Chen* erlaubt es, Farben und Symbole jeweils in Paletten darzustellen, um neue Designs zu entwickeln.

CTAN:macros/latex/contrib/palette

toneval von *Chen Yuanhao* stellt Makros bereit, mit denen man den lexikalischen Ton grafisch darstellen kann.

CTAN:graphics/pgf/contrib/tonevalue

aboensis von *Tommi Syrjänen* ist eine Font, der der Schrift nachempfunden wurde, die im frühen 15. Jahrhundert im *Codex Aboensis*, einem schwedischen Rechtsbuch, verwendet wurde.

CTAN:fonts/aboensis

beamertheme-simpliedarkblue von *Pin-Yen Huang* ist ein weiteres Theme für Präsentationen mit Beamer.

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

pdficons von Aaron Drews stellt Elemente bereit, die üblicherweise in Fließbildern (*process flow diagrams*) verwendet werden, um Abläufe in der technischen Chemie zu veranschaulichen.

CTAN:graphics/pgf/contrib/pdficons

xistercian von Jonathan P. Spratte stellt Zeichen bereit, mit denen man Zahlen setzen kann, die einst von den Zisterziensern im frühen 13. Jahrhundert entwickelt wurden.

CTAN:graphics/pgf/contrib/xistercian

easing von Ka-tsun Loh ist eine PGF-Bibliothek zum Zeichnen von Übergangsfunktionen.

CTAN:graphics/pgf/contrib/easing

Bücher

Newsletter zur Typografie

Christine Römer

Obwohl Newsletter oftmals reine Werberundschreiben sind, habe ich als eine Person, die sich für Typografie interessiert, zwei informative deutschsprachige Newsletter zur Typografie abonniert, die ich hier weiterempfehlen möchte. Deshalb möchte ich sie kurz vorstellen.

Typolexikon

Beim *Typolexikon* handelt es sich um einen alle zwei Monate erscheinenden Newsletter zu neuen typografischen Fachbeiträgen (<https://www.typolexikon.de/>), das von dem renommierten Grafikdesigner und Typografen Wolfgang Beinert herausgegeben wird.

Über das Lexikon wird auf der Webseite ausgeführt:

Typolexikon.de ist ein Lexikon bzw. eine Fachencyklopädie zur Typografie. Es ist ein populärwissenschaftliches Langzeitprojekt, um Fachwissen rund um die Typografie, Schrift und Grafikdesign zu dokumentieren und einer breiten Öffentlichkeit zugänglich zu machen.

Das Typolexikon ging erstmals 2001 online. Im Jahr 2020 nutzten über vier Millionen Besucher dieses Nachschlagewerk.

Es existiert auch ein Archiv des Newsletters¹, das mit dem Jahr 2015 beginnt. Über die Webseite kann man den kostenlosen Newsletter abonnieren. Einige Aufsätze von dort haben wir schon in der Mitgliederzeitung nachdrucken dürfen. Die Erlaubnisse wurden komplikationslos erteilt.

Typografie.info

Zum anderen soll auf *Typografie.info* verwiesen werden, die nach eigenen Aussagen »die meistbesuchte deutschsprachige Typografie-Website« ist. Sie wurde von dem



¹ <https://www.wolfgang-beinert.de/newsletterarchiv/>

Weimarer Typographen Ralf Herrmann gegründet und ist auch die Basis einer »Online-Community mit über 25 000 Mitgliedern«. Sie ist seit 20 Jahren online. Mit »TypoBriefen«, die über die Webseite abonniert werden können, wird monatlich »über ausgewählte neue Inhalte auf [Typografie.info](https://www.typografie.info) informiert«. Neben neuen Artikeln wird auf Typo-Treffen und Neues im Typografie-Shop verwiesen. Das mit [Typografie.info](https://www.typografie.info) verbundene »Schriftkontor Ralf Herrmann« bietet auch Zoom-Meetings zu typografischen Fragen an. Im Mai 2021 habe ich an dem »Typo Treff ONLINE: Oliver Weiß über Schrift-Revivals (kostenlos)« teilgenommen und fand es faszinierend, so einfach an diesem Treffen mit einem Gestalter in den USA teilnehmen zu können. Dabei habe ich mich dazu verleiten lassen, eine schöne, von Weiß nachgestaltete Jugendstilsschrift im Nachgang zu kaufen, die ich bisher nur einmal verwendet habe.

Im Typobrief vom 1.11. wird mitgeteilt: »Neues Video: Schriften-Revivals mit Oliver Weiß«². Folgt man dem Link, findet man allerdings folgende Einschränkung: »Typografie.info-Förderern steht die Aufzeichnung der über einstündigen Veranstaltung hier exklusiv als Video zur Verfügung.«

Generell ist [Typografie.info](https://www.typografie.info) kommerzieller orientiert als [Typolexikon.de](https://www.typolexikon.de), dies zeigte sich auch darin, dass die Bitte um einen Nachdruck des interessanten Artikels »Laufweite und Kerning: Unterschiede und Anwendung« ohne Angabe von Gründen abgelehnt wurde. Über den Link <https://www.typografie.info/3/artikel.htm/wissen/laufweite-und-kerning-unterschiede-und-anwendung-r466/> kann jedoch jeder diesen Artikel nachlesen.



²<https://www.typografie.info/3/artikel.htm/exklusiv/video-schriften-revivals-mit-oliver-weiss-r472/>



Spielplan

März 2022 **DANTE 2022**
und 64. Mitgliederversammlung von DANTE e.V.
Otto-von-Guericke-Universität
39016 Magdeburg

September 2022 **Herbsttagung**
und 65. Mitgliederversammlung von DANTE e.V.
Landesinstitut für Pädagogik und Medien
Beethovenstraße 26
66125 Saarbrücken



12. 9. – 18. 9. 2022 **ConTExT Meeting 2022**
Evangelische Freizeitstätte in Dreifelden/Westerwald
<https://www.rittersmorgen.de>

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-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

pausiert z. Zt. wegen Corona

Berlin

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

mevoges@t-online.de

»Mantée« – Café Restaurant, Chausseestr. 131, 10115 Berlin

Zweiter Donnerstag im Monat, 19:00 Uhr

Darmstadt

Karlheinz Geyer

geyerk@posteo.de

pausiert z. Zt. wegen Corona

findet vorrausichtlich 2022 an jedem ersten Freitag im Monat wieder statt.

Erlangen

Walter Schmidt & Peter Seitz

w.a.schmidt@gmx.net

<http://www.ca131a.de/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

*z. Zt. inaktiv. Bei Fragen steht Günther gern per Mail zur Verfügung.***Hannover**

Reiko Kaps

kaps@luis.uni-hannover.de

<http://tex-hannover.de/>*Zweiter Donnerstag im Monat, 18:30 Uhr**z. Zt. online über <https://vc.sonia.de/TeXStammtischHannover>***Heidelberg**

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

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <https://tinyurl.com/stammtisch-HD>*Physische Treffen bleiben ausgesetzt, solange keine dauerhafte Besserung bzw. Beendigung der gegenwärtigen pandemischen Lage eingetreten ist.***Köln**

Uwe Ziegenhagen

uwe@dante.de

*zur Zeit inaktiv, Interessenten bitte per Mail melden***Leipzig**

Erhard Pross

Erhard.Pross@gmx.de

Ab 2022 geplant. Interessenten bitte per Mail melden

München

Uwe Siart

uwe.siart@tum.de,

<http://www.siart.de/typografie/stammtisch.shtml>

pausiert z. Zt. noch wegen Corona



Stralsund

Heiner Richter

Heiner.Richter@hochschule-stralsund.de

z. Zt. inaktiv, Nachfolge in Planung

Stuttgart

Bernd Raichle

bernd.raichle@gmx.de

Zweiter Dienstag im Monat, 19:30 Uhr

»Paulaner am Alten Postplatz«, Calwer Str. 45, 70173 Stuttgart

Wuppertal

Andreas Schrell

as@schrell.de

Zweiter Donnerstag im Monat, 19:30 Uhr

»Restaurant Croatia«, Südstr. 10, 42103 Wuppertal

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.

Postfach 11 03 61

69072 Heidelberg

Tel.: (0 62 21) 2 97 66 (Mo., Mi., Do. von 9.00–11.30 Uhr)

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	
	Herbert Voß	

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

Luzia Dietsche 71394 Kernen dtkred@dante.de	[3]	Christine Römer christine_roemer@t-online.de	[92]
Walter Entenmann walter.entenmann@t-online.de	[48]	Volker RW Schaa Schriftführer von DANTE e.V. volker@dante.de	[6]
Jürgen Fenn Neu-Isenburg juergen.fenn@gmx.de	[86]	Martin Sievers siehe Seite 98	[4]
Marcel Kapfer tex@mmk2410.org	[11]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[15]
Frank Mittelbach, L^AT_EX Project frank.mittelbach@latex-project.org	[72]	Uwe Ziegenhagen Köln	[46]
Henning Hraban Ramm Hospitalstraße 3 65549 Limburg hraban@fieee.net	[65,69]		

Die T_EXnische Komödie

33. Jahrgang Heft 4/2021 November 2021

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Protokoll der 63. Mitgliederversammlung
- 11 Bericht zur Herbsttagung von DANTE e.V. 2021 in Neuland

Bretter, die die Welt bedeuten

- 15 Ganz- und doppelseitige Gleitumgebungen
- 46 Dymo-Aufkleber mit L^AT_EX gestalten
- 48 Einbetten von Statistik R-Code in L^AT_EX

Von fremden Bühnen

- 65 Der erweiterte Orbit
- 69 Ein neuer Motor für ConT_EXt
- 72 L^AT_EX News – Issue 33, Juni 2021
- 86 Neue Pakete auf CTAN

Bücher

- 92 Newsletter zur Typografie

Spielplan

- 94 Termine
- 95 Stammtische

Adressen

- 99 Autoren/Organisatoren