\textsc{\LaTeX} anniversaries — A look in two directions

Frank Mittelbach

Depending on how you count we have several \LaTeX\ anniversaries to celebrate in 2023: roughly forty years ago Leslie Lamport started his work on \LaTeX\ (which became \LaTeX\ 2.09 in 1986). Ten years later in 1993 we made the first beta version of \LaTeX\ 2e available — since then the standard \LaTeX\ version used across the world.

Thirty years of \LaTeX\ does not mean three decades of standstill — on the contrary. During that time thirty-six new kernel versions have been released and the \LaTeX\ ecosystem grew from a few hundred add-on packages to several thousands.

However, during the first two decades changes to the core of \LaTeX\ were rather minor and most activity was concentrated in the package universe, but the last decade showed an increased level of activity modernizing the \LaTeX\ core functionalities. This started around 2015 when the \LaTeX\ Project Team reimported bug fixes accumulated in a separate package back into the kernel. Since then the format was gradually modernized, e.g., by making UTF-8 the default in 2018 and by incorporating the L3 programming layer in 2020. This intensified further in the last two years when the team embarked on a multi-year journey to enable automatic tagging of the PDF output produced from \LaTeX\.

Once the results of this project are fully available it will be possible to generate accessible documents with \LaTeX\ without the need to post-process the \LaTeX\ output. With the June 2023 release of \LaTeX\ a major milestone of this project will be reached. With this release a restricted class of documents can already be automatically tagged — the digital version of this article is an example for this.

Together with the first release of \LaTeX\ the first edition of \textit{The \LaTeX\ Companion} [12] was published. In 2004 the second edition [42] (describing the extended ecosystem of \LaTeX\) hit the streets, and finally, after five years of writing, the third edition [43] has been published as a two-volume set this time — a living testimony to the widespread use of \LaTeX\ and its by now huge ecosystem.

The remainder of this article consists of an excerpt\footnote{© 2023, Pearson. Reprinted with permission.} from this third edition of \textit{The \LaTeX\ Companion} that describes the \LaTeX\ history in detail.

○ Frank Mittelbach
Mainz, Germany
https://www.latex-project.org

A brief history (of nearly half a century) — excerpt from \textit{The \LaTeX\ Companion, 3rd edition}

In May 1977, Donald Knuth of Stanford University [21] started work on the text-processing system that is now known as “\TeX\ and \METAFONT” [14–18]. In the foreword of \textit{The \TeX\book} [14], Knuth writes: “\TeX\ is a new typesetting system intended for the creation of beautiful books — and especially for books that contain a lot of mathematics. By preparing a manuscript in \TeX\ format, you are telling a computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world’s finest printers.”

In 1979, Gordon Bell wrote in a foreword to an earlier book, \textit{\TeX\ and \METAFONT, New Directions in Typesetting} [13]: “Don Knuth’s Tau Epsilon Chi (\TeX) is potentially the most significant invention in typesetting in this century. It introduces a standard language in computer typography and in terms of importance could rank near the introduction of the Gutenberg press.”

In the early 1990s, Donald Knuth produced an updated version and also officially announced that \TeX\ would not undergo any further development [22, 23] in the interest of stability. Perhaps unsurprisingly, the 1990s saw a flowering of experimental projects that extended \TeX\ in various directions; many of these are coming to fruition in the early 21st century, making it an exciting time to be involved in automated typography.

The development of \TeX\ from its birth as one of Don’s “personal productivity tools” (created simply to ensure the rapid completion and typographic quality of his then-current work on \textit{The Art of Computer Programming}) [19] was largely influenced and nourished by the American Mathematical Society on behalf of U.S. research mathematicians.

While Don was developing \TeX, in the early 1980s, Leslie Lamport started work on the document preparation system now called \LaTeX, which used \TeX\’s typesetting engine and macro system to implement a declarative document description language based on that of a system called
Scribe by Brian Reid [50]. The appeal of such a system is that a few high-level \LaTeX\ declarations, or commands, allow the user to easily compose a large range of documents without having to worry much about their typographical appearance. In principle at least, the details of the layout can be left for the document designer to specify elsewhere.

The second edition of \LaTeX\: A Document Preparation System [25] begins as follows: “\LaTeX\ is a system for typesetting documents. Its first widely available version, mysteriously numbered 2.09, appeared in 1985.” This release of a stable and well-documented \LaTeX\ led directly to the rapid spread of \TeX\-based document processing beyond the community of North American mathematicians.

\LaTeX\ was the first widely used language for describing the logical structure of a large range of documents and hence introducing the philosophy of logical design, as used in Scribe. The central tenet of “logical design” is that the author should be concerned only with the logical content of his or her work and not its visual appearance. Back then, \LaTeX\ was described variously as “\TeX\ for the masses” and “Scribe liberated from inflexible formatting control”. Its use spread very rapidly during the next decade. By 1994 Leslie could write, “\LaTeX\ is now extremely popular in the scientific and academic communities, and it is used extensively in industry.” But that level of ubiquity looks quite small when compared with the present day when it has become, for many professionals on every continent, a workhorse whose presence is as unremarkable as the workstation on which it is used.

The worldwide availability of \LaTeX\ quickly increased international interest in \TeX\ and in its use for typesetting a range of languages. \LaTeX\ 2.09 was (deliberately) not globalized; but it was globalizable; moreover, it came with documentation worth translating because of its clear structure and straightforward style. Two pivotal conferences (Exeter UK, 1988, and Karlsruhe Germany, 1989) established clearly the widespread adoption of \LaTeX\ in Europe and led directly to International \LaTeX\ [54] and to work led by Johannes Braams [1] on more general support for using a wide variety of languages and switching between them (see Chapter 13).

Note that in the context of typography, the word *language* does not refer exclusively to the variety of natural languages and dialects across the universe; it also has a wider meaning. For typography, “language” covers a lot more than just the choice of “characters that make up words”, as many important distinctions derive from other cultural differences that affect traditions of written communication. Thus, important typographic differences are not necessarily in line with national groupings but rather arise from different types of documents and distinct publishing communities.

Another important contribution to the reach of \LaTeX\ was the pioneering work of Frank Mittelbach and Rainer Schöpf on a complete replacement for \LaTeX\’s interface to font resources, the New Font Selection Scheme (NFSS) (see Chapter 9). They were also heavily involved in the production of the A\LaTeX\-based \LaTeX\ system that added advanced mathematical typesetting capabilities to \LaTeX\ (see Chapter 11).

As a reward for all their efforts, which included a steady stream of bug reports (and fixes) for Leslie, by 1989 Frank and Rainer “were allowed” to take over the maintenance and further development of \LaTeX\. One of their first acts was to consolidate International \LaTeX\ as part of the kernel of the system, “according to the standard developed in Europe”. Very soon version 2.09 was formally frozen, and although the change-log entries continued for a few months into 1992, plans for its demise as a supported system were already far advanced as something new was badly needed. The worldwide success of \LaTeX\ had by the early 1990s led in a sense to too much development activity: under the hood of Leslie’s “family sedan” many \TeX\icientists had been laboring to add such goodies as super-charged, turbo-injection, multivalved engines and much “look-no-thought” automation. Thus, the announcement in 1994 of the new standard \LaTeX, christened \LaTeX\ 2.05, explains its existence in the following way:

Over the years many extensions have been developed for \LaTeX\. This is, of course, a sure sign of its continuing popularity but it has had one unfortunate result: incompatible \LaTeX\ formats came into use at different sites. Thus, to process documents from various places, a site maintainer was forced to keep \LaTeX\ (with and without NFSS), SL\LaTeX, \LaTeX\,[2]

---


[3] *Kernel* here means the core, or center, of the system.

Frank Mittelbach
The development of this “New Standard \LaTeX\" and its maintenance system was started in 1993 by the \LaTeX\ Project Team \cite{45}, which soon comprised the author of this book, Rainer Schöpf, Chris Rowley, Johannes Braams, Michael Downes, David Carlisle, Alan Jeffrey, and Denys Duchier, with some encouragement and gentle bullying from Leslie. Although the major changes to the basic \LaTeX\ system (the kernel) and the standard document classes (styles in 2.09) were completed by 1994, substantial extra support for colored typography, generic graphics, and fine positioning control were added later, largely by David Carlisle. Access to fonts for the new system incorporated work by Mark Purtill on extensions of NFSS to better support variable font encodings and scalable fonts \cite{2–4}.

At this point in the story the first edition of the \LaTeX\ Companion was written, which helped a lot in making many important packages known to a wide audience and as a side effect helped shape a standard corpus of \LaTeX\ packages expected to be available on any installation across the world.

Although the original goal for this \LaTeX\ 2e was consolidation of the wide range of incompatible models carrying the \LaTeX\ marquee, what emerged was a substantially more powerful system with both a robust mechanism (via \LaTeX\ packages) for extension and, importantly, a solid technical support and maintenance system. This provides robustness via standardization and maintainability of both the code base and the support systems. The core of this system remains the current standard \LaTeX\ system that is described in this book. It has fulfilled most of the goals for “a new \LaTeX\ for the 21st Century”, as they were envisaged back in 1989 \cite{48, 49}.

The specific claims of the current system are “… better support for fonts, graphics and color; actively maintained by the \LaTeX\ Project Team”. The details of how these goals were achieved, and the resulting subsystems that enabled the claims to be substantially attained, form a revealing study in distributed software support: the core work was done in at least five countries and, as is illustrated by the bugs database \cite{27}, the total number of active contributors to the technical support effort remains high.

Although the \LaTeX\ kernel suffered a little from feature creep in the late 1990s, the package system together with the clear development guidelines and the legal framework of the \LaTeX\ Project Public License (LPPL) \cite{29, 34} have enabled \LaTeX\ to remain almost completely stable while supporting a wide range of extensions. These have largely been provided by a similarly wide range of people who have, as the project team are happy to acknowledge and the online catalogue \cite{56} bears witness, enhanced the available functionality in a vast panoply of areas.

All major developments of the base system have been listed in the regular issues of \LaTeX\ News \cite{26}. At the turn of the century, development work by the \LaTeX\ Project Team focused on the following areas: supporting multi-language documents \cite{32}; a “Designer Interface for \LaTeX\” \cite{40}; major enhancements to the output routine \cite{33}; improved handling of inter-paragraph formatting; and the complex front-matter requirements of journal articles. Back then prototype code had been made available (see \cite{19}), but the work has otherwise been kept separate from \LaTeX\ — partly because it was executing simply too slowly on the available hardware.

One thing the project team steadfastly refused to do at that time was to unnecessarily “enhance” the kernel by providing additional features as part of it, thereby avoiding the trap into which \LaTeX\ 2.09 fell in the early 1990s: the disintegration into incompatible dialects where documents written at one site could not be successfully processed at another site. In this discussion it should not be forgotten that \LaTeX\ serves not only to produce high-quality documents but also to enable collaboration and exchange by providing a lingua franca for various research communities.

With \LaTeX\ 2e, documents written in 1996\footnote{The time between 1994 and 1996 was a consolidation time for \LaTeX\ 2e, with major fixes and enhancements being made until the system was thoroughly stable. In fact, with some minor alterations in pagination or font usage, it is usually possible to reprocess even documents from the eighties (i.e., written for \LaTeX\ 2.09) or make them reusable with little effort.} can still be run with today’s \LaTeX\ In the opposite direction, new documents run on older kernel releases if the additional packages used are brought

\LaTeX\ anniversaries — A look in two directions
up-to-date — a task that, in contrast to updating the \LaTeX\ kernel software, is easily manageable even for users working in a multiuser environment (e.g., in a university or company setting).

But a stable kernel is not identical to a standstill in software development; of equally crucial importance to the continuing relevance and popularity of \LaTeX{} is the diverse collection of contributed packages building on this stable base. The success of the package system for nonkernel extensions is demonstrated by the enthusiasm of these contributors — many thanks to all of them! As can be easily appreciated by visiting the highly accessible and stable Comprehensive \TeX{} Archive Network (see Appendix C) or by reading this book (where more than 250 of these “Good Guys”\footnote{Unfortunately, this is nearly the literal truth: you need a keen eye to spot the few ladies listed.} are listed on page 11-96), this has supported the existence of an enormous treasure trove of \LaTeX\ packages and related software.

The provision of services, tools, and systems-level support for such a highly distributed maintenance and development system was itself a major intellectual challenge, because many standard working methods and software tools for these tasks assume that your colleagues are in the next room, not the next continent (and in the early days of the development, e-mail and FTP were the only reliable means of communication). The technical inventiveness and the personalities of everyone involved were both essential to creating this example of the friendly face of open software maintenance, but Alan Jeffrey and Rainer Schöpf deserve special mention for “fixing everything”.

A vital part of this system that is barely visible to most people is the regression testing system with its vast suite of test files \cite{31}. It was initially devised and set up by Frank and Rainer with Daniel Flipo; it has proved its worth countless times in the never-ending battle with the bugs. Over the years it has seen many refinements, culminating in a complete rewrite as part of \texttt{\texdoc{}} \cite{44}, which we describe in Section 17.3 on page 11-606.

In 2004, i.e., roughly a decade after its first edition, the second edition of the \LaTeX{} Companion was published. Due to the popularity of \LaTeX{} 2\epsilon and its extended features for developers, new important packages had emerged, and \LaTeX{} had reached out into new domains. While the advice given in the first edition remained largely valid (last but not least because of the long-term backward compatibility paradigm of \LaTeX{}), we ended up rewriting 90\% of the original content and added about 600 pages to account for new developments. As before, the second edition helped a lot in standardizing the use, and this way the interoperability, of \LaTeX{} across the world.

Some members of the \LaTeX{} Project Team have built on the team’s experience to extend their individual research work in document science beyond the current \LaTeX{} structures and paradigms. Some examples of their work up to now can be found in the following references: \cite{5, 7–9, 35–38, 46, 51, 53}. An important spin-off from the research work was the provision of some interfaces and extensions that are immediately usable with standard \LaTeX{}.

The decision to keep the core of the standard \LaTeX{} system stable and essentially unchanging had two major advantages over any other approach to support fully automated document processing. First, the system already efficiently provided high-quality formatting of a large range of elements in very complex documents of arbitrary size. Second, it was robust in both use and maintenance and hence offered the potential to remain in widespread use for at least a further 15 years.\footnote{One of the authors of the second edition had publicly staked a modest amount of beer on \TeX{} remaining in general use (at least by mathematicians) until at least 2010. He should have made a larger bet, given that this is now 2023 and \LaTeX{} is healthy and in fact growing its user base due to its many unsurpassed qualities.} In the second edition of this book we wrote on this topic:

As more such functionality is added, it will become necessary to assess the likelihood that merely extending \LaTeX{} in this way will provide a more powerful, yet still robust and maintainable, system. This is not the place to speculate further about the future of \LaTeX{} but we can be sure that it will continue to develop and to expand its areas of influence whether in traditional publishing or in electronic systems for education and commerce.

This reassessment became necessary in the second decade of the new century, when it became obvious that this position was gradually getting unsustainable, because more and more areas in which people were looking for solutions could not be adequately addressed with a model of a fixed

Frank Mittelbach
kernel and all developments outsourced to the package level. Examples are the move to Unicode in basically all operating systems and the growing pressure to produce “accessible” documents that conform to standards such as PDF/UA (Portable Document Format/Universal Accessibility).

Thus, in 2013, the \LaTeX{} Project Team changed its policy and restarted kernel development. To retain the best of both worlds this was accompanied by developing a rollback/roll-forward functionality for the kernel and packages (that care to implement it). This allows a current \LaTeX{} format to roll back to an earlier point in time in order to process old documents that rely on interfaces that have been changed since then or to process documents that explicitly worked around bugs (and so expect them to be there) that have been fixed in the meantime.

The first action of the team was to retire the fmtex package and instead include the accumulated fixes it contained directly in the format and to officially support \LaTeX{} when using the Unicode engines X\LaTeX{} and Lua\LaTeX{}. A big step forward happened in 2018 when \LaTeX{} switched its default input encoding to UTF-8. This change proved that the policy change was the right thing to do and that the preparatory work (e.g., providing rollback) allows executing even major changes without disruption in its user base in order to keep \LaTeX{} relevant and useful. A good indicator for the renewed and increased activity are the regular \LaTeX{} newsletters \cite{26} accompanying each release, which grew bulkier and again appeared semi-annually.

The event of providing the mythical \LaTeX{}3 had long become a standing joke as “two years from ‘now’ — with ‘now’ a moving target”. The reason was that the concepts and ideas for \LaTeX{}3 have been simply a decade or more too early, and while the team implemented a fully working version already in 1990, it was simply too slow to be usable with the then available computing power. Thus, we gave up pursuing it and instead concentrated on offering \LaTeX{}2ε, which then went public in 1994.

But ideas and concepts were never forgotten by the team, and especially its newer members (who joined in this century) pushed them back to the forefront and improved them dramatically. As a result, the code was eventually publicly made available as the expl3 package. It was then picked up by a number of enthusiastic package developers and used as the basis for their new packages. For example, if you use acro, breq, fontspec, siunitx, unicode-math, or xparse, to name a few, you use “\LaTeX{}3” under the hood; a recent count shows more than 200 such packages or classes as part of \LaTeX{} Live.

So in 2019 the \LaTeX{} Project Team made two wide-ranging decisions: there will not be a separate \LaTeX{}3 that is being developed alongside \LaTeX{}2ε (as was originally planned). Instead, we will modernize the current \LaTeX{} gradually from the inside, using the new rollback mechanism and “development” formats as a safety net to ensure that there is no disruption of service for our user base. As a first step on this journey, the L3 programming layer and the \LaTeX{}X document-level command declarations (formerly known as expl3 and xparse) were made an integral part of \LaTeX{} on February 2, 2020. Thus, more or less exactly 30 years after its conception, \LaTeX{}3 became a reality for every \LaTeX{} user — even though few will have immediately noticed.

The importance of this step is that it allows the team to modernize other parts of the kernel and develop new functionality entirely based on the L3 programming layer, which offers many features not available with legacy \LaTeX{} programming constructs. For example, the new Hook Management System for \LaTeX{}, which is a cornerstone for modernizing and transforming the existing \LaTeX{}, is entirely written using the new L3 programming layer, and other parts will follow suit.

As already mentioned, there is a steadily increasing interest in the production of “tagged” PDF documents that are “accessible”, in the sense that they contain information to assist screen reading software, etc., and, more formally, that they adhere to the PDF/UA (Portable Document Format/Universal Accessibility) standard \cite{55}, explained further in \cite{10}. In many disciplines this is starting to become a requirement when applying for grants or when publishing results.

At the moment, all methods of producing such “accessible PDFs”, including the use of \LaTeX{}, require extensive manual labor in preparing the source or in post-processing the PDF (maybe even at both stages); and these labors often have to be repeated after making even minimal changes to the (\LaTeX{} or other) source. This is a huge pity, because \LaTeX{} should in theory be well-positioned to do this work automatically, given that its source is already well-structured.

The production of tagged (i.e., structured) PDF documents is not only important in order to comply to accessibility standards. It also opens possibilities to reuse data from such PDFs, because
it allows other applications to correctly identify the structure inside the output document and this way extract or manipulate parts of the content — workflows that become increasingly important in the digital world.

The \texttt{\LaTeX} Project Team has for some years been well aware that these new usages are not adequately supported by the current system architecture of \texttt{\LaTeX} \texttt{2e} and that major work in this area is therefore urgently needed to ensure that \texttt{\LaTeX} remains an important and relevant document source format. However, the amount of work required to make such major changes to the \texttt{\LaTeX} system architecture is enormous and definitely way beyond the limited resources of a small team of volunteers working in their spare time (or maybe just about possible, but only given a very long — and most likely too long — period of time).

At the \texttt{\LaTeX} Users Group conference 2019 in Palo Alto the team’s previously pessimistic outlook on this subject became cautiously optimistic, because of discussions with senior executives from Adobe about the possibility of producing structured PDF from \texttt{\LaTeX} source without the need for the usual requirement of considerable manual post-processing. As a result of these discussions, towards the end of 2019 the team produced an extended feasibility study for the project, aimed primarily at Adobe engineers and decision-makers. This study \cite{Mittelbach2020} describes in some detail the various tasks that constitute the project and their interdependencies. It also contains a project plan covering how, and in what order, these tasks should be tackled both to achieve the final goal and, at the same time, to provide intermediate concrete results that are relevant to user communities (both \texttt{\LaTeX} and PDF); these intermediate results will help in obtaining feedback that is essential to the successful completion of later tasks.

This multi-year project found the approval of Adobe, which then committed to financially and otherwise supporting this endeavor \cite{Adobe2020}. Unfortunately — thanks to the COVID-19 pandemic — the start got delayed, but since the end of 2020, this exciting project is now well under way. First results from this project that are already in existence (such as the new hook management system and the alignment of the \texttt{hyperref} package with the \texttt{\LaTeX} kernel) are already described in this book. Other parts are obviously still vaporware at this point. Fortunately, none is expected to render any documentation or suggestion made in this book obsolete — after all, the project goal is to enable tagging of existing documents, simply by reprocessing with minor configuration changes as outlined in the “Spoiler alert” Section 2.1.1 on page 23.

References

\begin{enumerate}

The babel package was originally a collection of document-style options to support different languages. An update was published in \textit{TUGboat}, 14(1):60–62, April 1993.


A “guided tour” around the files in the basic \texttt{\LaTeX} distribution. File names and paths relate to the file hierarchy of the CTAN archives.


A “guided tour” around the “tools” and “graphics” packages. Note that Lamport’s manual \cite{Lamport1994} assumes that at least the graphics distribution is available with standard \texttt{\LaTeX}.


A “guided tour” through more distributions that are part of the standard \texttt{\LaTeX} system. The mfnss distribution provides \texttt{\LaTeX} support for some popular \texttt{METAFONT}-produced fonts that do not otherwise have any \texttt{\LaTeX} interface. The psfss distribution consists of \texttt{\LaTeX} packages giving access to \texttt{PostScript} fonts. The babel distribution provides \texttt{\LaTeX} with multilingual capabilities.


\item —. “\texttt{xmltex}: A non validating (and not 100\% conforming) namespace aware \texttt{XML} parser implemented in \texttt{\LaTeX}”. \textit{TUGboat}, 21(3):193–199, 2000.

\texttt{xmltex} is an \texttt{XML} parser and \texttt{typestern} implemented in \texttt{\LaTeX}, which by default uses the \texttt{\LaTeX} kernel to provide typesetting functionality.


This is the draft specification for a new version of the Mathematical Markup Language; the current version is 3.0 \cite{MathML4}. MathML4 extensions primarily relate to improving accessibility, with new attributes for improving audio rendering.

https://www.w3.org/TR/mathml4/

\end{enumerate}


Contains an article on “Mathematical Typography”, describing the author’s motivation for starting to work on \TeX\ and the early history of computer typesetting. Describes early (now obsolete) versions of \TeX\ and \Metafont.


The definitive user’s guide and complete reference manual for \TeX. A good secondary reading, covering the same grounds, is [11].


The complete source code for the \TeX\ program, typeset with several indices.


The user's guide and reference manual for \Metafont, the companion program to \TeX\ for designing fonts.


The complete source code listing of the \Metafont program.


More than 500 Greek and Roman letterforms, together with punctuation marks, numerals, and many mathematical symbols, are graphically depicted. The \Metafont\ code to generate each glyph is given and it is explained how, by changing the parameters in the \Metafont\ code, all characters in the Computer Modern family of typefaces can be obtained.


Donald Knuth’s major work on algorithms and data structures for efficient programming.

\TeX\ anniversaries — A look in two directions
[20] ———. Digital Typography. CSLI Publications, Stanford, CA, USA, 1999. ISBN 1-57586-011-2 (cloth), 1-57586-010-4 (paperback). A comprehensive collection of Knuth’s writings on \TeX{} and typography. While many articles in this collection are available separately on the Web, not all of them are, and they all in one place for studying is an additional benefit.

[21] ———. “Computers and typesetting”. In Knuth [20], pp. 555–562. Remarks presented by Knuth at the Computer Museum, Boston, Massachusetts, on 21 May 1986, at the “coming-out” party to celebrate the completion of \TeX{}.

[22] ———. “The new versions of \TeX{} and METAFONT”. In Knuth [20], pp. 563–570. Knuth explains how he was convinced at the TUG Meeting at Stanford in 1989 to make one further set of changes to \TeX{} and METAFONT to extend these programs to support 8-bit character sets. He goes on to describe the various changes he introduced to implement this feature, as well as a few other improvements.

[23] ———. “The future of \TeX{} and METAFONT”. In Knuth [20], pp. 571–572. In this article Knuth announces that his work on \TeX{}, METAFONT, and Computer Modern has “come to an end” and that he will make further changes only to correct extremely serious bugs.

[24] Donald E. Knuth and Michael F. Plass. “Breaking paragraphs into lines”. In Knuth [20], pp. 67–155. This article, originally published in 1981, addresses the problem of dividing the text of a paragraph into lines of approximately equal length. The basic algorithm considers the paragraph as a whole and introduces the (now well-known) \TeX{} concepts of "boots", "glue", and "penalties" to find optimal breakpoints for the lines. The paper describes the dynamic programming technique used to implement the algorithm.


[26] \LaTeX{} Project Team. “\LaTeX{} news”. An issue of \LaTeX{} News is released with each \LaTeX{} 2e release, highlighting changes since the last release. There is also a document combining all issues since 1994, which offers a good overview about the history of \LaTeX{} 2e as well as providing an easy way to find information on all major updates and extensions that have been implemented over the years. \LaTeX{} News is available at \url{https://www.latex-project.org/news/}.

[27] ———. “Bugs in \LaTeX{} software”. Website. The bug reporting and tracking service run by the \LaTeX{} team as part of the \LaTeX{} 2e maintenance activity. \LaTeX{} bug reports are available at \url{https://www.latex-project.org/bugs/}.

[28] ———. The \LaTeX{}3 Interfaces, 2023. The reference manual for the L3 programming layer, which has been part of the \LaTeX{} format since 2010 and thus available for package development — the way for \LaTeX{} coding going forward. \LaTeX{}3 manual is available at \url{https://www.latex-project.org/l3pl/}.

[29] ———. “The \LaTeX{} project public license (version 1.3c)”, 2008. The Open Source License used by the core \LaTeX{}2e distribution and many contributed packages. See \cite{LPPL} for background and history. \LaTeX{} project public license is available at \url{https://www.latex-project.org/lppl/}.

[30] Frank Mittelbach. “E-\TeX{}: Guidelines for future \TeX{} Extensions”. \textit{TUGboat}, 11(3):337–345, 1990. The output of \TeX{} is compared with that of hand-set typeset documents. It is shown that many important concepts of high-quality typesetting are not supported and that further research to design a “successor” typesetting system to \TeX{} should be undertaken. A review of the findings, 23 years later, is provided in \cite{Mittelbach2023}.

[31] ———. “A regression test suite for \LaTeX{} 2e”. \textit{TUGboat}, 18(4):309–311, 1997. Description of the concepts and implementation of the test suite used to test for unexpected side effects after changes to the \LaTeX{} kernel. One of the most valuable maintenance tools for keeping \LaTeX{} 2e stable. \LaTeX{} regression test suite is available at \url{https://www.latex-project.org/tb18-4/tb67dirt.pdf}.


[34] ———. “Reflections on the history of the \LaTeX{} Project Public License (LPPL) — A software license for \LaTeX{} and more”. \textit{TUGboat}, 32(1):83–94, 2011. A review of the evolution of \LaTeX{} world’s predominant license \cite{LPPL}.

Frank Mittelbach
[35] ——. “E-TEx: Guidelines for future TEx Extensions — revisited". TUGboat, 34(1):47–63, 2013. This article compares the output of E-TEx with that of hand-typset documents. This is a reassessment of the findings made 33 years earlier [36]. With the new engines the situation has improved, but even though there is now engine support for most problems, the majority of them still represent important and open research problems for high-quality automated typesetting. https://tug.org/TUGboat/tb34-1/tb106mitt.pdf


[38] ——. “A general LuaTEx framework for globally optimized pagination". Computational Intelligence, 35(2):242–284, 2019. This article is an extended version (17 pages) of the 2016 ACM article "A General Framework for Globally Optimized Pagination" [36], providing much more detail and additional research results. The peer-reviewed publication is now freely available. https://www.latex-project.org/publications/indexbyyear/2020/

[39] Frank Mittelbach, David Carlisle, and Chris Rowley. “Experimental L\LaTeX\ code for class design". Vancouver, 1999. At the \LaTeX Users Group conference in Vancouver the \LaTeXX project team gave a talk on models for user-level interfaces and designer-level interfaces in \LaTeXX [40]. Most of these ideas have been implemented in prototype implementations (e.g., template design, front matter handling, output routine, galley and paragraph formatting). The source code is documented and contains further explanations and examples; see also [35]. The underlying programming interfaces are since 2020 part of the \LaTeX format as the L3 programming layer [48]. Articles: https://latex-project.org/publications/indexbytopic/13-expl13 Code: https://github.com/latex3/latex3

[40] ——. “New interfaces for L\LaTeX\ class design, Parts I and II". TUGboat, 20(3):214–216, 1999. Some proposals for the first-ever interface to setting up and coding \LaTeX classes. While all of them were implemented as experimental prototypes (see [39]), they have been developed at a time were computers were not powerful enough to enable them for general use. This has finally changed and several of these ideas are now making their reappearance as part of the "\LaTeXX Tagged PDF" project [47]. https://tug.org/TUGboat/tb20-3/tb64carl1.pdf

[41] Frank Mittelbach, Ulrike Fischer, and Chris Rowley. \LaTeXX Tagged PDF Feasibility Evaluation. \LaTeXX Project, 2020. This is the feasibility study undertaken by the \LaTeXX team prior to initiating the multiyear project for automatically providing tagged PDF with \LaTeXX. It explains in detail both the project goals and the tasks that need to be undertaken and concludes with a detailed project plan. See also [47]. https://latex-project.org/publications/indexbytopic/pdf/


[44] Frank Mittelbach, Will Robertson, and \LaTeXX team. "\LaTeXX — A modern Lua test suite for \LaTeX programming". TUGboat, 35(3):287–293, 2014. The workflow environment used by the \LaTeXX Project Team and others. Supports concepts developed over the years including regression testing methods, distribution builds, uploads to CTAN, and installation support. https://tug.org/TUGboat/tb35-3/tb111mitt-l3build.pdf


\LaTeX anniversaries — A look in two directions


[48] Frank Mittelbach and Rainer Schöpf. “With \LaTeX\ into the nineties”. TUGboat, 10(4):681–690, 1989. This article proposes a reimplementation of \LaTeX\ that preserves the essential features of the current interface while taking into account the increasing needs of the various user communities. It also formulates some ideas for further developments. It was instrumental in the move from \LaTeX 1.09 to \LaTeX 2e. https://tug.org/TUGboat/tb10-4/tb26mitt.pdf

[49] ——. “Towards \LaTeX 3.0”. TUGboat, 12(1):74–79, 1991. The objectives of the \LaTeX3 project are described. The authors examine enhancements to \LaTeX\’s user and style file interfaces that are necessary to keep pace with modern developments, such as SGML. They also review some internal concepts that need revision. https://tug.org/TUGboat/tb12-1/tb31mitt.pdf


[56] Graham Williams. “Graham Williams’ \LaTeX\ Catalogue”. TUGboat, 21(1):17–90, 2000. In 2000 this catalogue listed more than 1500 \LaTeX, \LaTeX\X, and related packages and tools on 74 pages and was linked directly to the items on CTAN. CTAN now offers it in the form of several indexes with more than 5000 items covering everything stored there. https://tug.org/TUGboat/tb21-1/tb66catal1.pdf

Latest version on CTAN at: https://ctan.org/pkg/catalogue

Frank Mittelbach