View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Dokumenten-Publikationsserver der Humboldt-Universität zu Berlin

LaTeX as an Archiving Format: Benefits and Problems

Experiences from the Math*Diss* International Project and the EMANI project

Thomas Fischer
Niedersächsische Staats- und Universitätsbibliothek, Göttingen
fischer@mail.sub.uni-goettingen.de
Papendiek 14, D-37073 Göttingen
Keywords: Archiving, file formats, mathematics

Abstract

Today, LaTeX is the standard format for writing papers in Mathematics as well as the preferred format for a major part of Physics. For presentations on the Web, these formats are usually transferred to PDF, a convenient format available for many different platforms, allowing direct viewing with appropriate rendering ("Reader") software.
On the other hand, PDF is not the optimal format for long-term storage, because

- *it is owned by a commercial company*
- *it is not stable over time (some older files cannot be read using the newer rendering software)*
- *it is not fault-tolerant: compressed versions of PDF in particular may become completely unreadable if corrupted.*
- *some PDF files do not allow the efficient extraction of the text behind the presentation, which prevents efficient indexing for search and retrieval.*

Since LaTeX is a pure text-based format with additional mark-up and available as open source software, LaTeX is a much safer choice for long-term preservation.
But this presents several other problems:

- *While the PDF format is not the original, it provides the fixed pagination for reference - different compilations of the same LaTeX file under different conditions may provide different paginations.*
- *LaTeX version of a dissertation may involve several different files: TeX, images, styles, macro packages etc., some of which may be necessary, some others not.*
- *Some LaTeX files do not compile correctly, because some necessary files are missing.*

Although these problems exist, the advantages of using LaTeX as an archival format outweigh the problems. However, for its efficient use, some developments are necessary:

- *automated validation of (collections of) TeX files*
- *efficient administration of auxiliary files*

It might be useful to consider the packaging of LaTeX files into one (opaque) file, which could be rendered using a "TeX-Reader". This could increase the acceptance of the LaTeX outside of the mathematics community tremendously.

Preface

Scientific content, and mathematics in particular, is today usually produced on a computer, put on the web as preprint and finally - after some sort or peer-review - published in a printed or electronic journal. So there is a growing amount of scientific contents that is never "officially" printed, and therefore will be lost for the community unless some measures for long-term-preservation are taken. Different projects (see e.g. Cedars, http://www.leeds.ac.uk/cedars/, CAMILEON, http://www.si.umich.edu/CAMILEON/research/research.html, the Digital Library Federation, http://www.diglib.org/preserve.htm) are working on this problem, and national libraries start to provide first solutions. In the Netherlands, the Koninklijke Bibliotheek and IBM (see http://www.kb.nl/kb/resources/frameset_kb.html?/kb/pr/pers/pers2000/ibm-en.html) are working on a solution based on the principles of the Open Archival Information System (OAIS, see http://wwwclassic.ccsds.org/documents/pdf/CCSDS-650.0-B-1.pdf). In Germany, a project funded by the Federal Ministry of Education and Research and lead by "Die Deutsche Bibliothek" is laying the groundwork for a cooperative archiving SYSTEM for digital documents (see http://www.dl-forum.de/Foren/Langzeitverfuegbarkeit/index.asp).

In this context, considerations related to the archiving format are of interest: what is the securest, most efficient way to preserve scientific articles over long time periods, when the technical development of software and hardware tends to make any particular file format and programme obsolete within few years? While there are some emerging "de facto standards" presented by software companies (see e.g. Adobe Systems Incorporated: PDF as a Standard for Archiving, http://www.adobe.com/products/acrobat/pdfs/pdfarchiving.pdf), there is reluctance to rely on commercial solutions and doubt about the long-term reliability and responsibility of commercial companies.

In mathematics, most articles and books nowadays are written using some version of Donald E. Knuth's typesetting SYSTEM TEX. Since this is the form the content is created in, it is an obvious question if this should not be used as well to archive the results. This article presents some experiences with TEX as an archiving format, gathered in the context of the project "Math*Diss* International" (see http://www.ub.uni-duisburg.de/mathdiss/) and its database of dissertations (http://www.sub.uni-goettingen.de/ssgfi/mathdiss/) in particular, and in considerations related to the project "EMANI" (http://www.emani.org/). I will start with some general principles and rules derived

from the work with mathematics files; afterwards some specific examples from the projects will be given.

To give a short conclusion in advance: (La)TeX is an excellent format for preserving mathematical contents, but it needs extensive work to be tamed.

Basis Considerations on File Types

Before entering the specific discussion of archiving formats for mathematics and TeX in particular, some general remarks on file types will be useful.

Types of file formats

We will consider essentially textual data, leaving aside the problems related to the various kinds of images, which will be attached to or embedded in the files carrying the text. The textual information is to be presented in a uniform fashion, leaving changes in layout (e.g. paper size) at the discretion of the individual user. It is often useful to distinguish between the textual content and the layout, the "look and feel" of the document, where the former usually will be more important in a scientific context than the latter. In general, both have to be preserved. For the purpose of these considerations we will distinguish between binary and mark-up formats.

One should note, that the formats described here usually come in different versions, in particular, a file ending with .doc may be formed according to one of many different formats developed by Microsoft, which can only determined by looking into the file.

Binary formats

Binary formats are files that contain all or some essential information in a non-textual format, encoded, compressed or as tables describing relations between parts of the document. Examples are Postscript, PDF, and DVI. Word documents are partially in a binary format, as all formatting is governed by tables at the end of the document, while the pure text is essentially contained in the second part of the document behind a binary header. These file formats usually can only be displayed by dedicated software.

Mark-Up formats

Mark-up formats are essentially text files with additional information contained in the flow of the text, usually set apart by some special symbols, like the "<>"-tags in HTML. Other examples are the other SGML-based mark-up formats like XML and MathML, Microsoft's Rich Text Format, some Versions of WordPerfect files and the family of TeX versions like LaTeX, AMSTeX etc. To present the full formatting, appropriate programmes are necessary, but the text can be created with any text editor, and read if the special meaning of the mark-up tags is understood.

Purposes of file formats

While dealing with file formats for scientific texts restricts the available formats somewhat, there are still numerous options to transport the information to the user. The path and target of transportation have to be considered when choosing the appropriate format.

On-screen rendering

For on-screen-display, much lower resolution is necessary than for printing, since standard displays use about 70 to 100 dots per inch (dpi), as opposed to high quality printing which uses 600 dpi or more. This means, that even for an enlarged rendering 150 dpi will suffice to give a good impression of the contents of an article. For image-like binary file formats (e.g. PostScript or DVI), this will usually mean different versions of the given content. Mark-up formats give only the textual contents and leave the rendering to the presentation software, relying on the presence of the appropriate font in the presenting SYSTEM. This makes the result easily scalable if these fonts are present, and possibly unintelligible if not.

Printing

High quality printing requires high-resolution graphics, which are usually produced by the printer (driver) itself. The standard description language for laser printers is Adobe's PostScript, so files using this format are highly portable and can be printed on most laser printers. The format comes in different versions, some are very large but compress well, others are already compressed; some scale well, others do not.

Data exchange

There are situations, where files are not to be read immediately, they only have to be transported from one place to the other. In this situation, compressed formats might make the exchange more effective if the transport protocol provides sufficient error checking, since compressed files may become corrupted by replacing a single byte.

Discovery and Retrieval

Since scientific texts are only useful to the community if they can be found, discovery of texts is of utmost importance. This can be supported by using metadata either included in or attached to the file. If there is an efficient retrieval SYSTEM, this can help to organize and retrieve files. Many popular file formats like Word documents or PDF support such metadata, but they are rarely used in a uniform and efficient manner. Another option is the ex-

traction of text from the given file, which is often possible if the associated programme has some sort of text search functionality. PDF made from text usually is searchable, while image based PDF files or Postscript files are usually not; to extract text one has to use OCR-like methods. Extracted text can be indexed to provide some full-text search for these files.

Archiving

Archiving digital versions of scientific articles has to solve two tasks: to preserve the intellectual contents and the "look and feel" of the file for as long a time as possible (essentially "eternally"), and to make it available to the scientific community. This is less of a contradiction for digital than it is for printed material, since reading, handling or copying does not harm the digital data. Nevertheless, different file formats make preservation, handling or retrieval of archived material easier or harder.

Criteria for File Formats for Archiving

The archiving of digital material always consists of copying some form of data to some form of carrier: disk, tapes, RAM memory, CD-ROMs or any number of (emerging) data storage devices. This article will neglect the problems related to the degradation of the storage media, since this is essentially independent from the file format. In general, archives will have to provide for some kind of quality control and storage media management to prevent the loss of information. In comparison with the task of keeping different file formats alive over time this is regarded as the easier task and as essentially manageable.

Error tolerance

Since even the most careful management of media will not be able to preserve every single byte of every file, an error tolerant file format is necessary for long term archiving. For a file in any mark-up format, close inspection will usually be able to correct single letters that have been changed. For the mark-up information, automatic correction is possible since there is only a restricted vocabulary is used for this information. Binary formats, on the other hand, require first the full understanding of the underlying file format (which is not always available, see the experience with wk1 and tiff format in Gregory W. Lawrence, William R. Kehoe, Oya Y. Rieger, William H. Walters, Anne R. Kenney: Risk Management of Digital Information: A File Format Investigation, http://www.clir.org/pubs/reports/pub93/pub93.pdf, p. 14). Since the file itself is not human readable, error recognition is very hard by human inspection; the error correcting abilities of the dedicated programmes need further investigation.

Long term stability

File formats change. There is not file format, which remains the same over its life-cycle, the mathematical file formats come with somewhat exotic versioning: TeX is now used in version 3.14159 (converging to π), and LaTeX has version 2ε. PDF uses version 1.4 and is preparing version 1.5, Word uses version 8. These file formats are not immediately visible to the user (unless they are produced on an Apple Macintosh, which uses its own SYSTEM of file types), and should not require the attention of the user. Unfortunately, some rendering programs are not backward compatible, so Word 2000 will not display every Word 2 document. Adobe claims that Acrobat Reader 5 will display every PDF file, but I came across some file on the Internet requiring an older reader. Furthermore, the current programmes usually will not allow the editing of old files without changing their file format, which may create unwanted side effects. In this sense, file formats which do not change or use fully backward compatible rendering programmes are preferable for archiving.

Full open specification

If any file format is used as an archiving format, the full specification of the format should be publicly available. That means that if the programme creating this format is no longer available or stops to support this format, then it will be possible in principle to recreate the contents of this file. This will also be useful for fixing minor errors that might occur due to media deterioration, although this could only be a last resort. In general it is essential to have full control over the contents of the archived files to be able to reuse them later.

System independence

For similar reasons the file format should not tie the archive to any particular hardware platform, since these might become obsolete as well.

Ease of handling

The archive has to administer the data it is archiving, and might have to transform the file to a different format for delivery. Depending on the underlying format and the target format, this may be more or less demanding for the SYSTEM, either in terms of sheer computing capacity or in creating a specific environment used to create the desired format. Some data, and TeX files in particular, consist of several files necessary to create the desired output.

Independence of commercial interests and influence

If an institution is ready to invest in the creation and maintenance of an archive, it will be useful if their data is not only be accessible through some commercial software. For example, it is unlikely, but not completely im-

possible that Adobe will start to charge for a new version of the Acrobat Reader, needed to read PDF on some new operating SYSTEM. There are other rendering options available (e.g. GhostView), but in general this could create serious problems.

Minor considerations

While they are of less importance than the other criteria, some additional thoughts might be given to the file format for a smooth operation and the convenience to the user.

There are some file formats that store data more efficiently than other formats, so if the one format uses ten times as much data to produce the same output, then the smaller format usually will be easier to handle: for the archive, for the data exchange and for the user. Since storage space has become cheap, this is not a very important financial issue anymore.

Some file formats support some kind of navigation: a table of contents linked to the different parts of the document, a clickable index or bookmarks like PDF. This is a serious advantage in particular for long documents. In addition, there might be hyperlinks referring to Internet sites or to other documents; like the relations the CrossRef SYSTEM is building.

Since it is user-friendlier, the file format should support the option to copy a quote from a source and paste it into the target document.

File formats in Mathematics

The basic property that makes mathematics texts more complicated than other scientific articles are the formulas used which may come in almost arbitrary complexity. While some programmes are very efficient in producing these formulas, they may need special environments to display them correctly. Others are essentially independent from the SYSTEM and thus very portable. In general, problems will arise when fonts used to create the file are not present when rendering it, so the possibility to embed special fonts into the file is crucial in this approach. On the other hand, files that essentially use images are independent of the presence of fonts, but tend to scale poorly and are usually much larger than files using fonts.

TeX

Usually some version of TeX is the starting point for the creation of a mathematical text. A TeX file is not *rendered*, but *compiled* and needs a dedicated environment, including fonts (which may be created using metafont or come as TrueType fonts), macro packages (".sty"), and possibly other information like class files. The compilation of the TeX file together with its associated data provides a DVI file.

DVI

Unfortunately, the DeVice Independent file format DVI is dependent on the given environment and not easily portable. Furthermore, to render the DVI file almost a complete TeX environment is needed, something one would not want to require from a browsing reader. So usually, the DVI file is transformed to PostScript or PDF.

PostScript

PostScript is a family of file formats developed by Adobe Inc., primarily used to provide a description of the file for the printer. These files can become very voluminous, but will compress efficiently. The file should provide identical output on any PostScript-savvy printer.

PDF

This is another file format family developed by Adobe. PDF can be used as a wrapper for image files as well as a pure text file with embedded fonts. The latter version creates reasonably small files that produce very readable results in various scales. There are safety measures built into the format than can prevent the user from copying or printing the file, options which may or may not be welcome for an archive. A problem is that these properties are not immediately visible, so they may be activated inadvertently and render the file useless for archival purposes. Formulas are reproduced clearly if the necessary fonts are present.

Experiences from Math*Diss* International

The doctoral thesis is often the first major scientific article a student writes. This means that usually the experiences with the programme used to create the file are limited. While this refers to every subject area, the mathematical formulas require special attention and create additional complexity. Therefore, the Math*Diss* International project took special measures to collected dissertations in different file formats, focussing on TeX files in particular. This format is usually not handled by the universities and libraries involved in the DissOnline process that is used to create an electronic dissertation. The result is a collection of about 400 dissertations, described in the Math*Diss* Database, comprising files in all the formats mentioned above, thus giving the opportunity to compare them using the criteria for archiving. To give a complete individual evaluation or statistics is beyond the scope of this article, but some particular examples will highlight the situation.

PostScript

The largest single document is a dissertation delivered in PostScript format with a size of 159 pages and 40.099 KB, which is obviously not easily delivered via a modem connection. As a special service, we are trying to provide compressed versions of large files. In this case, the result is quite impressive: compression with zip yields a size of 950 KB, with StuffIt even 613 KB. This shows that for the delivery of documents, compression may be a very useful service to the user.

PDF

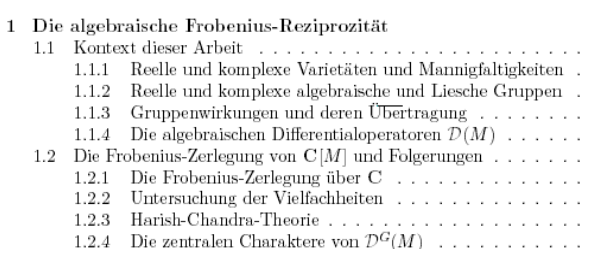
The largest PDF file has a size of 196 pages and 30.740 KB, compressing to 5.013 KB using Zip, 3.375 KB using StuffIt, and 2.514 KB using the new StuffItX format. The size seems to be due to the embedding of formulas as images, while relatively smaller PDF files would use a textual representation.

A problem in this file is that some pages do not render correctly:

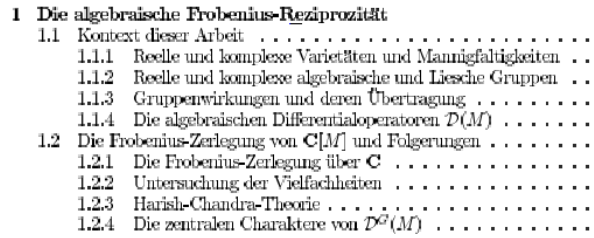
("There was an error when opening this page. The error occurred when analysing an image.") Since this only occurs when the respective page is opened, the error will not be obvious at first glance. Problems like this need either dedicated software or special knowledge about the file format, which sometimes may allow to fix the file using a text editor or a hex editor. In any case, the amount of time required makes this prohibitive on a large scale.

The rendering of PDF files delivered to the archive differs greatly, making them more or less readable on-screen. Since some dissertation actually come in various formats, a direct comparison is possible.

The following version was created with pdftex:

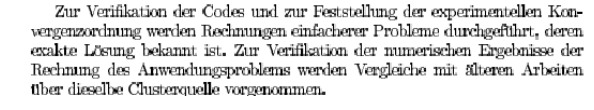


and this with dvipsk:

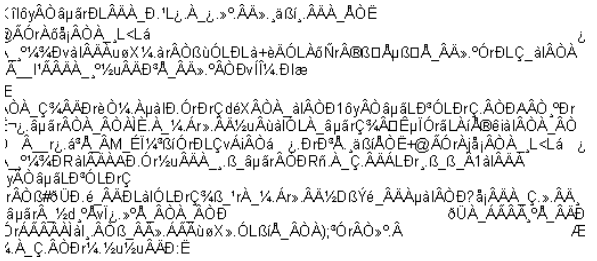


The second version is hardly legible on-line.

A final remark to PDF refers to the copying of text. In PDF, copying as images is always possible unless expressly forbidden by the security information:


Zur Verifikation der Codes und zur Feststellung der experimentellen Konvergenzordnung werden Rechnungen einfacherer Probleme durchgeführt, deren exakte Lösung bekannt ist. Zur Verifikation der numerischen Ergebnisse der Rechnung des Anwendungsproblems werden Vergleiche mit älteren Arbeiten über dieselbe Clusterquelle vorgenommen.

On the other hand, copying of text may be allowed, but produces useless results. Here is a copy of the text above:



Again, it is not clear from the outside, which files allow copying and which don't.

TeX

The basic advantage of TeX is its text-based file format with additional mark-up, with the program and its additions available as open source software through a distributed SYSTEM of CTAN-archives. Unfortunately the compilation process needed to turn a TeX file into a more readable form does not always run smoothly. The packages delivered to the archive might be incomplete, referring to macros not installed, using fonts not available or any other number of problems. Since no automated checking is installed in the Math*Diss* database, these problems will usually go unnoticed.

There are dissertations, consisting of a single TeX file and some additional macro (.sty) files, that compile perfectly, producing 100 pages of mathematics. On the other hand, the most complicated package includes 74 files: eight files with no ending, including three makefiles, one .aux, one .bak, three .bc, one .fot, eight .hex, one .jpeg, two .log, three .make, three .md, one .meta, four .mi, 30 .pre, seven .psi, one .tx. The actual dissertation is hidden in the .bak file. Needless to say that this collection will not compile easily. Most of the TeX packages in the archive comprise 30 to 60 files, often needing additional macros.

Experiences from the EMANI project

The EMANI project (see http://www.emani.org/) is a cooperation between four libraries, the European Mathematical Society and Springer Verlag. The goal is to provide retrodigitised copies of older and digital versions of new Springer Journals in a reliable fashion. In this context, TeX files from Springer Verlag were considered as potential archival material.

In some sense these files are at the opposite end of the TeX spectrum to the dissertations: these are highly professional and essentially standardised texts, based on the macros provided by Springer Verlag (see http://www.springer.de/author/tex/help-journals.html, note that not all browsers support the ftp protocol used for the packages).

These packages are provided for each journal, although not necessarily different. Analysing the files for "Numerische Mathematik" showed that the following files are needed to compile the articles:

- svjour.cls: a general class file definition for Springer Journals
- svnummat.clo: a special class option file for "Numerische Mathematik"
- TOTAL00.NUM: a somewhat obscure file that "redefines the things for journals to produce totally camera ready output"

So these files are: a general file for all Springer journals, a specific file for the particular journal, and an additional file for the volume (or some other special sub series) of the journal. In an archival context, these files have to be present when the TeX file is compiled, thus this information has to be included in the metadata describing the file, and the various files have to be archived as well. This requires some sort of management to bring the correct files together in the event of compilation. These additional files should be kept separately from the articles and should be stored with appropriate versioning.

Conclusions

PDF

The collection of the dissertations shows that PDF is a very convenient format for handling electronic documents. The basic advantage is that one single document can contain the whole contents of the dissertation and be rendered using one programme without additional requirements from the computing environment. Disadvantages are that

- some files can become very large,
- error tolerance is rather limited,
- the acrobat SYSTEM is owned by Adobe and is not open source.

Nevertheless, the file format specification is publicly available and is actually used by third party developers to create PDF-based software. So for the creation as well as the rendering of PDF files, alternative programmes are available.

Additional care has to be taken when incorporating new files into the archive.

- The general integrity of the file has to be checked.
- The security settings have to be set to what they should be. For example, the document should not prohibit printing.
- The document should give optimal rendering while viewing on the Internet as well as for high quality printing.
- The document should allow for copying if possible.

For large scale archiving these tests have to be run automatically, and it is not clear if this is possible for each of them. Probably additional software has to be developed to support the archiving process.

TeX

The strengths and the weaknesses of TeX are the opposites of the PDF properties. The major disadvantage is the multiplicity of files needed for one document and the dependence on an environment with the right macros, classes, fonts etc. In addition, the file cannot be rendered directly but needs compilation and then still is only available in DVI format. On the other hand, error tolerance is high, the files are relatively small, and the whole SYSTEM is built on open source software.

As a basic help to create correct TeX, the Math*Diss* International project has set up a reference SYSTEM for TeX (see http://www.ub.uni-duisburg.de/mathdiss/eform2.html), including a start file and a bibliography style. This can be a starting point for standardising TeX for dissertations, which would make archiving files much easier and more predictable.

Still, for the acceptance of a document into the archive, several tests should be passed:

- Are all files that are not part of the standard installation (e.g. images) included in the packet?
- Are all files contained needed, or are there some old leftovers around from earlier compilations?
- Does the file compile and create correct DVI files?

Again, the question is if these properties can be checked automatically.

To make the TeX format an effective competitor to PDF for archiving purposes, the handling of TeX files has to be improved. The IBM techexplorer is already a tool that tries to render a single TeX files, unfortunately not with enough precision, obviously because the programme has no way to know where all the components of the document are. So a packed format would be use-

ful for TeX, that would make one file from all the needed components and could be fed to some reader like techexplorer that would display the document. Although this might not be the most effective way for journals that use all the same additional files, for dissertations with their idiosyncrasies this would be an ideal solution. The libraries handling the electronic dissertations would only be confronted with one file that could be read immediately, so they would be in the same position as if the file were PDF. Since modern computers are much faster than at the time Donald Knuth developed the TeX SYSTEM, the rendering of the file should be essentially immediate. This would also prevent the separation of the main file from its auxiliary ones, thus the other main problem of TeX files would be solved: files not compiling would immediately be recognized and could be fixed on the spot. In addition, some method of incorporating additional files like fonts would allow for maximal portability.

Still the whole package would be fairly small compared to the PDF file (depending on compression schemes) and retain all the advantages of a text based mark-up SYSTEM.