```
\centerline{\titlefont\title}
\medskip\centerline{by}\medskip
\centerline{\bf C.G.\ van der Laan%
 \footnote*{Hunzeweg 57, 9893PB,
  Garnwerd, The Netherlands.
  email: cgl@rug.nl.}}
\vskip3cm
\setcornerrules
\vskip9cm%paste-up illustration
\vfill\centerline{\issue}\vfill
\eject
%inside cover
\null\vfill
\centerline{\bf Abstract}
\par{\smallskip\noindent\abstract
    \smallskip}\par\vfill
\noindent{\bf Keywords}: \keywords.
\vfill\eject
%
\pageno1
\footline{{\fiverm\rlap{\date}\hfil
 --\folio--\hfil\llap{\copyright cgl}}}
\endinput                  %cgl, March 93
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A table of contents can be made via extracting the chapter and section titles via the use of a programmable editor: extract the lines which contain `\beginchapter...`, and `\beginsection...`, store these, and add

```
\bgroup\beginchapter{} {}{}.Contents\par
\def\beginchapter#1 #2#3. #4\par{\par
 \noindent#2#3 #4}
\def\beginsection#1\par{\item{--}#1}
\input toc
\egroup
```

The above requires that the titles are ended by `\par`, and all one-liners. To circumvent extra editing there should be no copy on the same line after the `\par`. The contents as supplied below has been created in this way.

When page numbers are also wanted, we have to write the entries to a file, and let the OTR do the work.

Appendix C: Contents

AMS BLUes

professionals at work

Kees van der Laan

Hunzeweg 57
9893 PB, Garnwerd, The Netherlands
cgl@rug.nl

Abstract

The significance of the American Mathematical Society for the TEX community at large, and more general the leading role of the AMS in the area of professional computer-assisted typesetting, is praised.
AMS-TEX, AMS-LATEX, their accompanying styles `amsppt.sty`, respectively `amsart.sty`, as well as AMSfonts are discussed. AMS provides excellent user's and installation guides along with the software and fonts, all in the public domain. Despite the quality, an alternative approach—and in detail some alternative encodings—are provided.
A publisher is strongly encouraged to take notice of the computer-assisted publishing activities of this pace-setting society.
A new procedural idea with respect to specifying and formatting bibliographies, given a background file of all the references an author is familiar with, is proposed to suit the author and the publisher.

Keywords: Computer-assisted typography, math, bibliography, mark up, typesetting, AMS-(LA)TEX, (math and cyrillic) fonts, plain TEX, macro writing, education.

Introduction

This is the second paper in the series about general, public domain macro collections to format complex math-oriented documents. AMS-TEX has been developed by Michael Spivak. Frank Mittelbach, Rainer Schöpf, and Michael Downes from AMS, continued the project with AMS-LATEX, in the early nineties.

Next to these basic macro collections some styles have been designed to mimic the layout of the various AMS publication series. For an author these are even more important.

For a survey of how the production process goes at AMS, see Youngen, 1991. At the moment it is difficult to tell what belongs to a certain style and what is part of the underlying AMS-TEX. It is alive, and. . . boundaries are moving.[1]

Reality has it, that the AMS has succeeded in maintaining the high-quality of the appearance of their publications[2] while adopting new TEXnology. My wishful thinking is

> To maintain the same appearance in print, independent of whether the copy has been marked up via `amsppt.sty`, that is AMS-TEX, or via `amsart.sty`, that is AMS-LATEX.[3]

There are two author interfaces

> TEX- vs. LATEX-oriented.

Although both are aimed at mimicking AMS traditional layout, they yield different results in print, especially the amount of white space, and the numbering of theorems and the like (see the templates in the Appendixes B and C). They don't provide the same functionalities either. That LATEX's goodies are taken for granted is understandable. It is curious why the underlying `amstex.tex` etc. formatting has not been used for the other parts. For the mark-up of bibliographies BiBTEX can be used, next to LATEX's bibliography tool, instead of interfacing to `amstex.tex`'s `\Refs...\endRefs` et cetera.

That the similarity has not been attained completely is understandable. LATEX has its conventions, which

[1] As a rule of thumb the following from `amsppt.doc`: 'For the most part the internals of `amsppt.sty` aren't closely related to `amstex.tex`: `amstex.tex` handles mathematics, `amsppt.sty` handles visual design and overall structure of documents.' LATEX too did not define clearly what should be part of the sty-file and what not. Indeed sty-files are provided but also files which handle the sizes of the fonts.

[2] The contents is not the issue here.

[3] At the moment this is not the case.

are hard to supersede, I suppose. A balance had to be found, which does justice to TeX and LaTeX. It is subject to change into a more abstract and more functionally oriented author interface. The latter is perhaps my wishful thinking.

For the author the `amsppt.sty`, respectively `amsart.sty`, has emerged as the style to work with. AMS staff will take care of the details needed to typeset the electronically prepared manuscript via the other styles.

In the sequel I'll treat

- What is provided by AMS? The tools, the documentation, the services, and the philosophy, next to the hardware requirements
- AMS-TeX, `amsppt.sty`, the Joy of TeX's math, and how to access fonts
- AMS-LaTeX, `amsart.sty`, some remarks about LaTeX's math, and how to access fonts
- AMSfonts, what is provided in this collection, how to load and access
- Appendixes.
  - A: How to get it?
  - B: A rough template of an author script with respect to `amsppt.tex`.
  - C: A rough template of an author script with respect to `amsart.sty`.
  - D: Contents (the table of contents of this article).

It is in the nature of this paper that it is hard to read, because of the many issues treated—plain, AMS-TeX, LaTeX, BibTeX, AMS-LaTeX, AMSfonts, NFSS—on the one hand, and because of the necessary amount of detail on the other hand. The parts AMS-TeX and AMS-LaTeX are independent from each other; in both the use of the AMSfonts collection is discussed. A BLU type reader is encouraged to grasp the essentials from the abstract, the conclusion, or the titles and examples of the main sections.

> If only BLUe would start reading one of the AMS Guidelines, or gaze at the templates provided in Appendix B and C, I have reached my goal already.

What is provided by AMS?

Apart from the general goodies of publishing with the AMS, they provide TeXnically

- Software.
  As basis collections there are AMS-TeX, and AMS-LaTeX. The next layer is formed by the styles `amsptt.sty`, respectively `amsart.sty`, with variants for books, bulletins, journals,. . . . One-column format is used throughout. The encodings are highly systematic and consistent.
- Documentation.
  Especially useful are the 'Guidelines for Preparing Electronic Manuscripts.' One for AMS-TeX-oriented authors, and a similar one with respect to AMS-LaTeX. The main purpose of these booklets is to provide authors with know-how about mark-up and to inform them about procedural aspects, such as submission rules. These papers contain much worthwhile to know for preparing electronic manuscripts in general—if not for the samples in the appendixes— although biased towards AMS-TeX. Then there are the user's and installation guides. All very-well done!
- Fonts.
  The collection is called AMSFonts, version $\geq$ 2.1. There is a user's and an installation guide.
- Support.
  The collections are professionally maintained. No volunteers! email: tech-support@math.ams.org.

Hardware requirements. Hardly none. Just some ten-s of MBytes for the fonts. AMS-LaTeX works best with a big TeX, because of the many control sequences in use.

Not in there. Remarkable is that after so many years of TeX being around, AMS is modest and realistic in its use of it. AMS does not deal yet with hard issues like cost-effective electronic inclusion of graphics, and the preparation of tables. It is left to the author to do it by (LA)TeX, or by other means. Indexing tools are not the issue for the `amsppt` and `amsart` styles. There are no facilities to include files verbatim in AMS-TeX (Not so much needed, given the context.) In AMS-LaTeX an option is available which provides Schöpf's `verbatim.sty`.

Believe it or not. AMS-(LA)TeX, especially its accompanying customized `amsppt.sty`, is the 'camino royal' for every AMS author.

> Reality[4] has it, that `amsppt.sty`, respectively `amsart.sty`, gets more and more the role of the mark-up tool for each and every AMS author, with the AMS staff taking care of the finishing touch.

The latter involves solving hard mark-up problems, or more commonly, the handling of the details required by the style for the concrete publication series, or laying hands on a special symbol provided within the AMS-fonts collection.

[4] To be honest my wishful thinking.

This approach can be illustrated by the following scheme[5]

author(TeX)	author(LaTeX)
↓	↓
`amsppt.sty`	`amsart.sty`
↓	↓
`amstex.tex`	`amstex.sty` → LaTeX
↓	↓
TeX	TeX

Alternative approach

In my opinion a better company policy would have been to allow for a LaTeX author interface, next to the TeX one, as depicted by the following scheme.

author(TeX)		author(LaTeX)
↓		↓
`amsppt.sty` ⟶		⟵ `amsart.sty`
	↓	
	Tools	
	↓	
	TeX	

I omitted on purpose `amstex.tex`, because I consider the math mark-up as proposed there, as syntactic sugar of what has already been provided by plain. The tools are the macros provided by the TeX community at large, for example PiCTeX, (Bordered) Table macros, Commutative diagrams macros,. . . , see the Jones index for a survey.

Notations `\ea`, `\nx`, and `\ag`, are used as shorthand notations for `\expandafter`, `\noexpand`, respectively `\aftergroup`. Now and then 'we' is used, especially when text has been borrowed from documentation provided by AMS. Read in those cases 'AMS.'

AMS-TeX

If only the reader would now turn over to the excellent 'Guidelines for preparing electronic manuscripts—AMS-TeX,' much would have been gained.

1 The AMS-TeX package

The AMS-TeX package consists of

- amstex.tex, amstex.doc
- some styles, notably amsppt.sty, amsppt.doc
- amssym.def, amssym.tex, with symbol definitions

along with the documentation[6]

- AMS-TeX User's Guide
- AMS-TeX Installation Guide
- Guidelines for preparing electronic manuscripts—AMS-TeX.

In this part I'll treat

- amsppt style
- some math mark up details, and
- the use of AMSfonts.

1.1 Styles

A document is thought to consist of[7]

- preliminary part,
- top matter, and
- the document.

The preliminary part contains

```
\input amstex
<declarations and font loading>
\documentstyle{amsppt}
```

The top matter consists of the title, author and the like information. See the user's guide.

The document itself consists of `\head...\endhead`-s. Nested therein the `\subhead...` and `\subsubhead...`-s, and paragraphs. Appendixes are just `\head... \endhead`-s. At the paragraph level the plain and AMS-TeX (math) constructs can be used. The section about formatting references is special.

Strong points

- Structuring macros have explicit ending macros[8]
- Very-well developed top matter material
- A fancy bibliography tool[9]
- Elaborated context related error detection, correction, and messages
- Access to the rich AMSfonts collection
- The syntax-checking-only run mode has been made available, via `\syntax`, with confirmation interactively.

Not in there

- No automatic cross-referencing
- No full-fledged 14pt for title matter
- No automatic numbering schemes.

1.2 Encoding

With respect to the encoding I will discuss the systematic use of control sequences `\nofrills`, `\title`,

[5] Neglecting the AMSfonts files.

[6] In booklets and also available in electronic form.

[7] A more detailed structure is provided in the template in Appendix B: `amst-art.tem`.

[8] In the spirit of SGML's complete mark-up.

[9] The marked up fields can be supplied in arbitrary order, preceded by just an opening tag and no explicit closing tag, in the spirit of SGML's minimal mark-up.

`\head`, `\theorem` and the like,[10] and the mark up of the list of references. Furthermore, I will discuss the used font selection scheme.

Encoding: `\nofrills`

Purpose. `\nofrills` is a general optional parameter to overrule the default typesetting. From the example in the user's guide it turns out that the user *needs* it for appendixes, where `\chapter` has to be used and a caption different from CHAPTER is needed, via for example

```
\chapter\nofrills{Appendix B}...
```

or in general when captions in another language are wanted, for example

```
\Refs\nofrills{Literatuur}
```

For several macros, e.g., `\title`, we want to allow the user the following two options

```
\title ...\endtitle
\title\nofrills ...\endtitle .
```

In the first case, we want the text of the title to be stored in a box for later use. In the second case, we want certain parts of the typesetting to be omitted when constructing the box. Internally this is done by prefixing them with `\frills@`, which in the normal cases is simply an identity function (`\let\frills@\identity@`), whereas in the nofrills case it gobbles its argument (`\let\frills@\eat@`).

In order to test for `\nofrills` we have to use `\futurelet` (because if `\nofrills` is absent, we don't want to do anything that might affect the actual text of the title, such as stripping braces or adding an unwanted extra pair around the first token or group).

Use. `\nofrills` must immediately follow the control sequence it applies to. For example

```
\title\nofrills...\endtitle .
```

Typical use: if we want to execute a macro called `\title`, but check to see if `\nofrills` is coming up next, we do this via

```
... \nofrillscheck\title
```

If indeed the next thing is `\nofrills`, then we will execute `\nofrills@@` before calling `\title` again. This involves some shuffling around in order for us to get back the 'real' definition of `\title` afterwards and yet still be able to use a control sequence named `\title` to read the argument.

> This is important in giving the user a useful error message if they accidentally omit or misspell the matching `\endtitle` or whatever.

We could use instead the trick used with `\align`[11] of adding a space onto the `\csname` with which we read the argument, but that would increase the use of hash size.

Design. The general encoding mechanism for `\nofrills`, to be applied to the generic \\$\langle tag\rangle$, reads

`\def`\\$\langle tag\rangle$`{\let\savedef@`\\$\langle tag\rangle$
`\def`\\$\langle tag\rangle\langle parameters\rangle$`{\let`\\$\langle tag\rangle$`\safedef@`
$\langle code\ of\ tag\ for\ default\ formatting\rangle$`}`
`\nofrillscheck`\\$\langle tag\rangle$`}` .

The effect of the above template is that the user's invoke of \\$\langle tag\rangle$ is replaced by

`\nofrillscheck`\\$\langle tag\rangle$

with \\$\langle tag\rangle$ temporarily redefined. In this redefinition use is made of

- `\next` equals the token after \\$\langle tag\rangle$
- `\next@` equals \\$\langle tag\rangle$, and
- `\frills@` equals either `\eat@` or `\identity@`

with `\eat@` the function to gobble its argument and `\identity@` the identity function.

The code. From the file `amsppt.doc` I borrowed the following.

```
\def\nofrillscheck#1{\def\nofrills@{%
 \nofrills@@{#1}}\futurelet\next\nofrills@}
%
\def\nofrills@@#1{%
%This extra step in defining \next@
%is a precaution in case #1
%might be more than one token.
%Parameters use:
%\next contains the 'token after the
% item token,' e.g. \nofrills
%\next@ contains the item token, e.g.
% \title
%\frills@ is either \eat@ or \identity@
 \DN@{#1}%\def\next@{#1}
 \ifx\next\nofrills \let\frills@\eat@
   \ea\ea\ea\next@\ea\eat@
 \else\let\frills@\identity@\ea\next@\fi}
```

Explanation. The purpose of the redefinition of \\$\langle tag\rangle$ is to supply as argument of `\frills@` the process to be executed, and to give `\frills@` the appropriate meaning.[12]

`\savedef@` stores the original \\$\langle tag\rangle$ definition. The next redefinition reestablishes the original definition, and defines `\frills@` either as `\eat@` or `\identity@`, given the values of `\next` and `\next@`. At the end it invokes `\nexta`. `\nofrills` is eaten when present. And that is about it.

[10] With `\add@missing`—the former `\runaway@`—as the general error detecting (and correcting) macro.

[11] Spivak, 1986.

[12] This is a discipline for the encoders and not at all for the authors. The authors are ignorant about this. The encoding team creates in the style file from the template versions for every control sequence which allows for an optional `\nofrills`.

1.3 Encoding: `\title`

The title control sequence is typical for the top matter material. Once it is understood, the other control sequences won't provide mysteries anymore.

Purpose. The purpose is to typeset the title, default in upper case.

The code. From `amsppt.doc` the following.

```
\newbox\titlebox@
%
\def\title{\let\savedef@\title
%We use a \vtop here because we want to
%know the height of the first line of
%the title when we start typesetting
%the topmatter. In order to get the
%correct sinkage from the top of the page.
 \def\title##1\endtitle{\let\title
  \savedef@\global\setbox\titlebox@
  \vtop{\tenpoint\bf\raggedcenter@
%Increased \baselineskip is because of the
%uppercasing.  We do it like this instead
%of putting it inside the \uppercasetext@
%macro, because it only applies where an
%entire paragraph is made up of
%upper case text.
  \baselineskip1.3\baselineskip
  \frills@\uppercasetext@{##1}\endgraf}%
%Check to see if right and left hand
%running heads have been already assigned
%by the user---if so, don't override.
 \ifmonograph@\edef\next{%
                     \the\leftheadtoks}%
  \ifx\next\empty@\leftheadtext{##1}\fi
 \fi
 \edef\next{\the\rightheadtoks}
  \ifx\next\empty@
               \rightheadtext{##1}\fi}%
 \nofrillscheck\title}
```

Explanation. For completeness I also borrowed from `amsppt.doc` the following explanation. We store the title in a box using a `\setbox`. It could also be done by defining `\thetitle@`.

> One advantage of `\setbox` is that syntax errors within the title (say for math) are reported immediately instead of during the processing of `\endtopmatter`.

The title box is always put on the page by `\endtopmatter`, even if it's empty. (With other pieces of the topmatter we check first to see whether there's any text to typeset and skip to the next item if not.) The text of the title is stored up for the running heads, unless `\rightheadtoks` is nonempty—then presumably the user has already used `\rightheadtext` to set the right-hand running head, in which case we leave it unchanged. The default for titles is uppercasing. From a design standpoint it might be preferable to do titles in 14-point text with initial caps, but since titles may potentially contain any kind of math, doing this would require a full-fledged `\fourteenpoint` analogous to `\tenpoint`. So to give the title extra prominence we use uppercasing instead. The uppercasing can be removed by `\nofrills`.

Alternative encoding. For the title part the use of `\nofrills` comes down to override the default (upper case) typesetting. But if that is all what has to be done, we can simpler provide the author with the possibility to provide the name (and eventually mark-up) in `\titlename`. If the latter control sequence remains empty, then the user apparently accepts the default. The encoding is simple: just test for the emptiness of `\titlename`.

Furthermore there is the general question about end punctuation.

> Why not parameterize over the end punctuation as well? For example via `\titlepunctuation`?

If the user wants his own end punctuation, the latter control sequence can just be redefined. This could have been a general design choice—parameterizing over the end punctuation—eventually in a global way per level.

Remark. Practice has it that publications take long titles, despite the advice to keep them short. In the headlines of the paper a so-called (short) running title is supplied. In AMS-TeX this is done for the right pages.[13] AMS provides however the possibility to supply next to the title, an optional short title in `\righthandtext`, and ipso facto for author names in `\lefthandtext`.

1.4 Encoding: `\chapter`

Nested within the title part we can have `\chapter`, again with the `\nofrills` option. For example

```
\title\chapter\nofrills{Appendix B} The
 Poisson Integral\endtitle .
```

Especially when the publication is a monograph.

In `amsppt.sty` an appendix heading is marked up via

```
\head*{} Appendix*.
```

This shows that it is at least confusing to treat `amsppt.sty` along with the book style in one booklet, and also to have one code which accounts for both.

Purpose. The function is to typeset the chapter keyword—or the one supplied by the user (Appendix, for example)—and to typeset the supplied chapter 'number.' That is all, and one can question whether it can't be attained in a simpler way.[14]

[13] The author name(s) are used in the running heads for the left pages.

[14] There is no `\endchapter`, but as used it is not a structuring control sequence.

The code. From `amsppt.doc` the following.

```
\def\chapter{\let\savedef@\chapter
 \def\chapter##1{\let\chapter\savedef@
 \leavevmode\hskip-\leftskip
%Put the \chapter stuff in an \rlap so
%it doesn't affect centering of the title,
%and in an "uplap" so it is placed above
%where we want it.
 \rlap{\vbox to\z@{\vss
  \centerline{\eightpoint
%We do a baselineskip of 2pc from the
%base of the "CHAPTER" banner to the base
%of the first title line. (The baseline
%of the \null will coincide with the
%baseline of the first title line.)
   \frills@{CHAPTER\space\afterassignment
    \chapterno@\global\chaptercount@=}##1%
    \unskip}\baselineskip2pc\null}}%\vbox
  \hskip\leftskip}%end \rlap
 \nofrillscheck\chapter}
```

Explanation. Let us trace what happens after `\chapter\nofrills....`

First, the chapter definition is copied via

`\let\savedef@\chapter`

then `\chapter` is redefined, with a parameter, with the function to provide the default typesetting of the chapter keyword and the (supplied) number. The check for frills is done by the invocation of

`\nofrillscheck\chapter`.

Because our copy contained `\chapter\nofrills...` `\next` is defined as `\nofrills`, and `\nofrills@@{ \chapter}` is invoked. In the latter macro `\DN@{\chapter}` is processed, and then via an ingenious use of `\ea`

- `\nofrills` will disappear (is eaten),
- `\next@` is invoked (which has been `\let`-equal to `\chapter`), and
- `\frills@` will become `\eat@`,

and therefore the provided chapter-marked-data (read: alternative keyword) will be typeset in the copy. Quite something isn't it?

Alternative. At the basis of the alternative encoding for `\chapter` are the following questions.

> Why not have a control sequence `\chapterno`, that provides the contents (that is the number or something similar), and a control sequence `\chpkeyword`, for the keyword?

This could lead to the following alternative encoding, without the use of the optional parameter mechanism as encoded via `\nofrills`[15]

```
\def\chapter{\leavevmode\hskip-\leftskip
 \rlap{\vbox to\z@{vss\centerline{%
  \chapterkeywordandno}%
  \baselineskip2pc\null}}\hskip\leftskip}
%
\def\chapterkeywordandno{\eightpoint
 \chpkeyword\space\chapterno\unskip}
%
\def\chpkeyword{CHAPTER}
\def\chapterno{11}
```

In the above alternative the inner details of the chapter keyword and number encoding can be overridden by the author, for example another typeface can be used.[16] Of course one could decide to freeze the used typeface too. The point I like to make is, that the above is at least as clear—if not clearer—with the same functionality.[17] It is true however, that for an encoding-team manager it is more difficult to verify whether the same encoding discipline has been adhered to by every team member. Perhaps it must be emphasized stronger what should be the function, along with (inescapable) guidance about the discipline to be adhered to.

1.5 Encoding: `\head`

The special thing here is to prevent page breaks between `\head` and an immediate following `\subhead`, or between a `\subhead` followed by a `\subsubhead`, and the like. It is in general still an unsolved problem, as follows from the following from `amsppt.doc`.

In accordance with conventional design principles, the space below headings is not given any stretchability or shrinkability. Since we often want to do a penalty and vskip, and since there are extra complications involved if there is a preceding vskip from something else, we define a macro to do it. Normally it is used while we're working on the main vertical list, so we have to use `\removelastskip` (which does a negative skip) rather than `\unskip` (which *really* removes the last skip). If the last thing on the main vertical list is anything other than a vskip, say a penalty from `\pagebreak`, the value returned by `\lastskip` will be 0, but the potential complications are a whole other subject. This macro handles the straightforward cases.

In some cases we may not want to put a penalty at all. We refrain from doing any penalty if the first argument of `\penaltyandskip@` is 0; the essential effect of an explicit `\penalty0` can be gotten by doing a penalty of 1 instead.

We use `\penalty@` instead of `\penalty`, so that we can redefine `\penalty@` in `\nobreak`, to prevent page breaks between certain pairs such as `\head \subhead` or `\head \proclaim` or `\nopagebreak \proclaim` —for example, in

```
... which leads to this theorem:
\proclaim{Theorem 8.2} ...
```

[15] I understand that some meta-ness of encoding has been striven after. When it has to be changed only the metacode needs adaptation. I doubt it, whether it has worked out this way in reality. It is so intertwined.

[16] The end punctuation in other cases.

[17] By the way, it is such a simple task, just the keyword and the number! If that has to be handled in such a complex way, we are on the wrong track.

a page break after the colon would be bad, and a user might want to be able to add a \nopagebreak.

The sequence

```
\subhead Text...\endsubhead
\subsubhead Text...\endsubsubhead
```

will still have the weakness of allowing a pagebreak between the two headings, because \endsubhead doesn't do a \nobreak (since it's a run-in heading). If the next piece of the document after \nobreak is something like \subhead or \proclaim that calls \penaltyandskip@, then \penalty@ will reset itself in the way that we want. But if not, then we still want to reset \penalty@; so we use \everypar.

> There are probably some unusual cases that will still have problems, *but at the moment this is the best solution we have.*

We equate the old form of the headings to the new form, for backward compatibility. It's easier to do this now rather than later because \head and \subhead are going to be outer. The purpose of \restoredef@ is to work around problems caused by the outerness of things like \subhead. \relax at the beginning prevents the \savedef@ (which may be \outer) from being read prematurely in certain kinds of expansion.

```
\def\restoredef@#1{\relax\let#1\savedef@
 \let\savedef@\relax}
```

\subhead and \subsubhead are simpler, though they allow \nofrills. (\head, being centered, does not have automatic punctuation put in at the end and so \nofrills doesn't have anything to do.) The syntax of \subhead is changed from \subheading{...} (version 1) to \subhead...\endsubhead (version 2). This was done for the following reasons

- to be consistent with \head...\endhead (we could have changed \head instead of \subhead, but the \x...\endx syntax is the one currently in use in AMS production);
- if someone (perhaps us) ever wants to do something tricky with the headings, having the \end... may help them avoid technical complications.

1.6 Encoding: theorems, proofs, definitions, remarks

These structures all allow for the same mark-up scheme
\⟨*tag*⟩{⟨*heading text*⟩}
⟨*general text*⟩
\end⟨*tag*⟩

With for example as heading text

Lemma 1.	in case of ⟨*tag*⟩ equals proclaim,
Proof	in case of ⟨*tag*⟩ equals demo,
Example 5	in the case of ⟨*tag*⟩ equals example,

and the like.

Alternative

I find it very unnatural to have to provide

```
\demo{Proof}...
\endemo
```

and the like. Why not provide the structures, the user is familiar with, by their names? For example

```
\proof...
\endproof
```

The endcoding can be parameterized over \proofkeyword{Proof}, or something like that? Eventually as alias? This should all be done behind the scenes, and not bother the author.

Intermezzo *(Error detection and correction)*

For these environments a general error detection scheme has been encoded, based upon the concept of an environment stack. This stack keeps track of which environment the formatting process is in.

The environment stack \revert@ is a toks variable. \environ@stack stacks its argument in \revert@ and also as replacement text of \environ@end.[18] When an environment is ended the invocation of \revert@envir, with the appropriate argument, restores the previous \envir@end—pops up the stack—via the invocation \the\revert@.

The code. From amsppt.doc I borrowed the following. \revert@ and \envir@stack are for use by any environments that don't enclose their text in a group, e.g., \proclaim, \definition, \roster.

```
\newtoks\revert@
%
\def\envir@stack#1{\toks@\ea
 {\envir@end}
 \edef\next@{\def\noexpand
  \envir@end{\the\toks@}\revert@
  {\the\revert@}}%
 \revert@\ea{\next@}%
 \def\envir@end{#1}}
%
\gdef\revert@envir#1{\ea
 \ifx\envir@end#1\the\revert@
 \else\ifx\envir@end\enddocument
         \Err@{Extra \string#1}%
      \else\ea\add@missing
       \envir@end\revert@envir#1%
      \fi
 \fi}
```

Explanation. The expansion in \envir@stack is difficult to understand. So let us go through it step by step, where I use \end⟨*tag*⟩ as a generic name for the end tag. The replacement text of

```
\envir@stack\<endtag>
```

[18] The programmer can easily verify in what environment he is in, by inspection of \environ@end.

after expansion is

```
\toks@{\<current endtag>}
\def\next@{%
        \def\envir@end{\<current endtag>}
 \revert@{\<current revert@}}
%Substituting \next@ yields
\revert@{\def\envir@end{\<current endtag>}
 \revert@{\<current revert@}}%(1)
\def\envir@end{\<endtag>}    %(2)
```

This can be read more easily as

1. \revert@ has as replacement text the definition of the $\langle current\ endtag\rangle$ together with the (current) stack assignment
2. \envir@end has as replacement text the \end$\langle tag\rangle$.

The curious thing is that the \end$\langle tag\rangle$ is stored, and the *definition* is stacked, next to the stacking of the \revert@ assignment. The *separation of concerns* principle of programming has it, that code is easier to read and maintain, when the different tasks are separated. For the concrete situation at hand this means that the stacking and (re)definition can better be separated.

To complete our understanding, let us follow the 'pop-up' process step by step. After the above, the replacement of

```
\revert@envir{\<endtag>}
```

after expansion reads

```
\def\envir@end{\<current endtag>}
\revert@{\<current revert@>}
```

For arguments different from \end$\langle tag\rangle$ an error message will be supplied and the missing \end$\langle tag\rangle$ will be inserted via the invocation of \add@missing{\<endtag>}.

The error detecting (and correcting) macro reads

```
\def\add@missing#1{\ea\ifx\envir@end#1%
 \Err@{You seem to have a missing or
  misspelled \ea\string\envir@end ...}%
%It is useful to supply the necessary
%missing piece, especially in the case of
%\endref.
 \envir@end
\fi}
```

Alternative. At the heart of the alternative lies the question
'Why not use DEK's general \leftappend and \lop, TEXbook p.378, instead of \envir@stack, respectively \revert@envir?'

I'm not saying that DEK's macros are easier to understand. But, once mastered, they are more general—a double ended queue, de-queue for short, has been implemented, instead of a stack—and therefore can be used for other cases as well. Of course DEK's macros can be tailored for this special situation. This, together with not stacking the definition, yields as alternative

```
\newtoks\revert@%the stack variable
\revert@{\empty}\def\envir@end{\empty}
%To push
\def\envir@stack#1{\ea\ea\ea\revert@
 \ea\ea\ea{\ea\envir@end\the\revert@}
 \def\envir@end{#1}}
%To pop up
\def\revert@envir#1{\ea\pop\the\revert@.}
\def\pop#1#2.{\revert@{#2}%
        \def\envir@end{#1}}
```

Explanation. In pushing the stack \revert@ is left appended by the replacement text of \envir@end, and the argument of \envir@stack is stored as replacement text of \envir@end. In popping up the stack the first token of \revert@ is stored as replacement text of \envir@end, and the rest, also called tail, is (re)stored in the stack toks variable \revert@.

I have used the same names as those of AMS, but general names, related to a stack of tokens, would have been more appropriate.

End intermezzo

The code. After the above intermezzo we can understand the code, and the error detection and correcting parts of it, more easily. As usual with AMS the other environments are similarly encoded.
From amsppt.doc I borrowed the following as example.

```
\outer\def\proclaim{%
 \let\savedef@\proclaim\let\proclaim\relax
 \add@missing\endroster
 \add@missing\enddefinition
 \add@missing\endproclaim
 \envir@stack\endproclaim%push on the stack
%penalty-100 is the penalty amount used by
%plain.tex's \medbreak.
 \def\proclaim##1{\restoredef@\proclaim
  \penaltyandskip@{-100}\medskipamount
  \varindent@
  \def\usualspace{{\proclaimheadfont@
   \enspace}}\proclaimheadfont@
   \ignorespaces##1\unskip\frills@{.%
   \enspace}%
  \proclaimfont\ignorespaces}%
 \nofrillscheck\proclaim}
```

Explanation. First, the usual alias is created. Then, because of the outerness, it is \let-equal to \relax. Then via the various \add@missing\... it is verified whether we are still in one of those environments. The name \envir@end contains the \end$\langle tag\rangle$ of the current environment, and therefore it can be compared with the argument of \add@missing.

\endproclaim ends the paragraph, switches back to \rm and adds spacing. (This means that if, for some strange reason, a whole section of text happens to be in italics, then the user must type \it again after each \endproclaim, but that seems too special a case to need providing for.) The penalty of 55 is just the plain.tex penalty for \endproclaim, carried over without change. Version 2.1 change: removed \outer prefix, to simplify some programming related to \add@missing and \revert@envir.

```
\def\endproclaim{%
         \revert@envir\endproclaim
 \par\rm\penaltyandskip@{55}%
                    \medskipamount}
```

1.7 References

The encoding is complicated, but the mark-up possibilities are rich. The macros are powerful and ingenious.

Purpose. The mark-up of bibliographic information in an SGML-like way.

Use. All the references have to be enclosed by `\Refs...\endRefs`. Within these each reference starts with `\ref` and ends with `\endref`.

Between the latter tags the (marked up) detail fields can be supplied in any order. The detail fields don't take an explicit closing tag! Some special tags are `\bysame`, when the reference has the same authors as the previous one, and the `\...info` fields.

The *end* punctuation, whether supplied or not, is taken care of by the macros, unless `\nofrills` is used to override the default formatting. The kind of label is implicitly specified via the use of the tags

- `\no`—the supplied number as label—respectively,
- `\key`—the supplied label, usually letters.

The formatting of the labels—enclosed by square brackets—is done by the macros. The width of the label can be adjusted via `\widestnumber`.

A typical example of mark up is

```
\Refs%\nofrills{Your heading}
\ref\no 1
\by D.E. Knuth
\book The \TeX book
\publisher Addison-Wesley
\yr 1984
\endref
...
```

A glimpse at the code. The encoding with comments included, is some 500 lines. Therefore, I'll only deal with the outer `\Refs`, `\ref`, `\by`, and the TEXnical `\makerefbox`.

```
\outer\def\Refs{\add@missing\endroster
 \add@missing\endproclaim
 \let\savedef@\Refs
 \let\Refs\relax %because of \outer-ness
 \def\Refs##1{\restoredef@\Refs
%For a monograph where the title of the
%References section is done using \title,
%we want to omit the normal "References"
%heading and the vertical skips above and
%below.  This can be accomplished using
%\Refs\nofrills{}.  As long as the vskip
%at the end of \endtopmatter is not less
%than \aboveheadskip and \belowheadskip,
%this will be accomplished by the fact
%that \penaltyandskip@ doesn't add to a
%previous larger vskip, and the ragged
%center part will simply vanish if #1 is
%empty.
 \if\notempty{##1}\penaltyandskip@{-200}%
  \aboveheadskip
  \begingroup \raggedcenter@\headfont@
     \ignorespaces##1\endgraf\endgroup
  \penaltyandskip@\@M\belowheadskip
 \fi
 \begingroup\def\envir@end{\endRefs}%
  \refsfont@\sfcode`\.\@m}%
%This line here is a little tricky. If a
%\nofrills is found when we look ahead,
%then \frills@ will become equal to \eat@
%and it will eat "References" before \Refs
%is allowed to read its argument.
%Also we have to use a \csname trick to
%get around the outerness of \Refs.  MJD
 \nofrillscheck{\csname Refs\expandafter
  \endcsname\frills@{{References}}}}
%
\def\endRefs{\par % This will check for a
 \endgroup}       % missing \endref, also
%
\def\ref{\par
 \begingroup \def\envir@end{\endref}%
%Start the reference.
 \noindent\hangindent\refindentwd
%Change \par so that it will supply a
%(presumably) missing \endref, with an
%error message.
 \def\par{\add@missing\endref}%
%\nofrills@list should always be assigned
%globally, to conserve save stack.
 \global\let\nofrills@list\empty@
%Change \linebreak and \mathbreak to work
%properly in the special ref environment.
 \refbreaks
 \procpaper@false \book@false \moreref@false
%Start an initial box, to match up properly
%with the first upcoming \makerefbox; this
%will be discarded.
 \def\curbox{\z@}\setbox\z@\vbox\bgroup
}
%
\def\endref{%
%To wind up the preceding box it is
%convenient to call \makerefbox again;
%it will also open a new box, however,
%so we give it arguments \thr@@ and
%\endgraf\egroup that will cause the new
%box to be closed immediately and discarded.
%But we must first make sure box 3 is void
%or we'll trigger an error message. This
%is done by dumping the current contents
%of (global register) box 3 into (local
%register) box 2; the \box command always
%makes its argument globally void.
%(Because box 0 is used heavily in
% \makerefbox, it's easier to just use
% box 2 here, rather than try to verify that
% using box 0 would be safe in all cases.)
 \setbox\tw@\box\thr@@
 \makerefbox?\thr@@{\endgraf\egroup}%
%Then we call \endref@ to take all the saved
%material and combine it into a paragraph,
%adding punctuation to separate pieces.
 \endref@
%The \endgraf is done here rather than in
%\endref@ because in \moreref or \transl
%cases \endref@ shouldn't do the \endgraf.
 \endgraf
%Finally, we need to close the group that
%was started by \ref. This has the effect
%of killing the current definition of
%\envir@end, among other things.
 \endgroup\keyhook@
 \global\let\keyhook@\empty@
```

```
%\global to conserve save stack
}
%
\def\by{\makerefbox\by\bybox@\empty@}
\newbox\bybox@
%
%\makerefbox takes three arguments:
%- the first is the name of the calling
%  macro, for use in error messages;
%- argument 2 is the box used for storing
%  data (note: some boxes are shared
%       by more than one calling macro);
%- argument 3 is additional material
%  (optional---may be empty) that may be
%  used to affect the contents of the box.
\def\makerefbox#1#2#3{\endgraf
%Set box0 to the just-completed line of
%text.
 \setbox\z@\lastbox
%Although \holdoverbox will usually be void
%it still doesn't hurt to \unhbox it here
%in every case, which simplifies the
%programming.
 \global\setbox\@ne\hbox{\unhbox\holdoverbox
%\ifvoid test is necessary here to prevent
%a \linebreak at the end of a field from
%being removed by the \unpenalty.
   \ifvoid\z@\else\unhbox\z@\unskip\unskip
                  \unpenalty\fi}%
 \egroup%ends the group from the previous
%If box 1 is empty (width <= 0pt) then set
%the current box to void (it might still
%have a baselineskip glue or something in
%it at this point, for one thing).
%Otherwise set it to box 1.
 \setbox\curbox\box\ifdim\wd\@ne>\z@ \@ne
                  \else\voidb@x\fi
%That finishes the previous box.
%Now let's start a new one using
%the box given as arg 1 of \makerefbox.
%But first check to see if it's void and if
%not give an error message.
 \ifvoid#2\else\Err@{Redundant \string#1;
  duplicate use, or mutually exclusive
  information already given}\fi
 \def\curbox{#2}\setbox\curbox\vbox\bgroup
  \hsize\maxdimen \noindent#3%
}
```

Explanation. `\ref` opens a (zero) box, which is immediately closed in the replacement text of any field tag. Each field is stored in a box. But not via simply enclosing the contents between braces or so. `\makerefbox` opens the box. The copy is processed further—thus processed in the box—and upon the invocation of the next tag, the box is closed and a box corresponding to this new tag is opened. And so on. In between many subtle details have been taken care of, which makes the code hard to understand, even the main lines.

> This boxing via the *chain* of `\makerefbox`-es is an ingenious application of the two-part macro TEXnique.

In `\endref` the macro `\endref@` is invoked, which winds up all the preceding boxes, and closes in a tricky way the last opened one, via `\endgraf\egroup` as third parameter of `\makerefbox`.

En-passant error detection is taken care of: a blank line or `\par` will entail a message: `\end...` tag is missing.

Alternative. There are alternatives available but they are generally overdone. I consider it simpler to work with a basic TEXnique, which I completely understand and which is sufficient and handy.

Personally, I experienced the inconvenience to select and update a set of references from my collection of literature references, every time I prepare a paper.

To overcome this extra and error-prone work, I decided to mold my file of (annotated) literature references into a sequence of TEX definitions.

> The idea is that while preparing a publication, I just have to supply the *list of names* of the definitions, appropriately separated, with the typesetting done automatically, eventually after having sorted the list.[19]

The automatic suppression of identical author names in subsequent references is nice and context sensitive. The substitution of the author name part by a rule is done by the AMS, but there it has to be marked up for, while it can be automated easily.[20] Also context sensitive are the suffixes in the year of publication, and these are handled in a similar way.[21]

There you go. The specification of my references section comes down to

```
\input references
\head*References*%TUGboat.sty's \head
\ls\knuthdea
\ls\knuthdeb
\ls\laancgvanderd
\ls\lamportla
```

My file of references contains among others

```
\def\knuthdea{Knuth, D.E (1973):
 The Art of Computer Programming.
 {\it Sorting and Searching.}
 Addison-Wesley.}
%
\def\knuthdeb{Knuth, D.E (1984):
 The \TeX book. Addison-Wesley.}
%
\def\laancgvanderd{Laan, C.G van der (1992):
```

[19] The approach is similar to my enhancements of Amy Hendrickson's sorting of address labels, as detailed in Sorting in BLUe.

[20] In keeping these kinds of things behind the scenes it does not not burden an author at all.

[21] It must be confessed, that I did not look ahead much, and therefore the first year is there and subsequent years replaced by a rule and a, b, etc. The latter is not common. Generally, the first year starts with the suffix a. My approach is simple to encode and also unambiguous. Of course, I can make it in agreement with current practice—the first year takes an a, by the same name-year references—by storing the reference, and to look ahead appropriately.

```
 Syntactic Sugar. MAPS92.2, 130--136.
 (Also GUST Bulletin 1993, and submitted
  TUG\,'93.)}
%
\def\lamportla{Lamport, L (1986):
 \LaTeX\ User's Guide \& Reference Manual.
 Addison-Wesley.}
```

To typeset this nicely, I use as list separator

```
\def\ls#1{\ea\bibitem#1}
%with
\newcount\bcnt \newcount\suffixcnt
\let\lstnme\relax \let\lstyear\relax
%
\def\bibitem#1(#2){\global\advance\bcnt1
 \def\authornme{#1}\def\authoryear{#2}%
 \ifx\lstnme\authornme
  \def\authornme{------}%
  \ifx\lstyear\authoryear
   \global\advance\suffixcnt1
   \def\authoryear{--}%
  \else\let\lstyear\authoryear\suffixcnt0
  \fi
 \else\let\lstnme\authornme
      \let\lstyear\authoryear\suffixcnt0
 \fi
 \item{[\the\bcnt]}\authornme\,(\authoryear
  \suffix)}%end \bibitem
%
\def\suffix{\ifcase\suffixcnt\or a\or b\or
 c\or d\or e\or f\or g\or h\or i\or j\or
 k\or l\or m\or n\or o\or p\or q\or r\or
 s\or t\or u\or v\or w\or x\or y\or z\fi}
```

The above yields

[1] Knuth, D.E (1973): The Art of Computer Programming. *Sorting and Searching.* Addison-Wesley.
[2] ——(1984): The TEXbook. Addison-Wesley.
[3] Laan, C.G van der (1992): Syntactic Sugar. MAPS92.2, 130–136. (Also GUST Bulletin 1993, and submitted TUG '93.)
[4] Lamport, L (1986): LATEX User's Guide & Reference Manual. Addison-Wesley.

On second thoughts. Should we embarrass authors who publish with AMS, with the details of the mark-up of references at all? Isn't it sufficient for an author just to supply entries to the database of literature references available at AMS? I mean it is already there and the only important thing for an author is

> To know and supply the relevant labels for the entries.

Then the problem of consistent formatting of the references has been transformed into the logistics of providing each author with access to the bibliographic database, especially for the labels to be provided. Ça va sans dire, that it should be possible for an author to copy the references as well, in order to get more complete proofs. This is similar to copying macros from the file servers, which is current practice nowadays.

The above can be applied even if authors like to format their own references. The publisher just replaces the list of marked up references by the pointers to the preformatted set of references, and select the appropriate ones. The best of both worlds!

1.8 Fonts

An a priori point to realize is that authors only need awareness of fonts to pick the special symbols from in their detailed math mark-up. In structure commands it should be hidden: the author just asks for `\title`, `\head`, and the like where the used font is left to the style. This assumption entails that math authors should be aware of what symbols are available and how to access them. The AMSfonts user's guide is very good at this point in providing tables with the symbols and their control sequences. For fonts which are not standard loaded under amsppt the accessing of symbols is more down to earth via the use of `\mathchar`.

What is available is treated in the section AMSfonts. Here we deal with how to access the special alphabets and symbols, that is the loading and the mark-up for special symbols in the copy.

Loading. The problem is that only 16 font families are allowed, so we have to be careful with what to load.[22] To a lesser extent the fonts take memory and slow down proofing. For the latter a `\syntaxonly` option is available which does not format, and can save some 30% in speed.

Fonts loaded with amsppt automatically. For general use: cmcsc8, cmex8, cmex7 (similar to plain but in additional sizes)
Math fonts: msam, msbm, (extra symbols),[23] and eufm (medium-weight Euler Fraktur).

Special load commands.[24]
`\loadbold`: cmmib (bold math italic), cmbsy (bold math symbols).
For this class access macros are supplied as detailed below. For the following classes the accessing of symbols has to be done via `\mathchar`. The classes can be derived from the font names by the suffix `fam`.
`\loadeufb`: eufb (bold Fraktur)
`\loadeusm`: eusm (medium-weight script)
`\loadeusb`: eusb (bold script)
`\loadeurm`: eurm (medium-weight 'cursive roman')
`\loadeurb`: eurb (bold 'cursive roman').

---

[22] Note that seven are already in use before AMS-TEX begins.
[23] AMS-TEX provides `\newsymbol` to attach a name to the 4-positional specified character for msam, msbm, not much different from the use of `\mathchardef`, except for the error detection.
[24] Control sequences to be supplied in the preamble area.

Mark up in math part of copy

Bold and other symbols. Basically, one has to apply the basic TEXnique[25] to access a math character from a font via the use of `\mathchar`, similar to the use of the `\char` primitive. However, the difference is that for math characters one has to specify also for the 'class' and the family. Because of that complication AMS-TEX provides the following for the bold cmmib, cmbsy, and eufb.

There are the control sequences, of which each takes one argument

- `\bold`, for bold letters (bold text, but in contrast with plain's `\bf` only for one symbol, the argument. Plain's `\bf` remains available)
- `\boldkey`, for symbols that actually appear on the keyboard (letters are in bold italic)
- `\boldsymbol`, for symbols specified by a single control sequence
- `\frak`, for a Fraktur letter
- `\Bbb`, for 'blackboard bold' (upper case only).[26]

An example mark-up reads
`$\bold x \boldsymbol\in \boldsymbol \varGamma$`.
For the specification of the control sequences see the AMS-TEX user's guide, or the file `amssym.def` and `amssym.tex`.

For accessing symbols from eufb, eusm, eurb, eurm, with classes `\eufbfam`, `\eusmfam`, `\eurbfam`, `\eurmfam`, we have to rely on `\mathchar` as shown in the following example for the letter 'a' (neglecting the catcode changes for @, in order to make the class a hexadecimal number)
`\mathchar"0\eurmfam@69`
with `\eurmfam@` the hexadecimal value of `\eurmfam`.

The user's guide is somewhat complicated, because it deals also with the situation when you are short of memory.

1.9 Math mark-up

I don't like the use of the heavy braces on among others the pages 103, 104, 106, 110, 111, 116, of Spivak, 1986.

Syntactic sugar. Much functionality of (plain) TEX has been renamed on the one hand, while on the other hand some functionality has been altered under the same name.

> Therefore, I consider the math mark-up syntactic sugar of plain's macros.[27]

This makes an AMS-TEX script incompatible with plain.[28]

- Instead of plain's infix command `\over`, there is the prefix command `\frac`[29]
- `\align`, `\aligned`, `\alignat`, `\split`, `\multiline`, and `\gather` with their `\end...` endings, add to plain's `\eqalign`, `eqalignno`, and `\displaylines`
- `\tag` supersedes `\eqno`
- Instead of `\matrix`, and `\pmatrix`, the substitutes `\bmatrix`, `\vmatrix`, `\pmatrix`, `\Vmatrix`, and `\smallmatrix`[30]
- At the definition front there is `\define`, `\redefine`, and `\predefine` as substitutes for `\def` and `\let`. (No replacement of the TEXnical `\futurelet`, of course.)
- `\edef` etc. disappeared as such. In `\accentedsymbol` it is used from the application viewpoint. This holds for some other TEX macros (or control sequences) too: `\overfullrule=0pt`, `\cr`, `\openup`, `\noalign`, `\phantom`, `\atop` and the like, `\vbox{\hsize=...}`, `\cal`, `\dots`, `\oldstyle`, `\hoffset`, `\voffset`, `\vadjust`, and the abbreviations period.

Next, some examples of the same names but with altered functionality

- `\item` has been redefined within the `\roster` environment[31]
- different use of `\footnote`
- different use of `\proclaim`,[32] and
- different attitude with respect to font changes.

---

[25] TEXbook chapter 17.
[26] Spivak, 1986, also provided some poor man's bolds via `\pmb`.
[27] I agree, however, that `\text{...}` is more natural than the use of `\hbox{...}`, and that the whole business of extra fonts is really something, not to forget the `\widehat`, `\widetilde`, the multiple integrals and the triple dots of the Schwarz derivative, to name but a few.
[28] Agreed, the reader is warned for those occurrences in Appendix C of the Joy of TEX.
[29] I know that the philosophy behind it is to have opening and closing braces not too far apart. But that could have been attained via defining substructures, which is a good habit anyway.
[30] The powerful and practical `\bordermatrix` disappeared from the stage, I mean is not mentioned at all.
[31] I love plain's `\item`. Happily, Appendix C reassures me that it can still be used as such.
[32] In the mean time `\proclaim` and its variants have moved into `amsppt.sty`.

$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX

If only the reader would now turn over to the excellent 'Guidelines for preparing electronic manuscripts—$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX,' much would have been gained.

2 The $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX package

Because $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX is biased towards LaTeX, Spivak's 'The Joy of TeX' is not quite appropriate as manual. The user's guide has been designed as a self-contained booklet. Alas, the booklet is not sufficient. The accompanying 'Guidelines for preparing electronic manuscripts—$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX,' is not only handy but necessary.

The $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX package consists of[33]
- `amstex.sty`, an extensive modification of `amstex.tex` that allows it to be used in LaTeX as documentstyle option
- `amsart.sty`, `amsbook.sty`, to parallel LaTeX's article and book styles
- some options: `amscd.sty` (commutative diagrams), extra math style options, for example left or right placement of equation numbers, verbatim, theorem, ...
- `amsfonts.sty`, `amssymb.sty`, and the definition files `fontdef.ori`, `fontdef.max`, and `fontdef.ams`

along with the documentation[34]
- $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX User's Guide
- $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX Installation Guide
- Guidelines for preparing electronic manuscripts—$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX

and some sample files
- `testbook.tex`
- `pref.tex`
- `chap1.tex`
- `chap2.tex`
- `app.tex`
- `testbook.bbl`.

In this part I'll treat
- `amsart.sty`
- some math mark-up details, and
- the use of fonts from the AMSfonts collection.
- syntax checking run mode can be selected via `\syntaxonly`.

2.1 Styles: amsart.sty

A document is thought to be marked up via the LaTeX structure[35]

`\documentstyle{amsart}`,
⟨*preamble*⟩,
`\begin{document}`
⟨*top matter*⟩
`\maketitle`
⟨*document proper*⟩
`\end{document}`.

Strong points
- Inheritance of some LaTeX goodies: picture environment, the automatic and symbolic cross-referencing, automatic numbering schemes
- A very-well developed title part
- Access to the AMSfonts collection via incorporation of NFSS[36]
- A fancy bibliographic tool with an option to include BiBTeX files
- `\syntaxonly` option.

Not in there
- to supply for pre- or suffixes, easily, in the automatically maintained number schemes
- the Babel option as such (There are no hard-wired names. The language determined keywords are parameterized)
- no general policy with respect to \⟨*tag*⟩... `\end`⟨*tag*⟩, as has been adopted in $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX
- no special provision for setting up marginal notes or two-column format
- 11pt and 12pt options have been reduced to a minimal kernel
- changed function of star-ed forms of `\chapter`, `\section`, and the like
- numbers and punctuation in italic text is set in upright font.

Encoding. With respect to encoding, I will discuss
- the systematic use of LaTeX's optional parameter mechanism,
- some structuring control sequences: `\title`, `\chapter` (c.q. `\section` and `\appendix`)
- math mark-up, and
- how to supply bibliographic items.

Encoding: Optional parameter mechanism. LaTeX's optional parameter mechanism consists of providing after the keyword the optional parameter, enclosed by the brackets, [ ]. The use of the (optional) parameters are detailed with in the LaTeX reference manual and user's guide, LRM for short.

Encoding: `\title`. For the `\title` the optional parameter has the function to supply a short title to be used in the running heads.

[33] The style files are part of the corresponding `.doc` files.
[34] In booklets and also available in electronic form.
[35] For a more complete, and detailed, structure see the template in Appendix B: amsl-art.tem.
[36] The New Font Selection Scheme, as detailed in the user's guide, and in the publications of Mittelbach and Schöpf.

Encoding: `\chapter`, and non-English keywords The keywords needed in `\abstract`, `\chapter`, and the like, make use of `\`⟨*tag*⟩`name`, which contain respectively `Abstract`, `Chapter`, and can be redefined.[37]

2.2 Math mark-up

When I started to use TEX,[38] I marked up my math, tables, and diagrams via LATEX. Since then I started to use more and more plain TEX, culminating in Math into BLUes, and Table Diversions. From working on those projects, I concluded—as detailed in my Syntactic Sugar—that the variant mark-up of math and tables—in contrast with the positioning of the element within the context—is just syntactic sugar of what plain already provides, except for a macro or two for special cases! A matter of taste.

The Math facilities provided by LATEX are augmented with some from AMS-TEX.

However, the lack of mentioning—in the LRM—the concept of formula classes is an oversight.

Encoding: Theorems and the like. In order to customize these structures, a `theorem.sty` option has been developed. It is advised to use this one instead of LATEX's `\newtheorem` environment. This option is based upon Mittelbach, 1989.

Three typographically different environments are considered: plain, definition, and remark. Via `\newtheorem` quite some environments have been created. I don't like that the descendants within a class share the same counter, although this can be undone.

2.3 References

There are two possibilities

- direct via the `\begin{thebibliography}{10}... \end{thebibliography}`, or
- after the use of the BiBTEX tool.

For more details, see the examples in the user's guide and the LRM.

An alternative to two-pass bibliography typesetting. It is curious that the placement of the bibliography items *in the manuscript* has to be at the end. This entails the forward referencing problem with its (costly) two-pass mechanism.[39] There is no good reason for not supplying the bibliography at the beginning of the compuscript, set it in 'boxes' (giving the formal labels their values), and typeset it at the end. The forward referencing problem is then no longer there. Active documents, aha!

A similar idea holds for the typesetting of table of contents. Why not typeset the toc and the end of the job with the page numbers set appropriately, I mean they usually take i, ii, ... After printing the toc pages can just be put in front. No two-pass job needed!

2.4 Fonts

Invariantly, we have to deal with the loading of the fonts to be used—special alphabets, like Fraktur, or extra math symbol tables—and the mark-up in the copy for the right character, or symbol.

Mittelbach and Schöpf state that typographical tradition has it, to characterize fonts by the attributes

shape, series, size and family.

This is another naming scheme than used by DEK. They developed this approach as the NFSS. An author using NFSS, has to be aware of

- the attributes and their accompanying control sequences: `\shape`, `\series`, `\size`, `\family`[40]
- the font change control sequence is `\selectfont` (apply this after having specified the attributes)
- the fontdef concept, and the realization in fontdef files (essentially these are the 'tables' to associate the combination of the attributes to the available fonts)
- the various shorthand notations, which do some of the actions of the above under the same name a LATEX author is used to.[41]

It must not be forgotten, that no more fonts are accessed as the AMS-TEX author can select already. Just the way of talking about it and the naming is different. The NFSS is essentially a mapping from the

imaginary 4-dimensional space onto the
1-dimensional space

of available fonts. I'm still pondering about the role the virtual font concept can play herein. I mean what is the difference between a variant of Euler Fraktur and the one provided by another firm, not belonging to the CM-family? Both are brands of the same essentially. To quote DEK

'The idea behind VF files is that a general interface mechanism is needed to

[37] Curious though, because I guess AMS is not publishing in languages different from English. On the one hand the styles are targeted at AMS publications and on the other hand flexibility is strived after. This is a little contradictory. For an AMS author there should be just the styles to format the publication, or even better the generic preprint styles. With as simple documentation as possible, not too much dealing with side-track issues, however important these issues are in general.

[38] De Bruin et al., 1988.

[39] Forward referencing means that the reference to the item precedes the occurrence of the item in the copy, and if this is done via symbolic names then this name is not yet defined.

[40] Especially the family concept is totally different from DEK's.

[41] But, sometimes it works differently, so the author has to delve into the matter. It is not completely upwards compatible and therefore error-prone.

> switch between the myriad font layouts provided by different suppliers of typesetting equipment. Without such an equipment people must go to great lengths writing unscrutinable macros whenever they want to use typesetting conventions based on one font layout in connection with actual fonts that have another layout.'

At the lower level the same names are still there. Abstraction has its features, though.

I would have welcomed the classical font tables of DEK with the fontdef-s in a direct way correlated to them. It is true that from the fontdef files, it is clear what you can ask for in your mark-up and what you get. For example

```
\family{cmr}\shape{it}\series{m}
\size{14}{18pt}\selectfont
```

yields

```
cmti12 at14.4pt
```

which can be looked up in the font table.

Loading. AMS-LaTeX provides the files `fontdef.ori`, `fontdef.max`, and `fontdef.ams`.[42] My wishful thinking is that the loading will be done via the inclusion of the fontdef file. Not so! Appendix A5 of the user's guide explains how to make a new format file via iniTeX. Not easy at all!

Mark-up of math copy. Basically one needs from the user's guide

- Table 3: Font commands used in text, and
- Table 4: Font commands used in math (these are the same as within AMS-TeX: `\bold`, `\boldsymbol`, `\pmb`, `\cal`, `\frak`, and `\Bbb`. `\mathrm` is usually hidden in `\text{...}`.).

> For the moment, I expect that an AMS-LaTeX author will be deprived from the wealth provided by AMSfonts.

I like the TeXnical way the attributes have been parameterized.

2.5 Some LaTeX drawbacks

All those (semi-)automatic features provided for example by LaTeX are dangerous, and paradoxically errorprone. The AMS-LaTeX user's guide for example, demonstrates how easy it is to forget for an unnumbered section in the table of contents.[43]

Another hindering thing is that counters with value 0 are ignored. If this is hidden in examples, without mentioning, I feel myself unhappy: LRM p.93, C.6 p.197.

AMSfonts

It is true that math mark-up requires some extra alphabets—for example Fraktur—and extra symbols, next to what is already provided by plain. The AMS did a great job in filling up this gap.

If only the reader would now turn over to the excellent user's guide AMSfonts, much would have been gained.

3 The AMSfonts package

The package consists of the

- various font files (see below)
- the user's and installation guide, also provided electronically.

3.1 Contents of the AMSFonts collection[44]

The AMSFonts collection contains the following fonts, in the sizes indicated:

- The Euler family, all but `EUEX` in 5, 6, 7, 8, 9, and 10 point:
  - Fraktur (German), medium-weight and bold (`EUFM` and `EUFB`)
  - "Roman" cursive, medium-weight and bold (`EURM` and `EURB`)
  - Script, medium-weight and bold (`EUSM` and `EUSB`)
  - Euler-compatible extension font (`EUEX`), in 7, 8, 9, and 10 point
- Additional sizes of some Computer Modern math fonts (the 10-point fonts are included in standard TeX distributions):
  - bold math italic (`CMMIB`), in 5, 6, 7, 8, and 9 point
  - bold math symbols (`CMBSY`), in 5, 6, 7, 8, and 9 point
  - math extension font (`CMEX`), in 7, 8, and 9 point (the 10-point font is included in standard TeX distributions)
- Extra math symbols, in 5, 6, 7, 8, 9, and 10 point:
  - first series, medium-weight (`MSAM`)
  - second series, including Blackboard Bold, medium-weight (`MSBM`)
- Cyrillic, developed at the University of Washington
  - lightface (`WNCYR`), in 5, 6, 7, 8, 9, and 10 point
  - bold (`WNCYB`), in 5, 6, 7, 8, 9, and 10 point
  - italic (`WNCYI`), in 5, 6, 7, 8, 9, and 10 point
  - caps and small caps (`WNCYSC`), in 10 point
  - sans serif (`WNCYSS`), in 8, 9, and 10 point

[42] It is advised to customize form these files your own fontdef file.

[43] The reference section is missing, while it is not at the end of the booklet, and therefore it should be listed in the table of contents.

[44] Borrowed from the AMSfonts user's guide.

- Computer Modern caps and small caps (CMCSC), in 8 and 9 point (the 10-point font is included in standard TeX distributions)
- The "dummy font," used in AMS-TeX for syntax checking, exists only as metrics (dummy.tfm)
- Other files needed to use these fonts:
  - amssym.tex, a file defining the symbols in fonts MSAM and MSBM
  - amssym.def, a file that loads the fonts MSAM, MSBM and EUFM and defines some control sequences required by amssym.tex
  - cyracc.def, a file containing definitions needed for proper access to characters in the cyrillic fonts
- Other useful files:
  - userdoc.tex, the source file for the AMS-fonts user's guide
  - userdoc.cyr, the source file for the table showing cyrillic input conventions, input by userdoc.tex
  - userdoc.fnt, the source file for the tables of the principal 10-point fonts in the AMSFonts collection, input by userdoc.tex; this file may also be TeXed by itself
  - userdoc.def, the macros used to format the AMSfonts user's guide
  - userdoc.ins, the source file for the appendixes to the AMSfonts user's guide, input by userdoc.tex; this file may also be TeXed by itself.

Each font at a particular size is provided in seven standard TeX magnifications, magsteps 0 through 5, including magstephalf. The AMSFonts package for the IBM PC and compatibles includes all magnifications. For use with *Textures* on the Macintosh, the Standard AMSfonts package includes only magsteps 0 and 1; the Extended AMSFonts package includes all seven magnifications. All instances of every font have been newly generated for this release of the AMSFonts collection.

Epilogue

The most difficult thing is to know when to stop, to keep the right balance, not to sub-optimize. The American Mathematical Society is doing a tremendous job, for sure. However, I believe that

> with a little less automation nearly the same results can be attained, . . . with less energy.[45]

Perhaps all the energy spent is the price the AMS is willing to pay for

> the benefit of their members, and the Math, TeX, and publishing scientific community at large,

in name of. . . progress.

TeXniques used

- Mark up of top matter in the SGML spirit
- Minimal mark-up style for references
- \csname... for using \outer control sequences non-outer
- Near generic encoding of \proclaim, \proof, \demo, \example, \remark, and the like
- General optional parameter TeXniques, to override the default formatting
- An environment stack to keep track of the environment the formatting process is in
- Font selection via the NFSS
- \add@missing, a general error detecting (and correcting) mechanism.

Conclusion

AMS-TeX together with the AMSfonts collection is a very good, and rich, extension of TeX for math authors, although I consider Spivak's math mark-up commands mainly as syntactic sugar of what has already been provided by plain.

Because AMS-TeX facilities have been dropped in favor of facilities provided by LaTeX, I find myself in good company with the conclusion that the math mark-up via AMS-TeX, or via LaTeX, are both syntactic sugar of what plain already provides.

It is confusing that the amsppt.sty and some of the other styles have been mixed in the booklet (and in the code).

AMS-LaTeX's user's guide is confusing in introducing a new font selection scheme—which is not more powerful—but makes perhaps the *talking* about it easier. It is symptomatic that the source file of the user's guide does not run with the NFSS.

> For the moment, I expect that an AMS-LaTeX author will be deprived from the wealth provided by AMSfonts.

General. As can be seen from the Appendices B and C the mark-up via amsppt.sty, looks quite different from the mark up via amsart.sty. Furthermore, the results in print differ, as can be seen from the samples provided in the Guidelines for preparing electronic manuscripts. This is due to the different tools: TeX vs. LaTeX, JoT vs. LaTeX math, \Refs vs. LaTeX's bibliography environment or BiBTeX, DEK's font selection and switching vs. NFSS.[46]

[45] An example of the famous 80%-20% results-investments rule, better known as the law the reduced gains.

[46] Not so much a problem for a specific TeX, respectively LaTeX, author, but it complicates and makes the maintenance costly. I myself use (LA)TeX whenever suitable, and then it is confusing.

> I would welcome `amsppt.sty`, and eventually a `amsart.sty`! as *the* generic author interfaces for a TEX, respectively LATEX-oriented author.

Also handy are templates to start from, next to the worked out samples, as supplied in the Guidelines . . . But should not they be part of the user's guides?

Acknowledgements

The American Mathematical Society, especially Regina Girouard, Ralph Youngen, and Kevin Curnow, are kindly acknowledged for providing the information, the booklets and the files.

References

[1] AMS (1991a): A look inside the AMS. (A nice brochure of what AMS is all about.)

[2] AMS (1991b): Think about publishing with the AMS. (Another nice brochure about the merits of publishing with AMS: effective marketing, extensive promotion, world-wide distribution, better sales, longer life of book, royalties, support worthwhile non-commercial activities for the benefit of the scientific community at large, for example the TEX project.)

[3] AMS (1993a): AMS-TEX User's Guide 2.1.

[4] AMS (1993b): AMS-TEX Installation Guide 2.1.

[5] AMS (1993c): AMS-LATEX User's Guide 1.1.

[6] AMS (1993d): AMS-LATEX Installation Guide 1.1.

[7] AMS (1993e): AMSfonts. User's Guide 2.1.

[8] AMS (1993f): AMSfonts. Installation Guide 2.1.

[9] AMS (1993g): Guidelines for preparing electronic manuscripts.
AMS-TEX (booklet, 52p), and the mirrored one AMS-LATEX (booklet, 58p).
(The first is very well-done. I have not seen a guideline of similar quality of yet! Simply the best available. Much experience is embodied to learn from. The second is verbose, incomplete, and deals at length with issues an author should not be bothered with.)

[10] Braams, J.L (1991): Babel, a multilingual style-option system for use with LaTeX's standard document styles. *TUGboat* 12, no. (2), 291–301.

[11] Bruin, R de, C.G van der Laan, J.R Luyten, H.F Vogt (1988): Publiceren met LATEX. CWI Syllabus 19. (Dutch. Useful additional remarks have been published in MAPS92.1.)

[12] Knuth, D.E (1984): The TEXbook. Addison-Wesley.

[13] Knuth, D.E (1990): Virtual Fonts: More Fun for Grand Wizards. *TUGboat* 11, no. (1), 13–23.

[14] Laan, C.G van der (1990): SGML(,TEX and . . .). Proceedings EuroTEX '90. *TUGboat* 12, no. (1), 90–104. (Has also appeared in GUTenberg Cahiers 5, and MAPS90.2.)

[15] Laan, C.G van der (1991): Math into BLUes.
Part I: Mourning. Proceedings TUG '91, *TUGboat* 12, no. (4), 485–501.
Part II: Sing your song. Proceedings EuroTEX '91, GUTenberg Cahiers, 10&11, 147–170. (An early version has appeared in MAPS91.1.)

[16] Laan, C.G van der (1992a): Syntactic Sugar. MAPS92.2, 130–136. (To appear GUST Bulletin. Submitted TUG '93.)

[17] Laan, C.G van der (1992b): FIFO & LIFO sing the BLUes. MAPS92.2, 139–144. (To appear *TUGboat* 14, no. (1).)

[18] Laan, C.G van der (1992c): Spivak's Œuvre. MAPS92.1. 139–142.

[19] Laan, C.G van der (1992d): Table Diversions. Proceedings EuroTEX '92, 191–211. (Also in MAPS92.2.)

[20] Laan, C.G van der (1993): Sorting in BLUe. MAPS93.1. 21p. (Submitted TUG '93.)

[21] Lamport, L (1986): LATEX, user's guide & reference manual. Addison-Wesley.

[22] Mittelbach, F (1989): An extension of the LATEX theorem environment. *TUGboat* 10, no. (3), 416–426.

[23] Mittelbach, F, R Schöpf (1989a): A new font selection scheme for TEX macro packages—the basic macros. *TUGboat* 10, no. (2), 222–238.

[24] Mittelbach, F, R Schöpf (1989b): The new font family selection—user interface to standard LATEX. *TUGboat* 11, no. (2), 297–305.

[25] Schöpf, R (1989): A new implementation of the LATEX `verbatim` and `verbatim*` environments. *TUGboat* 11, no. (2), 284–296.

[26] Spivak, M.D (1986): The Joy of TEX—AMS-TEX. AMS. ISBN 0-8218-2999-8. (Updated 2^{nd} printing 1990.)

[27] Spivak, M.D (1989): LAMS-TEX—The Synthesis. TEXplorators.

[28] Youngen, R (1989): Computers and Mathematics. Notices AMS. (Discusses TEX, LATEX and AMS-TEX, summarizing also the relative advantages. Since the publication of this note AMS-LATEX has been released, and Spivak has provided LAMS-TEX. Of course these are not dealt with.)

[29] Youngen, R (1991): Typesetting with TEX at the AMS. 4p. (A nice survey of why-and-what AMS is using TEX for.)

[30] Youngen, R (1992): TEX-based production at AMS. MAPS92.2, 63–68.

Appendix A: How to get it?

NTG is discussing with AMS the possibility of NTG being a redistributor of AMS TEX formatting tools ($\mathcal{A}\mathcal{M}\mathcal{S}$-TEX, $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX, styles, and corresponding documentation), off-the-shelf.

The files are in the public domain and can be obtained from the AMS—as detailed in their note below—and also from the file servers, for example Piet van Oostrum's server at RUU.

AMERICAN MATHEMATICAL SOCIETY
P.O. Box 6248, Providence, RI 02940[47]

This is a guide for accessing TEX macro packages and AMSFonts from the Society's public domain archive on the Internet node `e-MATH.ams.org`.

If you wish to obtain an entire package from the archive, such as the entire $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX or $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX package, you should create a directory structure on your local machine which parallels the directories from which you get the files on the host. In this way you will be able to transfer the entire contents of a directory to your local machine, and you will best be able to use the instructions for installing the package on your local machine.

The current it directory tree structure on the host is (Note added: not the files)

```
ams (50.6 MB total)
  amsfonts (21 MB total)
       doc (115K)
       pk-files (19 MB total)
          118dpi (2.1 MB)
          180dpi (2.9 MB)
          240dpi (3.4 MB)
          300dpi (4.5 MB)
          400dpi (6.1 MB)
       sources (2 MB total)
          cyrillic (359K)
          euler (1.3 MB)
          extracm (98K)
          symbols (282K)
  amslatex (1.5 MB total)
       doc (518K)
       fontsel (237K)
       inputs (330K)
       latex (347K)
  amstex (410K total)
       doc (245K)
       author-info (1.2 MB total)
          guidelines (261K)
          sty-files (961K)
       macintosh (3.8 MB total)
       tfm-files (150K)
```

There is a file TeXfiles on the /ams directory that contains a list of ALL the files on the directory, with size and last-change date that is suitable for downloading.

The host is a Unix environment. File and directory names are *case-sensitive*. (Note that directory names are lower case.) The basic FTP commands you will need to know are listed below.[48]

```
cd (directory name)
 connect to a directory on the remote host
cd ..
 connect to the parent of current directory
 on the host
lcd (directory name)
 locally connect to a directory on your
 computer
pwd
 ask for name of current directory on the
 host
lpwd
 ask for name of current directory on your
 computer
dir
 list contents of current directory on the
 host
ldir
 list contents of current directory on your
 computer
get (file name)
 get a single file from the host
mget (file specifications)
 get multiple files from the host
binary
 when you are about to transfer binary
 files, such as tfm files or pk files
ascii
 when you are about to transfer text files
 (most of the files on the host are text
  files)
exit
 leave FTP and return to your system
```

A1. How to log on

First, type

```
ftp e-math.ams.org    <return>
      or
ftp 130.44.1.100      <return>
```

When you see a message indicating that a connection has been opened, you need to log in using the username `anonymous`. If your version of FTP is now prompting you for a Name or Username, simply type `anonymous` (no quotes) and $\langle return\rangle$. Otherwise, type `login anonymous` and hit $\langle return\rangle$.

You will be asked to enter a password; enter your name here and press $\langle return\rangle$.

A2. Which directories do you need

(Users of TEXtures on a Macintosh will find everything they need in the directory /ams/macintosh. As the READ.ME file in that directory explains, files stored here are BinHex'ed StuffIt archives of entire distributions. These are all ascii files. If you are a TEXtures user, you may skip ahead to section 3. below.)

Whether you are going to use $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX, AMSFonts, $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX, or any combination of the three, you will need to have the TFM files for AMSFonts 2.1. To get them, retrieve all the files in the directory /ams/tfm-files using the instructions in Section 3. (These are binary files.)

If you are going to use $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX 2.1, you need to retrieve all of the files in the directories

[47] JULY 1990 [updated Dec. 22, 1992; NGB]

[48] Some implementations of FTP function slightly differently. For example, you may have to use 'quit' instead of 'exit'. If you have any problems, contact the support people at your site.

/ams/amstex
and
/ams/amstex/doc
(These are all ascii files.)

If you are going to use $\mathcal{A}_\mathcal{M}\mathcal{S}$-LaTeX 1.1, you need to retrieve all of the files in the directories

/ams/amslatex
/ams/amslatex/doc
/ams/amslatex/fontsel
/ams/amslatex/inputs

(These are all ascii files.)

If you are going to use AMSFonts 2.1, you need to retrieve all of the files in the directories

/ams/amsfonts
/ams/amsfonts/doc

(All of these are ascii files.) In addition:

- If you are using a 300dpi printer, you need the files from the directory /ams/amsfonts/pk-files/300dpi. (These are binary files.)
- If you are using a screen previewer which uses 118dpi resolution, you need the files from the directory /ams/amsfonts/pk-files/118dpi. (These are binary files.)
- If you have an implementation of Metafont and are able to create your own raster images of fonts from Metafont source files, you may wish to retrieve the files from the directory /ams/amsfonts/sources and some or all of its subdirectories. (These are ascii files.)

If you want to retrieve the Guidelines for Preparing Electronic Manuscripts, you need to retrieve the files in the directories[49]

/ams/author-info
/ams/author-info/guidelines .

(All of these are ascii files.)

A3. How to get the files from a directory

In the instructions below, when you are directed to retrieve the files from a directory or subdirectory on the host, perform each of the following steps

1. Connect to the directory on the host, using the `cd` command (You can do it in one command, e.g.: `cd /ams/amsfonts/pk-files/300dpi`; or a series of commands, e.g.: if you are already connected to the directory /ams/amsfonts, enter `cd pk-files` followed by `cd 300dpi`)
2. Enter `pwd` to make sure that step (1) was successful.
3. Enter `dir` to see how many files are in the directory. (optional)
4. Connect to the directory on your computer where the files will go, using the `lcd` command.[50]
5. Enter `lpwd` to make sure that step (3) was successful.
6. Enter either `ascii` (for text files) or `binary` (for tfm or pk files).
7. Enter `mget *.*`
8. Enter `ldir` to be sure that all the files were copied. (optional)

A4. How to end the ftp session and log off

To exit FTP simply type `exit` and $\langle return \rangle$.

A5. What to do on your own system

BEFORE attempting to install any of these macro or font packages, READ the appropriate READ.ME files which were copied to your system when you typed `mget *.*`.

Any questions concerning the TeX-ware products can be sent to the Internet address
Tech-Support@math.ams.org.
Reports of problems in accessing the FTP node itself should be sent to the Internet address
support@e-math.ams.org

Appendix B: amst-art.tem

For the complete sample, see the Guidelines for preparing electronic manuscripts—$\mathcal{A}_\mathcal{M}\mathcal{S}$-TeX.

```
%Stripped and modified version of:
% AMS-TeX 2.0+ file for a sample article
% for electronic submission.
%by C.G. van der Laan,
%in order to stress the main structure,
%and to provide more or less a template.
%
\input amstex
\documentstyle{amsppt}
%...
\topmatter
\title Sample \AmSTeX{}et cetera\endtitle
\rightheadtext{SAMPLE \AmSTeX{} ...}
\author BLUe\endauthor
\address Department of ...\endaddress
\email BLUe\@rug.nl\endemail
\keywords AMSppt.sty,...\endkeywords
\subjclass Primary 54C40, 14E20;
 Secondary 46E25, 20C20\endsubjclass
\abstract This paper is ...\endabstract
\thanks The author ...\endgraf\endthanks
\endtopmatter
%
\document
%
\head 1. Introduction%bold, centered;
\endhead%don't type final punctuation
This sample paper illustrates ...

\subhead Top matter\endsubhead
The input format ...

\head 2. Theorems, ...\endhead
%
Theorems and lemmas are varieties of
```

[49] Note added: To get the sample article from the guidelines directory, say `cd /ams/author-info/guidelines` followed by `get amst-art.tex`, respectively `get amsl-art.tex`.

[50] The format in which you type the directory name should be the normal format in which you type it on your operating system.

```
\proclaim{Lemma 1} Let $f, g\in  A(X)$...
\roster
\item"(a)" If $f$ is $E$-regular...
\item"(b)" If $f$ is $E$-regular...
\item"(c)" If $f(x)\ge c>0$ for all...
\endroster
\endproclaim

\demo{Proof}
\roster\runinitem "(a)" Obvious.
\item"(b)" Let $h, k\in A(X)$ ...
\endroster
\enddemo

\definition{Definition}
For $f\in A(X)$, we define
$$...$$
\enddefinition

\head 5. Figures\endhead
Figures are handled as inserts,...

\example{Example 5}
For the link in Figure 5a,...
\endexample
% art work measures 11.5pc for figure 5a
\topinsert\vskip 11.5pc
\botcaption{Figure 5{\rm a}}\endcaption
\endinsert

\Refs%\nofrills{Your heading}
\ref\no 1
\by     V. L. Arnol\cprime{}d,...
\book   Singularities of ... . \rom{I}
\publ   ``Nauka''
\publaddr Moscow \yr1982 \lang Russian
\transl English transl.
\publ Birkh\"auser\publaddr Basel\yr1985
\endref
\ref\no 2
\bysame
...
\endref
\endRefs
\enddocument
```

Appendix C: amsl-art.tem

For the complete sample, see the Guidelines for preparing electronic manuscripts—$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX.

```
%Stripped and modified version of
% AMS-LaTeX 1.1 file for a sample
% article for electronic submission.
%by C.G. van der Laan,
%in order to stress the main structure,
%and to provide a template.
\documentstyle{amsart}
\makeatletter
\def\LaTeX{\leavevmode L\raise.42ex%
 \hbox{\kern-.3em\size{\sf@size}{0pt}%
 \selectfont A}\kern-.15em\TeX}
\makeatother
\newcommand{\AMSLaTeX}{\protect
 \AmS-\protect\LaTeX}
\newcommand{\BibTeX}{{\rm B\kern-.05em{\sc
 i\kern-.025emb}\kern-.08em\TeX}}
\newtheorem{thm}{Theorem}[subsection]
\newtheorem{lem}[thm]{Lemma}

\theoremstyle{definition}
\newtheorem{defn}{Definition}[subsection]
\newtheorem{exmp}{Example}[subsection]
\newcommand{\subfig}{a}
\makeatletter
\def\alphenumi{%
  \def\theenumi{\alph{enumi}}%
  \def\p@enumi{\theenumi}%
  \def\labelenumi{(\@alph\c@enumi)}}
\makeatother

\begin{document}

\title[Sample \AMSLaTeX...]%
 {Sample \AMSLaTeX{}...}
\author{BLUe}
\address{Department of...}
\email{BLUe@@rug.nl}
\thanks{The author ...}

\keywords{amsart.sty, amstex.sty,...}
\subjclass{54C40, 14E20;
 Secondary 46E25, 20C20}

\maketitle

\begin{abstract}This paper...
\end{abstract}

\section{Introduction}
This sample paper...

\subsection{Top matter}
The input format ...

\section{Theorems,...}
Theorems and lemmas and similar...

\begin{lem}
Let $f, g\in  A(X)$...
\begin{description}
\item[(a)] If $f$ is $E$-regular...
\item[(b)] If $f$ is $E$-regular...
\end{description}
\end{lem}

\begin{pf}
\begin{enumerate}
\alphenumi % Defined in the preamble.
\item Obvious.
\item Let $h, k\in A(X)$...
\qed
\end{enumerate}
\renewcommand{\qed}{}
\end{pf}

\begin{defn}
For $f\in A(X)$,...
\begin{equation}...\end{equation}
\end{defn}

\section{Figures}
Figures are handled as inserts,...
\begin{exmp}...\end{exmp}
% art work measures 14.5pc for figure a
\begin{figure}[t]
  \makeatletter\makeatother
  \renewcommand{\thefigure}%Look at this!
    {\arabic{figure}\rom{\subfig}}
  \vspace{14.5pc}
  \caption{}  % figure a
  \label{f:afig}
\end{figure}
% When BibTeX is used... (see guidelines)
\makeatletter \renewcommand{\@biblabel}%
 [1]{\hfill#1.}\makeatother
\begin{thebibliography}{10}
```

```
\bibitem{AVG1}
 V.~L. Arnol$'$d,...
 {\em Singularities of...\rom{I}},
 ``Nauka'', Moscow, 1982 (Russian), English
   transl., Birkh\"auser, Basel, 1985.
\bibitem{AVG2}
 \bysame,
 {\em Singularities of...},
...
\end{thebibliography}
\end{document}
```

Appendix D: Contents