

# The `amsart`, `amsproc`, and `amsbook` document classes

American Mathematical Society  
Michael Downes  
updated by Barbara Beeton

Version 2.20.4, 2017/10/31

## 1 Introduction

This file (`amsclass.dtx`) is the master file for three L<sup>A</sup>T<sub>E</sub>X document classes, `amsart`, `amsproc`, and `amsbook`, which are intended for articles and books containing mathematical research. They produce output that follows the style conventions of American Mathematical Society publications. The theorem setup features of these document classes are also available in a separate package, `amsthm`.

## 2 Implementation

Three document class files and one package file (`amsthm.sty`) are produced from this source. Most of the code of the `amsthm` package is used in all four derived files. Most of the remaining code is used in all three document class files. Fine tuning is done with additional docstrip guards.

The usual name, date, and version information. (Note: the reason each `\ProvidesClass` command is placed on a line by itself, with separate begin and end guards for docstripping, is to make automatic update of file date and version slightly easier and more robust.)

```
1 \NeedsTeXFormat{LaTeX2e}% LaTeX 2.09 can't be used (nor non-LaTeX)
2 [1995/06/01]% LaTeX date must be June 1995 or later
3 <*amsart>
4 \ProvidesClass{amsart}[2017/10/31 v2.20.4]
5 </amsart>
6 <*amsproc>
7 \ProvidesClass{amsproc}[2017/10/31 v2.20.4]
8 </amsproc>
9 <*amsbook>
10 \ProvidesClass{amsbook}[2017/10/31 v2.20.4]
11 </amsbook>
```

For `amsthm` we need to guard against redundant loading via

```
\documentclass{amsart}
\usepackage{amsthm}
```

because in that case the usual `\RequirePackage` mechanism for avoiding redundant loading will not apply. We need to simulate the loading of the `amsthm` package.

```
12 <*classes>
13 \global\expandafter\let\csname ver@amsthm.sty\expandafter\endcsname
14   \csname ver@\currname.\@current\endcsname
15 </classes>
16 <*amsthm>
17 \ProvidesPackage{amsthm}[2017/10/31 v2.20.4]
18 </amsthm>
```

The following code is shared by the classes and the `amsthm` package. Cf. `amsgen.sty`.

```
19 \let\@xp=\expandafter
20 \let\@nx=\noexpand
21 \def\@oparg#1[#2]{\@ifnextchar[#{#1}{#1[#2]}}
22 \long\def\@ifempty#1{\@xifempty#1@.\@nil}
23 \long\def\@xifempty#1#2@#3#4#5\@nil{%
24   \ifx#3#4\@xp\@firstoftwo\else\@xp\@secondoftwo\fi}
25 \long\def\@ifnotempty#1{\@ifempty{#1}{}}
26 \def\setboxz@h{\setbox\z@\hbox}
27 \def\@addpunct#1{%
28   \relax\ifhmode
29     \ifnum\spacefactor>\@m \else#1\fi
30   \fi}
```

`\nopunct` should have a value for `\spacefactor` that is not used for `\frenchspacing`.

```
31 \def\nopunct{\spacefactor 1007 }
32 \def\frenchspacing{\sfcode'\.1006\sfcode'\?1005\sfcode'\!1004%
33   \sfcode'\:1003\sfcode'\;1002\sfcode'\,1001 }
```

If this class is loaded by a parent document class, then we want to use the name of the parent class. Otherwise the name of the current class file.

```
34 <*classes>
35 \def\@tempa#1#2\@nil{\edef\@classname{#1}}
36 \expandafter\@tempa\currnamestack{}{}{}@nil
37 \ifx\@classname\empty \edef\@classname{\currname}\fi
```

## 2.1 Support for conditional text

[This needs to be documented in the users' guide, including the idea of using `\for{5ed}{\linebreak}` to mark edition-specific line and page breaks. [mjd,1999/12/27]]

We would sometimes like to be able to mark fragments of text to be conditionally discarded or typeset. For example in the title of a section if we want to add a linebreak but prevent this linebreak from also taking effect in the table of contents.

Certain kinds of switches need to be built into the low-level structure of our document class in order to be useful. For example, inside the toc we need to arrange for an “in-toc?” test to yield true.

```

38 \def \@True{00}
39 \def \@False{01}
40 \newcommand\newswitch[2][False]{%
41   \expandafter@ifdefinable\csname ?@#2\endcsname{%
42     \global\expandafter\let\csname ?@#2\expandafter\endcsname
43     \csname @#1\endcsname
44   }%
45 }
46 \newcommand{\setFalse}[1]{%
47   \expandafter\let\csname ?@#1\endcsname \@False
48 }
49 \newcommand{\setTrue}[1]{%
50   \expandafter\let\csname ?@#1\endcsname \@True
51 }

```

The empty switch is by default false; i.e., if you write

```
\for{}{...}
```

the material will be discarded.

```
52 \newswitch{}
```

To get a line break in a section title but not in the table of contents line for that section, use `\except{toc}{\linebreak}`. (Presumably you are already giving a shortened running head version separately, if applicable.)

```

53 \DeclareRobustCommand{\except}[1]{%
54   \if\csname ?@#1\endcsname \expandafter@gobble
55   \else \expandafter@firstofone
56   \fi
57 }
58 \DeclareRobustCommand{\for}[1]{%
59   \if\csname ?@#1\endcsname \expandafter@firstofone
60   \else \expandafter@gobble
61   \fi
62 }

```

The `\forany` command needs to run through a comma-separated list of switch names and print its second argument if any of the switches are true.

```

63 \DeclareRobustCommand{\forany}[1]{%
64   \csname for@any@01\endcsname#1,?,\@nil
65 }
66 \@namedef{for@any@\@False}#1,{%
67   \csname for@any@%
68     \csname ?@\zap@space#1 \@empty\endcsname
69   \endcsname
70 }
71 \@namedef{?@?}{x}
72 \@namedef{for@any@\@True}#1\@nil#2{#2}
73 \def\forany@x{\@car@gobble}

```

## 2.2 Options

### Notes

Options will be processed in the order they are declared; cf. `\ProcessOptions`.

### Paper size

The option `letterpaper` (default) sets the target paper width and height to U.S. letter size, 8.5 in x 11 in. An option `a4paper` is also supported, but we don't include some of the more unusual paper options (`legalpaper`, `a5paper`, `executivepaper`) of the generic `article` class. For A4 paper we not only change the paper size but also add 4pc to the normal `textheight` of 50.5pc (the difference between 297mm and 11in is 50pt).

```
74 \DeclareOption{a4paper}{\paperheight 297mm\paperwidth 210mm
75   \textheight 54.5pc }
76 \DeclareOption{letterpaper}{\paperheight 11in\paperwidth 8.5in }
```

The options `landscape` and `portrait` swap paper height and width.

```
77 \DeclareOption{landscape}{\@tempdima\paperheight
78   \paperheight\paperwidth \paperwidth\@tempdima}
79 \DeclareOption{portrait}{}
```

### Two-sided or one-sided printing

For two-sided printing we set the switch `\if@twoside` which will cause the margins to be adjusted so that the type blocks of back-to-back pages will line up. The `\if@mparswitch` makes margin paragraphs print in the outside margin.

```
80 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
81 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
```

### Draft or final version

The `draft` option causes overfull lines to be marked with a black slug in the right margin.

```
82 \DeclareOption{draft}{\overfullrule5\p@
83   \ClassWarningNoLine{\@classname}{%
84     When the draft option is used, the
85       \protect\includegraphics\MessageBreak
86       command will print blank placeholder boxes\MessageBreak
87       for the graphics}%
88 }
89 \DeclareOption{final}{\overfullrule\z@ }
```

### Posting date

The date when an article is officially posted to the WWW is recorded in a variable `\@dateposted` with the `\dateposted` command.

```
90 \def\dateposted#1{\def\@dateposted{#1}}%
91 \let\@dateposted\@empty

92 <amsart>
93 \def\@setdateposted{%
94   \newline Article electronically published on \@dateposted}
95 </amsart>
```

**Logos**

The following logo is used for regular journal articles. The one for proceedings articles and the one for e-only journals are given separately.

```

96 <*amsart>
97 \def\article@logo{%
98   \set@logo{%
99     \publname

Current volume might be empty when an article is first posted to the WWW.
In that case leave out the issue-specific info.

100   \ifx\@empty\currentvolume
101   \else \newline\volinfo, \pageinfo
102   \fi
103   \newline \@PII
104   \ifx\@empty\@dateposted \else \@setdateposted\fi
105   }%
106 }
107 \def\eonly@logo{%
108   \set@logo{%
109     \publname
110     \newline\volinfo, \pageinfo
111     \ifx\@empty\@dateposted \else \@setdateposted\fi
112     \newline \@PII
113   }%
114 }
115 </amsart>

116 <*amsart | amsproc>
117 \def\@logo@font{\fontsize{6}{7p@}\selectfont}
118 \long\def\set@logo#1{%
119   \vbox to\headheight{%
120     \@parboxrestore \@logo@font
121     \noindent#1\par\vss
122   }%
123 }
124 </amsart | amsproc>

125 <*amsproc>
126 \def\procart@logo{%
127   \set@logo{\publname
128     \ifx\@empty\volinfo \else\newline\volinfo\fi}%
129 }
130 </amsproc>

```

**E-only journal**

Electronic-only journals (for `amsart` only) have different information in the series logo than paper-only or dual journals. Only the volume number is reported (no issue or year), and the posting date is added following the page numbers. [bnb, 1996/10/31]

This option will be invoked only from publication-specific `.cls` files.

```

131 <*amsart>
132 \DeclareOption{e-only}{%
133   \def\volinfo{Volume \currentvolume}%
134   \dateposted{Xxxx XX, XXXX}%
135   \def\@setdateposted{\ (\@dateposted)}%
136   \let\article@logo\only@logo
137 }
138 </amsart>

```

### Title page

The title and related information can optionally be printed on a separate page.

```

139 \newif\if@titlepage
140 \DeclareOption{titlepage}{\@titlepagetrue}
141 \DeclareOption{notitlepage}{\@titlepagefalse}

```

### Start on right- or left-hand page

For some book series, it's permissible to start chapters on a left-hand page. Default to 'openright', the usual AMS book style.

```

142 <*amsbook>
143 \newif\if@openright
144 \DeclareOption{openright}{\@openrighttrue}
145 \DeclareOption{openany}{\@openrightfalse}
146 \@openrighttrue
147 </amsbook>

```

### Two-column printing

Two-column layout is handled through a predefined internal switch.

```

148 \DeclareOption{onecolumn}{\@twocolumnfalse}
149 \DeclareOption{twocolumn}{\@twocolumntrue}

```

### The nomath option

The `nomath` option causes most of the extra math features to be omitted. Some utility functions will be defined below if this option is specified.

```

150 \DeclareOption{nomath}{-}

```

### Some font options

The `noamsfonts` option means to avoid declaring math alphabets or symbol fonts for the extra math fonts in the AMSFonts set. If these fonts are declared, it means that the corresponding `.tfm` files are required even for documents that do not use any symbols from those fonts. So we allow optionally to not declare them, for convenience of users who don't have those fonts on their system and don't want the hassle of getting them.

```

151 \DeclareOption{noamsfonts}{-}

```

The `psamsfonts` option, passed on to the `amsfonts` package, means that alternative `.fd` files should be used that do not refer to `.tfm` files for sizes 6,8,9 (which are not present in the PostScript (Type 1) AMS fonts set from Y&Y/Blue Sky Research). This should also trigger the `cmex10` option of `amsmath`, to avoid trying to load sizes 7–9 of `cmex`.

```

152 \DeclareOption{psamsfonts}{%
153   \PassOptionsToPackage{psamsfonts}{amsfonts}%
154   \PassOptionsToPackage{cmex10}{amsmath}}

```

### Equation numbering on the left or right

The option `leqno`—equation numbers on the left—is the default in AMS styles. Therefore we provide also a `reqno` option.

```

155 \newif\iftagsleft@
156 \DeclareOption{leqno}{%
157   \tagsleft@true \PassOptionsToPackage{leqno}{amsmath}}
158 \DeclareOption{reqno}{%
159   \tagsleft@false \PassOptionsToPackage{reqno}{amsmath}}

```

### Vertical centering of equation numbers

For multiline equations the equation number is by default centered vertically on the total height of the equation. To make the equation number print on the first line (for left-hand numbers) or the last line (right-hand numbers), there is a `tbtags` option ‘top/bottom tags’.

```

160 \newif\ifctagsplit@
161 \DeclareOption{centertags}{%
162   \ctagsplit@true \PassOptionsToPackage{centertags}{amsmath}}
163 \DeclareOption{tbtags}{%
164   \ctagsplit@false \PassOptionsToPackage{tbtags}{amsmath}}

```

### Flush left displays

The option `fleqn` causes displayed equations to print aligned on the left instead of centered, with an indentation of `\mathindent` from the prevailing left margin. If the `amsmath` package is loaded, most of this code will be overridden, but it seems we need it anyway because of the possibility of the `nomath` class option.

```

165 \DeclareOption{fleqn}{%

```

### Dealing with font sizes

`\@mainsize` Instead of the miserly `\@ptsize` variable from L<sup>A</sup>T<sub>E</sub>X’s ancient history that contains only the last digit of the main typesize, we set up a proper variable `\@mainsize` that contains all the digits of the main typesize. Just in case it is needed for someone using an old package, we will keep `\@ptsize` also.

```

166 \newcommand{\@mainsize}{10}
167 \newcommand{\@ptsize}{0}

```

`\larger` This function is an easy byproduct of the work done to fold typesize-specific  
`\smaller` code into the main class file. The range of font sizes is `\normalsize`, `\small`, `\Small`, `\SMALL`, `\tiny`, `\Tiny`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`. Spaces are left at either end of the case statement to accommodate adding `\TINY` and `\HUGE` in the future but it’s not clear that they’re really needed.

```

168 \newcommand{\larger}[1][1]{%
169   \count@\@currsizeindex \advance\count@#1\relax
170   \ifnum\count@<\z@ \count@\z@ \else\ifnum\count@>12 \count@12 \fi\fi

```

The various size-changing commands `\normalsize`, etc., will take care of updating `\currsizeindex`.

```

171 \ifcase\count@
172 \Tiny\or\Tiny\or\tiny\or\SMALL\or\Small\or\small
173 \or\normalsize
174 \or\large\or\Large\or\LARGE\or\huge\or\Huge\else\Huge
175 \fi
176 }
177 \newcommand{\smaller}[1][1]{\larger[-#1]}

```

The `\@adjustvertspacing` function adapts some vertical spacing amounts to the current type size. We don't expect large sections of vertical text to occur in the extraordinarily small or large type sizes, so the `\@adjustvertspacing` function is only called in the range between 'footnote' size and 'Large' size. Notice that no shrinkability is used.

```

178 \def\@adjustvertspacing{%

```

`\big/med/smallskipamount` are generic space values that will be used by the commands `\bigskip`, `\medskip`, `\smallskip`. We also link the spacing around displayed equations to these amounts.

```

179 \bigskipamount.7\baselineskip plus.7\baselineskip
180 \medskipamount\bigskipamount \divide\medskipamount\tw@
181 \smallskipamount\medskipamount \divide\smallskipamount\tw@
182 \abovedisplayskip\medskipamount
183 \belowdisplayskip \abovedisplayskip

```

The above-display short space is zero but with the same stretchability as the above-display normal space. And the below-display short space is similar, but has a base value equal to `\smallskipamount`. Use of the multiplier 1 is an arcane T<sub>E</sub>X trick that coerces the skip value to a dimen value, i.e., gives us the base value of the skip register without the stretch or shrink values.

```

184 \abovedisplayshortskip\abovedisplayskip
185 \advance\abovedisplayshortskip-1\abovedisplayskip
186 \belowdisplayshortskip\abovedisplayshortskip
187 \advance\belowdisplayshortskip 1\smallskipamount

```

The traditional value for `\jot` is 3pt, which we generalize to `\baselineskip/4`. This is used to adjust interline spacing in multiline displayed equations.

```

188 \jot\baselineskip \divide\jot 4 \relax
189 }

```

We fill out the range of typesize changing commands to a full eleven: five large/huge commands and five small/tiny commands. (The capitalization of the command names suggests that there should actually be thirteen—add `\TINY` and `\HUGE`—but let's be conservative and leave those out until a real need for them is known to exist.) An unavoidable side effect is that `\tiny` now selects 6pt instead of 5pt by default.

In version 1.1 of `amsart` and `amsbook` `\small` was the same as `\footnotesize` (`amsproc` didn't exist in v. 1.1).

The only size-changing command that is predefined by L<sup>A</sup>T<sub>E</sub>X is `\normalsize`; that's why it's the only one for which we use `\renewcommand` below.

```

190 \renewcommand\normalsize{\@xsetfontsize\normalsize 6%
191   \@adjustvertspacing \let\@listi\@listI}
192 \DeclareRobustCommand{\Tiny}{\@xsetfontsize\Tiny 1}
193 \DeclareRobustCommand{\tiny}{\@xsetfontsize\tiny 2}
194 \DeclareRobustCommand{\SMALL}{\@xsetfontsize\SMALL 3}
195 \DeclareRobustCommand{\Small}{\@xsetfontsize\Small 4%
196   \@adjustvertspacing
197   \def\@listi{\topsep\smallskipamount \parsep\z@skip \itemsep\z@skip
198     \leftmargin=\leftmarginI
199     \labelwidth=\leftmarginI \advance\labelwidth-\labelsep
200   }%
201 }
202 \DeclareRobustCommand{\small}{\@xsetfontsize\small 5\@adjustvertspacing}

```

For backward compatibility we had better define `\footnotesize` and `\scriptsize`. Also there is the small discrepancy with `\tiny` to worry about.

```

203 \def\footnotesize{\Small}
204 \def\scriptsize{\SMALL}

```

The sizes above 10pt use magstep values, stored in the functions `\@xipt`, `\@xiipt`, etc.

```

205 \DeclareRobustCommand{\large}{\@xsetfontsize\large 7\@adjustvertspacing}
206 \DeclareRobustCommand{\Large}{\@xsetfontsize\Large 8\@adjustvertspacing}
207 \DeclareRobustCommand{\LARGE}{\@xsetfontsize\LARGE 9}
208 \DeclareRobustCommand{\huge}{\@xsetfontsize\huge{10}}
209 \DeclareRobustCommand{\Huge}{\@xsetfontsize\Huge{11}}
210 %\DeclareRobustCommand\HUGE{\@xsetfontsize\HUGE{12}}

```

So now we had better define the `\@xsetfontsize` function. The size-changing commands use `\@setfontsize` instead of `\fontsize` to (a) give an error message if used in math mode and (b) set the `\@currsize` variable.

```

211 \def\@xsetfontsize#1#2{%
212   \chardef\@currsizeindex#2\relax
213   \edef\@tempa{\@nx\@setfontsize\@nx#1%
214     \xp@ifcase\@xp\@currsizeindex\@typesizes

```

Add nonsense values 99/99 at the end just in case some extreme error turns up.

```

215     \else{99}{99}\fi}%
216   \@tempa
217 }

```

For the record let's initialize `\@currsizeindex`.

```

218 \chardef\@currsizeindex=6

```

Set page-breaking penalties to prevent all widows, orphans, and hyphens at the end of a page.

```

219 \widowpenalty=10000
220 \clubpenalty=10000
221 \brokenpenalty=10000

```

Set some default linespacing values. The variable `\linespacing` is usually the normal interline space in the main text. It is used to specify vertical space for elements such as section heads and theorems in proportion to the normal interline space.

```
222 \newdimen\linespacing
223 \lineskip=1pt \lineskiplimit=1pt
224 \normallineskip=1pt \normallineskiplimit=1pt
225 \let\baselinestretch=\@empty
```

Settings for `\textheight` and `\textwidth`. We start with the value 50.5pc specified in AMS journal specifications as the total height of the type block and then subtract the running head height and adjust for `\topskip` to get the proper value for the text block.

```
226 \headheight=8pt \headsep=14pt
227 \amsbook\footskip=18pt
228 \amsart\amsproc\footskip=12pt
229 \textheight=50.5pc \topskip=10pt
230 \textwidth=30pc
231 \columnsep=10pt \columnseprule=0pt
```

Some settings for marginpars.

```
232 \marginparwidth=90pt
233 \marginparsep=11pt
234 \marginparpush=5pt
```

To avoid setting text before begin-document, we postpone the setting of `\footnotesep` using `\AtBeginDocument`.

```
235 \AtBeginDocument{\settoheight{\footnotesep}{\footnotesize M$^1$}}
236 \skip\footins=7pt plus11pt
237 \skip\@mpfootins=\skip\footins
238 \fboxsep=3pt \fboxrule=.4pt
239 \arrayrulewidth=.4pt \doublerulesep=2pt
240 \labelsep=5pt \arraycolsep=\labelsep
241 \tabcolsep=\labelsep \tabbingsep=\labelsep
242 \floatsep=15pt plus 12pt \dblfloatsep=15pt plus 12pt
243 \textfloatsep=\floatsep \dbltextfloatsep=15pt plus 12pt
244 \intextsep=\floatsep
245 \@fptop=0pt plus1fil \@dblfpptop=0pt plus1fil
246 \@fpbot=0pt plus1fil \@dblfpbot=0pt plus1fil
247 \@fpsep=8pt plus2fil \@dblfpsep=8pt plus2fil\relax
```

Note that `\parskip` gets no stretch; this is at variance with the generic L<sup>A</sup>T<sub>E</sub>X classes.

```
248 \parskip=0pt \relax
```

`\@parboxrestore`, used by `\@footnotetext`, sets `\parindent` to 0pt; since this is not what we want, we make a new dimen `\normalparindent` and after calling `\@parboxrestore`, `\@footnotetext` resets `\parindent` back to normal.

```

249 \newdimen\normalparindent
250 <amsart>\normalparindent=12pt
251 <amsproc | amsbook>\normalparindent=18pt
252 \parindent=\normalparindent

253 \partopsep=0pt \relax \parsep=0pt \relax \itemsep=0pt \relax

254 \@lowpenalty=51          \medpenalty=151          \@highpenalty=301
255 \@beginparpenalty=-\@lowpenalty
256 \@endparpenalty=-\@lowpenalty
257 \@itempenalty=-\@lowpenalty

```

### Typesize-specific code

The class option `12pt` sets the main typesize to 12 pt and makes various adaptations, primarily sliding the size-changing commands up the scale of magsteps. This makes it more likely that someone with bitmapped fonts will have all the fonts and sizes that they need. The `8pt` option is for those who like to conserve paper.

By parameterizing some aspects it is possible to make a great deal of the typesize-specific code automatically adapt to the selected size. Then there is so little typesize-specific code remaining that it no longer makes sense to put the code in separate `.clo` files. So instead of analogs for the generic `size10,11,12.clo` files we have the code for those options entirely contained in the `.cls` file in the form of declared options.

Some miscellaneous remarks.

—If PostScript fonts are used, it may seem a little strange to use fonts following the magstep'd point sizes 10.95, 14.4, 17.28, 20.74, 24.88 instead of simply 11, 14, 17, 21, 25. But it is not easy for us to make that distinction here in the document class definitions of the fontsize changing commands. So we don't try.

```

258 \DeclareOption{10pt}{\def\@mainsize{10}\def\@ptsize{0}%
259 \def\@typesizes{%

```

There should be 11 typesize/baselineskip pairs: five below `\normalsize` and five above.

```

260   \or{5}{6}\or{6}{7}\or{7}{8}\or{8}{10}\or{9}{11}%
261   \or{10}{12}% normalsize
262   \or{\@xipt}{13}\or{\@xiipt}{14}\or{\@xivpt}{17}%
263   \or{\@xviipt}{20}\or{\@xxpt}{24}}%
264 \normalsize \linespacing=\baselineskip
265 }
266 %
267 \DeclareOption{11pt}{\def\@mainsize{11}\def\@ptsize{1}%
268 \def\@typesizes{%
269   \or{6}{7}\or{7}{8}\or{8}{10}\or{9}{11}\or{10}{12}%
270   \or{\@xipt}{13}% normalsize
271   \or{\@xiipt}{14}\or{\@xivpt}{17}\or{\@xviipt}{20}%
272   \or{\@xxpt}{24}\or{\@xxvpt}{30}}%
273 \normalsize \linespacing=\baselineskip

```

```

274 }
275 %
276 \DeclareOption{12pt}{\def\@mainsize{12}\def\@ptsize{2}%
277 \def\@typesizes{%
278 \or{7}{8}\or{8}{10}\or{9}{11}\or{10}{12}\or{\@xipt}{13}%
279 \or{\@xiipt}{14}% normalsize
280 \or{\@xivpt}{17}\or{\@xvipt}{20}\or{\@xxpt}{24}%
281 \or{\@xxvpt}{30}\or{\@xxvpt}{30}}%
282 \normalsize \linespacing=\baselineskip
283 }
284 %
285 \DeclareOption{8pt}{\def\@mainsize{8}\def\@ptsize{8}%
286 \def\@typesizes{%
287 \or{5}{6}\or{5}{6}\or{5}{6}\or{6}{7}\or{7}{8}%
288 \or{8}{10}% normalsize
289 \or{9}{11}\or{10}{12}\or{\@xipt}{13}%
290 \or{\@xiipt}{14}\or{\@xivpt}{17}}%
291 \normalsize \linespacing=\baselineskip
292 }
293 %
294 \DeclareOption{9pt}{\def\@mainsize{9}\def\@ptsize{9}%
295 \def\@typesizes{%
296 \or{5}{6}\or{5}{6}\or{6}{7}\or{7}{8}\or{8}{10}%
297 \or{9}{11}% normalsize
298 \or{10}{12}\or{\@xipt}{13}\or{\@xiipt}{14}%
299 \or{\@xivpt}{17}\or{\@xvipt}{20}}%
300 \normalsize \linespacing=\baselineskip
301 }

```

### Running heads

The normal application of pagestyle functions `\ps@xxx` is to determine the contents of running heads and feet. The function `\mkboth` is used internally by commands `\chapter`, `\section`, and the like to set the running heads.

```

302 \def\ps@empty{\let\@mkboth\@gobbletwo
303 \let\@oddhead\@empty \let\@evenhead\@empty
304 \let\@oddfoot\@empty \let\@evenfoot\@empty

```

The current implementation in `amsart/amsproc/amsbook` of the vertical space at the top of an opening page uses `\topskip`, which means that we need to do some resetting here.

```

305 \global\topskip\normaltopskip}

```

Pagestyle ‘plain’ has the page numbers in the running feet.

```

306 \def\ps@plain{\ps@empty
307 \def\@oddfoot{\normalfont\scriptsize \hfil\thepage\hfil}%
308 \let\@evenfoot\@oddfoot}

```

Pagestyle ‘headings’ uses text from sectioning commands for running heads. Empty running feet.

```

309 \newswitch{runhead}

```

```

310 \def\ps@headings{\ps@empty
311   \def\@evenhead{%
312     \setTrue{runhead}%
313     \normalfont\scriptsize
314     \rlap{\thepage}\hfil
315     \def\thanks{\protect\thanks@warning}%
316     \leftmark{}\hfil}%
317   \def\@oddhead{%
318     \setTrue{runhead}%
319     \normalfont\scriptsize \hfil
320     \def\thanks{\protect\thanks@warning}%
321     \rightmark{}\hfil \llap{\thepage}}%
322   \let\@mkboth\markboth
323 \amsbook
324   \def\partmark{\@secmark\markboth\partrunhead\partname}%
325   \def\chaptermark{%
326     \@secmark\markboth\chapterrunhead{}}%
327   \def\sectionmark{%
328     \@secmark\markright\sectionrunhead\sectionname}%
329 \amsbook
330 }

```

`\sectionname` Initialize section headings.

```

\subsectionname 331 \let\sectionname\@empty
\subsubsectionname 332 \let\subsectionname\@empty
\paragraphname 333 \let\subsubsectionname\@empty
\subparagraphname 334 \let\paragraphname\@empty
335 \let\subparagraphname\@empty

```

The default definitions of `\leftmark`, `\rightmark` are not what we want: the section title (or whatever) reported in the right-hand running head should report the section that is current at the bottom of the right-hand page. And the left-hand running head should report the status at the top of the page. Cf. `amspt.sty`.

```

336 \def\leftmark{\expandafter\@firstoftwo\topmark{}}
337 \def\rightmark{\expandafter\@secondoftwo\botmark{}}

```

Journal and Proceedings articles require an indication of the first page so the logo and copyright line can appear.

```

338 \amsart | amsproc
339 \def\ps@firstpage{\ps@plain
340   \def\@oddfoot{\normalfont\scriptsize \hfil\thepage}\hfil

```

Stick in the reset of `\topskip` here so it only gets executed after the first page is completed.

```

341   \global\topskip\normaltopskip}%
342   \let\@evenfoot\@oddfoot
343   \def\@oddhead{\@serieslogo\hss}%
344   \let\@evenhead\@oddhead % in case an article starts on a left-hand page
345 }

```

```
346 </amsart | amsproc>
```

`\@nilgobble` Something that apparently doesn't exist in the kernel?

```
347 \long\def\@nilgobble#1\@nil{}
```

A general section-marking function. Arg 1 is either `\markright` or `\markboth` indicating which kind of marking action is desired (this gives us some string pool/hash table savings by allowing the `\@secmark` function to serve for both cases). Arg 2 is the function that should be called in the running head to process the remaining three args. Arg 3 is normally `\xxxname` (but could be empty). Arg 4 is the section-title text. Assumption: whenever `\@secmark` is called, the section-number variable `\@secnumber` has been set to the value of the current section number (possibly empty, in the case of a `*` section for example).

```
348 <*amsbook>
```

```
349 \def\@secmark#1#2#3#4{%
```

We want to apply expansion to `\xxxname` and `\thexxx` but not to the other elements.

```
350 \begingroup \let\protect\@unexpandable\protect
```

```
351 \edef\@tempa{\endgroup \toks@{\protect#2{#3}{\@secnumber}}}%
```

```
352 \@tempa
```

```
353 \toks@\@xp{\the\toks@{#4}}%
```

If a `\markright` operation is called for, use the current left-mark via `\@temptokena`.

```
354 \afterassignment\@nilgobble\@temptokena\@themark{}\@nil
```

```
355 \edef\@tempa{\@nx\@mkboth{%
```

```
356 \ifx\markright#1\the\@temptokena\else\the\toks@\fi}{\the\toks@}}%
```

```
357 \@tempa}
```

Init `\@secnumber`.

```
358 \let\@secnumber\@empty
```

```
359 </amsbook>
```

Fix `\markboth` so that `\@secmark` can work without too much thrashing.

```
360 \def\markboth#1#2{%
```

```
361 \begingroup
```

```
362 \@temptokena{{#1}{#2}}\xdef\@themark{\the\@temptokena}%
```

```
363 \mark{\the\@temptokena}%
```

```
364 \endgroup
```

```
365 \if@nobreak\ifvmode\nobreak\fi\fi}
```

With the `myheadings` pagestyle, no automatic running heads will be provided by the document class; only running heads specified by the user through explicit `\markboth` or `\markright` commands will be used.

```
366 \def\ps@myheadings{\ps@headings \let\@mkboth\@gobbletwo}
```

Save normal topskip value in a skip register.

```
367 \newskip\normaltopskip
```

```
368 \normaltopskip=10pt \relax
```

We also want to turn off all section marks. First-level section heads will be defined in `\ps@headings`.

```
369 \let\sectionmark@gobble
370 \let\subsectionmark@gobble
371 \let\subsubsectionmark@gobble
372 \let\paragraphmark@gobble
```

### Unrecognized options

The `makeidx` option is redundant; everything that it does in the generic L<sup>A</sup>T<sub>E</sub>X classes is already done anyway in this class.

```
373 \DeclareOption{makeidx}{}
374 \}
```

Unrecognized options for `amsthm` are treated as references to auxiliary theorem setup (`.thm`) files. This allows a user to create theorem styles using internal commands (with `@` signs) without having to be concerned about category coding.

Here is an example from the file `thmtest.tex` which is part of this collection. See that file for further information.

```
\begin{filecontents}{exercise.thm}
\def\th@exercise{%
  \normalfont % body font
  \thm@headpunct{:}%
}
\end{filecontents}
```

This facility is available only when `amsthm` is used as an independent package, not as part of an AMS document class.

```
375 \*amsthm
376 \DeclareOption*{\input{\CurrentOption .thm}}
377 \ProcessOptions
378 \}
```

## 2.3 Process options

Black boxes for overfull lines are turned off by default (the `final` option). This can be overridden with the `draft` option.

```
379 \*classes
380 \ExecuteOptions{leqno,centertags,letterpaper,portrait,%
381 10pt,twoside,onecolumn,final}
```

Options will be processed in the order of the associated `\DeclareOption` commands.

```
382 \ProcessOptions\relax
```

In compatibility mode, we want to load the frozen version of `amstex.sty` instead of the `amsmath` package. This is rather a horrible kluge but I can't see anything better at the moment. [mjd,1995/01/27]

```
383 \ifcompatibility
384 \def@tempa{\RequirePackage{amstex}\relax}%
385 \else
386 \@ifclasswith{\@classname}{nomath}{%
```

```

387 \let\@tempa\relax
388 }{%
389 \def\@tempa{\RequirePackage{amsmath}\relax}%
390 }%
391 \fi
392 \@tempa % load amstex.sty or amsmath.sty

```

If the `nomath` option was specified, then `\numberwithin` and `\emptytoks` remain to be defined.

```

393 \@ifundefined{numberwithin}{%
394 \newcommand{\numberwithin}[3][\arabic]{%
395 \@ifundefined{c@#2}{\@nocounterr{#2}}{%
396 \@ifundefined{c@#3}{\@nocnterr{#3}}{%
397 \addtoreset{#2}{#3}%
398 \xp\xdef\csname the#2\endcsname{%
399 \xp\@nx\csname the#3\endcsname .\@nx#1{#2}}}%
400 }
401 \csname newtoks\endcsname\@emptytoks
402 }{}

```

If the `noamsfonts` option was called for, skip the `amsfonts` package load.

```

403 \if@compatibility
404 \else
405 \@ifclasswith{\@classname}{noamsfonts}{%
406 % amsfonts package is not wanted
407 }{%
408 % amsfonts package IS wanted; test whether a recent enough version
409 % seems to be installed
410 \begingroup \fontencoding{U}\fontfamily{msa}\try@load@fontshape\endgroup
411 \global\@xp\let\csname U+msa\endcsname\relax % reset
412 \@ifundefined{U/msa/m/n}{%
413 \ClassError{\@classname}{%
414 Package ‘amsfonts’ not installed, or version too old?MessageBreak
415 Unable to get font info for the ‘msam’ fonts in the expected form%
416 }{%
417 The amsfonts package will not be loaded, to avoid probableMessageBreak
418 incompatibility problems. You can (a) use the ‘noamsfonts’
419 documentclassMessageBreak
420 option next time, or (b) check that the amsfonts package is
421 installedMessageBreak
422 correctly, and is not too old to be compatible.%
423 }%
424 }{%
425 \RequirePackage{amsfonts}[1995/01/01]\relax
426 }%
427 }
428 \fi % end yesamsfonts branch

```

## 2.4 Basic AMS style features

AMS style requires that blank pages between chapters be *really* blank: no running heads, no page numbers. To accomplish this, redefine `\cleardoublepage`

to do the right thing. [bnb, 1999/07/17]

```
429 \let\cleardouble@page\cleardoublepage
```

Postpone the redefinition of `\cleardoublepage` to begin-document to work around difficulties with old versions of `gsm-1.cls`.

```
430 \AtBeginDocument{%
431   \ifx\cleardouble@page\cleardoublepage
432     \def\cleardoublepage{\clearpage{\pagestyle{empty}\cleardouble@page}}
433   \fi
434 }
```

Now a utility macro to do `\uppercase` but sidestep any math, to prevent uppercasing math variables. In order to be handled properly the `$. . . $` or `\( . . . \)` must be on the outer level (not enclosed in braces). We did not try to handle the possibility `\begin{math} . . . \end{math}` in a title at the present time (too complicated). Also we increase inter-word space in the uppercase text.

One other little problem: uppercasing of a few special characters like the German  $\beta$  (`\ss`) and the undotted *i* and *j* (`\i` and `\j`), used sometimes with accents. We redefine them to be uppercase equivalents. (Undotted `\i` and `\j` in math would be typed as `\imath` and `\jmath`.)

`Spaceskip` is changed in accordance with recommendations for increased interword spacing in all-caps text by e.g. ‘Words into Type’.

```
435 \newcommand{\uppercasenonmath}[1]{\toks@{\@emptytoks}
436 %   Insert an extra \@empty to avoid removing braces around arg \arg{1}.
437   \@xp\@skipmath\@xp\@empty#1$$%
```

The `\protect` here is in case the shorttitle gets used for shortauthors and we get redundant application of `\MakeUppercase`. Double braces limit the scope so that later elements in title block aren’t uppercased, e.g., `\i` in an address. [bnb, 2004/04/01]

```
438 \edef#1{\@nx\protect\@nx\@upprep\the\toks@}}%
439 }
```

`\@upprep` Preparations for printing all-caps text.

```
440 \newcommand{\@upprep}{%
441   \spaceskip1.3\fontdimen2\font plus1.3\fontdimen3\font
442   \upchars@}
```

`\upchars@` In all-caps text, esszet should print as SS, dotless *i* should print as normal cap *I*, *Mc* should print with a small-caps (not lowercase) *c*, and so forth.

```
443 \newcommand{\upchars@}{%
444   \def\ss{SS}\def\i{I}\def\j{J}\def\ae{AE}\def\oe{OE}%
445   \def\o{\0}\def\aa{AA}\def\l{\L}\def\Mc{M{\scshape c}}}
```

`\Mc` The use of `\Mc` makes it possible for ‘*Mc*’ to get special treatment when uppercasing is applied.

```
446 \providecommand{\Mc}{Mc}
```

`\@skipmath` `\@skipmath` searches for `$. . . $` in order to keep from applying `\uppercase` to `\@xskipmath` the contents. Then it calls `\@xskipmath` to search for `\( . . . \)`.

```
447 \newcommand{\@skipmath}{}
448 \long\def\@skipmath#1$#2${%
449   \@xskipmath#1\(\)%
450   \@ifnotempty{#2}{\toks@\xp{the\toks@#2$}\@skipmath\@empty}}%
451 %
452 \newcommand{\@xskipmath}{}
453 \long\def\@xskipmath#1\(#2\){%
454 %   Expand away the added \@empty
455   \uppercase{\toks@\xp{\xp{\xp{\xp{the\@xp\toks@#1}}}}}%
456   \@ifnotempty{#2}{\toks@\xp{the\toks@\(#2\)}\@xskipmath\@empty}}%
```

`\altucnm`

```
457 \def\altucnm#1{%
458   \MakeTextUppercase{\toks@{#1}}%
459   \edef#1{\the\toks@}%
460 }
461 \AtBeginDocument{%
462   \@ifundefined{MakeTextUppercase}{}{\let\uppercasenonmath\altucnm}%
463 }
```

For older versions of L<sup>A</sup>T<sub>E</sub>X this might be needed too:

```
464 \@ifundefined{MakeUppercase}{\let\MakeUppercase\uppercase}{}%
```

`\today` The command `\today` produces today's date in the form most commonly used in the U.S.

```
465 \newcommand{\today}{%
466   \relax\ifcase\month\or
467   January\or February\or March\or April\or May\or June\or
468   July\or August\or September\or October\or November\or December\fi
469   \space\number\day, \number\year}
```

## 2.5 Old font commands

The `\em` command is not redefined here (let's say, to give an 'obsolete' warning and recommend instead `\emph`) because there is no alternative internal command `\emshape`.

```
470 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
471 \DeclareOldFontCommand{\sf}{\normalfont\sfamily}{\mathsf}
472 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
473 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
474 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
475 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
476 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

This warning might have been a good idea back in 1995 but I don't think we can add it now [mjd,2000/03/10].

```
%\ifcompatibility
```

```

%\else
% \def\@obsolete@fontswitch#1#2#3{%
%   \@latex@warning@no@line{%
%     Command {\string#1...}\on@line\space is obsolete;\MessageBreak
%     the LaTeX2e equivalent is \string#3{...}}%
%   \gdef#1{\@fontswitch\relax#3}%
% }
% \DeclareRobustCommand*\cal{%
%   \@xp\@obsolete@fontswitch\csname cal \endcsname\relax\mathcal}
% \DeclareRobustCommand*\mit{%
%   \@xp\@obsolete@fontswitch\csname mit \endcsname\relax\mathnormal}
%\fi

```

There's too great a chance that some people out there have documents that begin with

```

\documentclass{amsart}
...
\newcommand{\cal}{\mathcal}

```

and such documents would now get error messages.

Since `\cal` is not documented as a valid command for  $\text{\LaTeX} 2_{\epsilon}$ , I think it is OK to leave the status as is for AMS document classes. If you use `\documentstyle`, `\cal` will work. If you use `\documentclass`, well, it's a good idea to update to `\mathcal` instead of leaving old instances of `\cal`.

## 2.6 Top matter

For the `\title` command, we support an optional argument to give a shortened version of the title for running heads.

```
477 \renewcommand*\title}[2] []{\gdef\shorttitle{#1}\gdef\@title{#2}}
```

The default value for the optional argument is 'same as the mandatory arg' but there doesn't seem to be an easy way to get that effect with `\[re]newcommand`. Here is how to use `\@dblarg` in conjunction with the preceding `\newcommand`:

```
478 \edef\title{\@nx\@dblarg
479   \@xp\@nx\csname\string\title\endcsname}
```

The `\author` command accepts an optional argument similar to that of the `\title` command. Moved update of `\addresses` within scope of `\else` to avoid adding anything if no authors, and thus avoiding output of "Author address" on monograph titlepage. [bnb, 1996/11/03]

```
480 \renewcommand*\author}[2] []{%
481   \ifx\@empty\authors
482     \gdef\authors{#2}%
483   \else
484     \g@addto@macro\authors{\and#2}%
485     \g@addto@macro\addresses{\author{}}%
486   \fi
487   \@ifnotempty{#1}{%
488     \ifx\@empty\shortauthors
489       \gdef\shortauthors{#1}%

```

```

490   \else
491     \g@addto@macro\shortauthors{\and#1}%
492   \fi
493 }%
494 }
495 \edef\author{\@nx\@dblarg
496   \@xp\@nx\csname\string\author\endcsname}

Initialize some variables.

497 \let\shortauthors\@empty   \let\authors\@empty

```

`\contrib` Contributors are similar to authors except that they are responsible for only part of a work, e.g., an appendix. The optional argument for the first contributor of a group identifies what has been contributed. There can be more than one group of contributors; each group is treated separately, using the same “and” conventions within the group as for authors. Contributor groups are strung together separated by a comma; if the word “and” is desired before the final group of contributors, it must be included in the optional argument for that group.

```

498 (*amsart | amsproc)
499 \newif\ifresetcontrib \resetcontribfalse
500 \newcommand\contrib[2] [] {%
501   \def\@tempa{#1}%
502   \ifx\@empty\@tempa
503     \else
504       \ifresetcontrib \@xcontribs
505       \else \global\resetcontribtrue
506     \fi
507   \fi
508   \ifx\@empty\contribs
509     \gdef\contribs{#1 #2}%
510   \else
511     \g@addto@macro\contribs{\and#1 #2}%
512   \fi

```

Accumulate contribs separately for the table of contents. Here, this is just a dummy; it is fully defined for in-house processing.

```

513   \@wraptococontribs{#1}{#2}%
514 }
515 \def\wraptococontribs#1#2{}
516 \def\@xcontribs{%
517   \author@andify\contribs
518   \ifx\@empty\xcontribs
519     \xdef\xcontribs{\contribs}%
520   \else
521     \xdef\xcontribs{\xcontribs, \contribs}%
522   \fi
523   \let\contribs\@empty
524 }

```

Initialize some more variables.

```
525 \let\contribs\@empty \let\xcontribs\@empty \let\tocontribs\@empty
526 </amsart|amsproc>
527 \let\addresses\@empty \let\thankses\@empty
```

The optional arguments of `\address`, `\curraddr`, `\email` are to indicate which author the address applies to, if a document has multiple authors and there is not a normal one-to-one correspondence between authors and addresses.

```
528 \newcommand{\address}[2] [] {\g@addto@macro\addresses{\address{#1}{#2}}}
529 \newcommand{\curraddr}[2] [] {\g@addto@macro\addresses{\curraddr{#1}{#2}}}
530 \newcommand{\email}[2] [] {\g@addto@macro\addresses{\email{#1}{#2}}}
531 \newcommand{\urladdr}[2] [] {\g@addto@macro\addresses{\urladdr{#1}{#2}}}
```

Someone who does not look closely at the `amsart` documentation is likely to put the `\thanks` command inside that argument of `\author`.

```
532 \long\def\thanks@warning#1{%
533 \ClassError{\@classname}{%
534 \protect\thanks\space should be given separately, not inside author name.%
535 } \@ehb
536 }

537 \renewcommand{\thanks}[1] {%
538 \@ifnotempty{#1}{\g@addto@macro\thankses{\thanks{#1}}}%
539 }
```

The following example of addresses for three authors of a tri-author paper illustrates the kind of complications that need to be handled.

```
\author{Roland Campbell}
\address{Department of Mathematics\
  Pennsylvania State University\
  Pittsburgh, Pennsylvania 13593}
\email[R.~Campbell]{campr@galois.psu.edu}

\author{Mark M. Dane}
% Same address as R. Campbell
\curraddr[M.~Dane]{Atmospheric Research Station\
  Pala Lundi, Fiji}
\email[M.~Dane]{DaneMark@ffr.choice}

\author{Jeremiah Jones}
\address[J.~Jones]{Department of Philosophy\
  Freedman College\
  Periwinkle, Colorado 84320}
\email[J.~Jones]{id739e@oseoi44 (Bitnet)}
```

In an article, typesetting of the address information is done at the end of the document, by calling `\@setaddresses`. This is done through a parent function `\enddoc@text`, because some AMS journals also print the abstract there instead of at the beginning, and it's easier to redefine `\enddoc@text` than to try undoing material already added to the `\AtEndDocument` hook.

```
540 < *amsart | amsproc >
```

```

541 \def\enddoc@text{\ifx\@empty\@translators \else\@settranslators\fi
542 \ifx\@empty\addresses \else\@setaddresses\fi}
543 \AtEndDocument{\enddoc@text}
544 </amsart | amsproc>

```

```

545 \def\curraddrname{\itshape Current address}
546 \def\emailaddrname{\itshape Email address}
547 \def\urladdrname{\itshape URL}
548 \def\@setaddresses{\par
549 \nobreak \begingroup
550 <amsart | amsproc>\footnotesize
551 \def\author##1{\nobreak\addvspace\bigskipamount}%

```

Address is supposed to go all on one line, so we redefine \\ to just insert a comma instead of doing a line break.

```
552 \def\\{\unskip, \ignorespaces}%
```

No page breaks in the address section is accomplished by `\interlinepenalty\@M` and by the `\nobreak` before the `\bigskip`.

```

553 \interlinepenalty\@M
554 \def\address##1##2{\begingroup

```

If there are two addresses for the same author, add a `\bigskip` between them.

```
555 \par\addvspace\bigskipamount\indent
```

If the name of the author to whom this address applies was given, typeset it.

```
556 \@ifnotempty{##1}{(\ignorespaces##1\unskip) }%
```

Now the main part of the address:

```
557 {\scshape\ignorespaces##2}\par\endgroup}%
```

Current address:

```

558 \def\curraddr##1##2{\begingroup
559 \@ifnotempty{##2}{\nobreak\indent\curraddrname
560 \@ifnotempty{##1}{, \ignorespaces##1\unskip}\:/\space
561 ##2\par}\endgroup}%

```

And then email. In versions 1.0 and 1.1 @@ was required to print a single @ character; for bulletproofing we convert doubled @ characters if found.

```

562 \def\email##1##2{\begingroup
563 \@ifnotempty{##2}{\nobreak\indent\emailaddrname
564 \@ifnotempty{##1}{, \ignorespaces##1\unskip}\:/\space
565 \ttfamily##2\par}\endgroup}%

```

URLaddr is simply a replica of the email address, with the addition of a feature to enable ~ to print.

```

566 \def\urladdr##1##2{\begingroup
567 \def~{\char`~}%
568 \@ifnotempty{##2}{\nobreak\indent\urladdrname
569 \@ifnotempty{##1}{, \ignorespaces##1\unskip}\:/\space
570 \ttfamily##2\par}\endgroup}%
571 \addresses
572 \endgroup
573 }

```

Some other administrative info. For `\date` we can just use the default definition provided by L<sup>A</sup>T<sub>E</sub>X, except that we initialize the date to empty instead of to `\today`.

```
574 \let\@date\@empty
575 \def\dedicatory#1{\def\@dedicatory{#1}}
576 \let\@dedicatory=\@empty
577 \def\keywords#1{\def\@keywords{#1}}
578 \let\@keywords=\@empty
```

To allow various versions of the subject classification, accept an optional value to identify the version, provide text for the two currently in use, and give a warning if the version specified is unknown. Default to 1991 version. [bnb, 1999/04/30]

```
579 \newcommand*\subjclass[2][1991]{%
580   \def\@subjclass{#2}%
581   \ifundefined{subjclassname@#1}{%
582     \ClassWarning{\@classname}{Unknown edition (#1) of Mathematics
583       Subject Classification; using '1991'.}%
584   }{%
585     \exp\let\@xp\subjclassname\csname subjclassname@#1\endcsname
586   }%
587 }
588 \let\@subjclass=\@empty
589 \amsart\def\commby#1{\def\@commby{(Communicated by #1)}}
590 \amsart\let\@commby=\@empty
```

We handle translator names like author names, just in case there is more than one translator. [mjd,1994/10/19]

```
591 \def\translname{Translated by}
592 \def\translator#1{%
593   \ifx\@empty\@translators \def\@translators{#1}%
594   \else\g@addto@macro\@translators{\and#1}\fi}
595 \let\@translators=\@empty

596 <*amsart | amsproc>
597 \def\@settranslators{\par\begin{group}
598   \addvspace{6\p@\@plus9\p@}%
599   \hbox to\columnwidth{\hss\normalfont\normalsize
600     \translname{ }%
601     \andify\@translators \uppercase\nonmath\@translators
602     \@translators}
603   \endgroup
604 }
605 </amsart | amsproc>
```

The general function to convert a list of items in the form

`A\and B\and C\and D`

to the form ‘A, B, C, and D’ is `\xandlist`:

`\xandlist{, }{ and }{, and }{A\and B\and C\and D}`

This is a completely expandable macro, with the return value being the converted list. There is also a ‘no-execute’ version whose fourth argument should be a macro; the text to be converted will be taken from that macro and after conversion will be put back as the macro’s new replacement text.

```
\nxandlist{, }{ and }{, and }\result
```

I don’t think I want to explain this except by recommending that you watch it in operation with `\tracingmacros` if you’re interested. [mjd,1994/10/19]

```
606 \newcommand{\xandlist}[4]{\@andlista{#1}{#2}{#3}#4\and\and}
607 \def\@andlista#1#2\and#3\and{\@andlistc{#2}\@ifnotempty{#3}{%
608   \@andlistb#1{#3}}
609 \def\@andlistb#1#2#3#4#5\and{%
610   \@ifempty{#5}{%
611     \@andlistc{#2#4}%
612   }{%
613     \@andlistc{#1#4}\@andlistb{#1}{#3}{#3}{#5}%
614   }}
615 \let\@andlistc\@iden
616 \newcommand{\nxandlist}[4]{%
617   \def\@andlistc##1{\toks@\xp{\the\toks@##1}}%
618   \toks@{\toks@\@emptytoks \@andlista{#1}{#2}{#3}}%
619   \the\@xp\toks@#4\and\and
620   \edef#4{\the\toks@}%
621   \let\@andlistc\@iden}
```

**andify** The `\andify` function is provided as a convenient abbreviation for the most common case. See also `\author@andify` (for `amsart` and `amsproc` only), which gives better results in cases with a large number of authors. Provide a substitutable text string to simplify language-specific modifications.

```
622 \def\@@and{and}
623 \newcommand{\andify}{%
624   \nxandlist{\unskip, }\unskip{ } \@@and~}{\unskip, \@@and~}}
```

Override the funny default definition of `\and` from L<sup>A</sup>T<sub>E</sub>X. This is not actually used by AMS classes, however.

```
625 \def\and{\unskip{ } \@@and{ } \ignorespaces}
```

**maketitle** Set up the style of an article opening page. For books, see below. For articles, we must add the copyright info footnote.

```
626 (*amsart | amsproc)
627 \def\maketitle{\par
628   \@topnum\z@ % this prevents figures from falling at the top of page 1
629   \@setcopyright
630   \thispagestyle{firstpage}% this sets first page specifications
```

Do some setup for the running heads here. If there are no author names, we set the left-hand running head to the value of the right-hand running head.

```
631   \uppercase\nonmath\shorttitle
632   \ifx\@empty\shortauthors \let\shortauthors\shorttitle
```

```
633 \else \andify\shortauthors
634 \fi
```

The following hook is used to activate the writing of author and title information to an ‘issue table of contents’ when multiple articles are being processed for a journal issue or a proceedings volume.

```
635 \@maketitle@hook
636 \begingroup
637 \@maketitle
638 \toks@{\xp{\shortauthors}\@temptokena\@xp{\shorttitle}}%
639 \toks4{\def\{\ignorespaces}}% defend against questionable usage
640 \edef\@tempa{%
641   \nx\markboth{\the\toks4
642     \nx\MakeUppercase{\the\toks@}}{\the\@temptokena}}%
643 \@tempa
644 \endgroup
645 \c@footnote\z@
646 \cleartopmattertags
647 }
648 \def\cleartopmattertags{%
649   \def\do##1{\let##1\relax}%
650   \do\maketitle \do\@maketitle \do\title \do\@xtitle \do\@title
651   \do\author \do\@xauthor \do\address \do\@xaddress
652   \do\contrib \do\contribs \do\@xcontribs \do\tocontribs
653   \do\email \do\@xemail \do\curraddr \do\@xcurraddr
654 \amsart) \do\commby \do\@commby
655   \do\dedicatory \do\@dedicatory \do\thanks \do\thankses
656   \do\keywords \do\@keywords \do\subclass \do\@subclass
657 }
658 \amsart | amsproc)
```

The hook `\@maketitle@hook` is placed into `\maketitle` rather than `\@maketitle` because the latter tends to get redefined by derived classes using this one as a base. The initial motivation for this hook is to extract title and author information to an external file, so we can’t do it with `\AtBeginDocument: \title` and `\author` commands might occur between `\begin{document}` and `\maketitle`.

```
659 \amsart | amsproc)
660 \def\@maketitle@hook{\global\let\@maketitle@hook\@empty}
661 \amsart | amsproc)
```

Set up the style of an article opening page.

```
662 \amsart | amsproc)
663 \def\@maketitle{%
```

Set font to normal, just in case.

```
664 \normalfont\normalsize
```

Special footnotes are put here to ensure that they come first at the bottom of the page.

```
665 \@adminfootnotes
```

If `\pagestyle{myheadings}` was specified, `\@mkboth` will be a no-op.

```
666 \@mkboth{\@nx\shortauthors}{\@nx\shorttitle}%
667 <amsproc> \global\topskip8pc\relax % 10pc to base of first title line
668 <amsart> \global\topskip42\p@\relax % 5.5pc " " " " "
669 \@settitle
670 \ifx\@empty\authors \else \@setauthors \fi
```

Likewise with `\@dedicatory` and `\@date`.

```
671 \ifx\@empty\@dedicatory
672 \else
673 <amsproc> \baselineskip26\p@
674 <amsart> \baselineskip18\p@
675 \vtop{\centering{\footnotesize\itshape\@dedicatory\@par}%
676 \global\dimen@i\prevdepth}\prevdepth\dimen@i
677 \fi
678 \@setabstract
```

Space before the main text should be 32 + 14 base-to-base; we accomplish this by doing a `vskip` of that amount with `\baselineskip` subtracted.

```
679 \normalsize
680 \if@titlepage
681 \newpage
682 \else
683 \dimen@34\p@ \advance\dimen@-\baselineskip
684 \vskip\dimen@\relax
685 \fi
686 } % end \@maketitle
```

Segregate the definitions of administrative footnotes to permit easier customization, especially for translation journals.

```
687 \def\@adminfootnotes{%
688 \let\@makefnmark\relax \let\@thefnmark\relax
689 <amsart> \ifx\@empty\@date\else \@footnotetext{\@setdate}\fi
690 \ifx\@empty\@subjclass\else \@footnotetext{\@setsubjclass}\fi
691 \ifx\@empty\@keywords\else \@footnotetext{\@setkeywords}\fi
```

In order to make multiple thanks footnotes work inside a single `\@footnotetext` argument we need to make the first `\par` be ignored. Cf. `\@setthanks`.

```
692 \ifx\@empty\thankses\else \@footnotetext{%
693 \def\par{\let\par\@par}\@setthanks}%
694 \fi
695 }
696 </amsart | amsproc>
```

## 2.7 Journal/series logo for articles

`\publname` either will be defined by a parent class that is calling `amsart` or `amsproc` as a base class, or will be absent, in which case this can be used as a signal to omit the `\@serieslogo`. A typical value for `\publname` would be

```
\def\publname{JOURNAL OF THE\newline
AMERICAN MATHEMATICAL SOCIETY}
```

Initialize `\publname` and `\@serieslogo` to no-op if `\publname` is not already defined.

```

697 <*amsart | amsproc>
698 \AtBeginDocument{%
699   \@ifundefined{publname}{%
700     \let\publname\@empty
701     \let\@serieslogo\@empty
702   }{%
703 <amsart>   \def\@serieslogo{\article@logo}%
704 <amsproc>  \def\@serieslogo{\procart@logo}%
705   }%
706 }
707 </amsart | amsproc>

```

The `\number` prefix on current issue is to work around inconsistencies in the form of issue numbers as passed in from system level. Sometimes they will get passed in with a leading zero, which we don't want to print if it happens to occur. The 0 prevents an error if `\currentissue` happens to be empty.

```

708 <*amsart>
709 \AtBeginDocument{%
710   \@ifundefined{volinfo}{%
711     \def\volinfo{%
712       Volume \currentvolume, Number \number0\currentissue

```

Month/year is not included initially when a journal article is posted on the WWW prior to print publication.

```

713     \if\@printyear , \currentmonth\ \currentyear\fi
714   }%
715 }{}%
716 }
717 \def\@printyear{TF}% boolean false
718 </amsart>
719 <*amsproc>
720 \AtBeginDocument{%
721   \@ifundefined{volinfo}{\let\volinfo\@empty}{}
722 }
723 </amsproc>

```

Default values for information such as volume, year, and so on are provided as follows.

```

724 <*amsart | amsproc>
725 \def\issueinfo#1#2#3#4{\def\currentvolume{#1}\def\currentissue{#2}%
726   \def\currentmonth{#3}\def\currentyear{#4}}
727 \issueinfo{00}% volume number
728 {0}%           % issue number
729 {Xxxx}%        % month
730 {XXXX}%        % year
731 </amsart | amsproc>

```

Copyright year may be different from issue year. Allow it to be specified separately. It is probably more natural anyway, from the user's perspective,

to give the copyright year in the same command when giving the name of the copyright holder.

```
732 \newcommand{\copyrightinfo}[2]{%
733   \def\copyrightyear{#1}%
734   \@ifnotempty{#2}{\def\copyrightholder{#2}}%
735 }
736 \copyrightinfo{0000}{(copyright holder)}
```

Provide page span information. If negative number is given, convert to roman numeral form.

```
737 <*amsart | amsproc>
738 \def\pagespan#1#2{\setcounter{page}{#1}%
739   \ifnum\c@page<\z@ \pagenumbering{roman}\setcounter{page}{-#1}\fi
740   \def\start@page{#1}\def\end@page{#2}}
741 \pagespan{000}{000}
742 </amsart | amsproc>
```

Formatting for journal page numbers. [bnb, 1996/09/11]

```
743 <*amsart>
744 \AtBeginDocument{%
745   \@ifundefined{pageinfo}{%
746     \def\pageinfo{%
747       \ifnum\start@page=\z@
748         Pages 000--000
749       \else
750         \ifx\start@page\end@page
751           Page \start@page
752         \else
753           Pages \start@page--\end@page
754         \fi
755       \fi}%
756   }{}%
757 }
758 </amsart>
```

Publisher Item Identifier (we started using them in journal logos as of January 1997).

```
759 <amsart>\@ifundefined{ISSN}{\def\ISSN{0000-0000}}{}
760 <amsart>\newcommand\PII[1]{\def\@PII{#1}}
761 <amsart>\PII{S \ISSN{XX}0000-0}
```

## 2.8 Copyright block

Doing the copyright info on the first page is a little tricky. We want it to come at the bottom, after any footnotes and floating inserts, but before the page number. If we simply put it into `\@oddfloat` (in `\ps@plain`) its height will not be subtracted from the height of the text and then the page number will be lower than we want. So we do it as an insert.

Through version 1.2, this code was included only for `amsart` and `amsproc`; the formatting of some book series requires an insert at the bottom of the text

block, so this code has been extended to all AMS document classes. [bnb, 1999/07/14]

```
762 \newinsert\copyins
```

We set the skip register associated with this insert to the *base-to-base* distance from the bottom of the page contents to the base of the first line in the copyright info. See the definition of `\@setcopyright`.

```
763 \skip\copyins=1.5pc
```

```
764 \count\copyins=1000 % magnification factor, 1000 = 100%
```

```
765 \dimen\copyins=.5\textheight % maximum allowed per page
```

`\copyins` is ignored if a float is input on the first page; adding it to `\@reinserts` will make the output routine behave. [bnb; 2004/06/09; B-365]

```
766 \g@addto@macro\@reinserts{%
```

```
767   \ifvoid\copyins\else\insert\copyins{\unvbox\copyins}\fi
```

```
768 }
```

Put the contents into a  $\TeX$  insert. This information is omitted unless `\@serieslogo` is non-null. In other words it will normally not print except when an AMS publication-specific document class such as `tran-1` is used. And even if the series logo is printed, omit the copyright line if requested by `\copyrightinfo{}{}`. [bnb, 1996/10/17]

```
769 \def\@copyinsfontsize{\fontsize{6}{7\p@}\normalfont\upshape}
```

```
770 \newif\if@extracrline \@extracrlinefalse
```

```
771 \let\@extracrline\@empty
```

```
772 \relax
```

```
773 \def\@setcopyright{%
```

```
774   \ifx\@empty\@serieslogo
```

```
775   \else\ifx\@empty\copyrightyear
```

```
776   \else
```

```
777     \insert\copyins{\hsize\textwidth
```

```
778       \parfillskip\z@\relax
```

```
779       \leftskip\z@\@plus.9\textwidth\relax \rightskip\z@\relax
```

```
780       \@copyinsfontsize
```

The spacing between the preceding text and the copyright info is done with a strut of height `\skip\copyins`. (Note that `\lineskip` and `\baselineskip` are 0 in the  $\LaTeX$  output routine.) The negative `vskip` gives an effective distance of 0 from the top of the box to the base of the first line (assuming `\skip\copyins` is greater than the height of that line). Then the apparent total height of the box will work well with  $\TeX$ 's calculations involving `\skip\copyins` for how much room to leave for this object. An extra 6pt is allowed when an additional line is present; this adjustment was found adequate in some borderline cases where tight pages reset with the additional line had text lines moved to the next page, causing the paper length to expand by a page. [bnb, 2004/05/07-06/24]

```
781   \everypar{}%
```

```
782   \vskip-\skip\copyins
```

```
783   \if@extracrline
```

```
784     \vskip-6pt
```

```
785   \fi
```

```

786     \nointerlineskip
787     \leavevmode\hfill\vrule\@width\z@\@height\skip\copyins
788     \copyright\copyrightyear\ \copyrightholder\ignorespaces
789     \if@extracrine \@extracrine \fi
790     \par

```

This kern of 0pt forces the depth of the last line (if any) to be added to the height of the box.

```

791     \kern\z@}%
792 \fi\fi
793 }

```

When `\@combinefloats` is called, the box `\@outputbox` already contains the main text of the page and any footnotes. Then L<sup>A</sup>T<sub>E</sub>X adds top and bottom figures. We want to add our copyright info at the very bottom, but still inside of the vbox.

```

794 \def\@combinefloats{%
795   \ifx \@toplist\@empty \else \@cflt \fi
796   \ifx \@botlist\@empty \else \@cflb \fi
797   \ifvoid\copyins \else \@cflci \fi
798 }

```

#### **combine-floats-copyright-insert**

In the twocolumn/firstcolumn case, postpone adding the drop folio. Put an empty box of the same height at the bottom of the left-hand column to make the columns balance. Allow a smidge of stretch in case the first page of a chapter has no internal stretch, so that the drop folio will be flush to the bottom of the text block; don't do this for a twocolumn page, to avoid an uneven bottom.

```

799 \def\@cflci{%
800   \setbox\@outputbox\vbox{%
801     \unvbox\@outputbox
802     \vskip\skip\copyins
803     \if@twocolumn \else \vskip\z@ plus\p@ \fi
804     \hbox to\columnwidth{%
805       \hss\vbox to\z@\vss

```

In two-column layout, put an empty box in the first column instead of the drop folio.

```

806     \if@twocolumn
807       \if@firstcolumn \else \unvbox\copyins \fi
808     \else
809       \unvbox\copyins
810     \fi
811   }}}}

```

Now redo the insert to make sure we get the right amount of space reserved for it in the second column.

```

812 \if@twocolumn \if@firstcolumn
813   \insert\copyins{\unvbox\copyins}%
814 \fi\fi

```

Reset the `\copyins` flag so that a subsequent insert (e.g. `\@dropfolio` in some book series) will work.

```
815 \global\count\copyins=999 \relax
816 }
```

[End of code to support inserts at end of text block.]

For journals only, provide a switch that indicates the author has agreed to revert copyright to the public domain; this results in an addition to the copyright block on the article.

```
817 <amsart>
818 \newif\if@revertcopyright \@revertcopyrightfalse
819 \newcommand{\revertcopyright}{%
820 \global\@revertcopyrighttrue
821 \global\@extracrlinetrue}
```

Add notation regarding reversion of copyright to public domain if author has agreed to it. Permit this to be set in a different size than the copyright line (required for some author packages).

```
822 \def\@revertcrfontsize{\fontsize{6}{7p@}\normalfont\upshape}
823 \def\@extracrline{%
824 \if@revertcopyright
825 \unskip\
826 \@revertcrfontsize
827 Reverts to public domain 28 years from publication
828 \fi
829 }
830 </amsart>
```

## 2.9 Other top matter info

Some name setup.

```
831 \newcommand{\abstractname}{Abstract}
832 \newcommand{\keywordsname}{Key words and phrases}
```

For now, support just the two most recent versions; earlier versions were:

- 1980 Mathematics Subject Classification
- 1980 Mathematics Subject Classification (1985 Revision)

Default to the 1991 edition, as that's what is in production at the time this change is made. [bnb, 1999/04/30]

```
833 \newcommand{\subjclassname}{%
834 \textup{1991} Mathematics Subject Classification}
835 \exp\let\csname subjclassname@1991\endcsname \subjclassname
836 \@namedef{subjclassname@2000}{%
837 \textup{2000} Mathematics Subject Classification}
838 \@namedef{subjclassname@2010}{%
839 \textup{2010} Mathematics Subject Classification}
```

For the date we have a special little problem: We only want to add the 'Received by the editors' text for publication-specific document classes such as `tran-1`.

```
840 <amsbook>\def\@tempb{amsbook}
```

```

841 <amsproc>\def\@tempb{amsproc}
842 <amsart>\def\@tempb{amsart}
843 \ifx\@classname\@tempb
844   \newcommand{\datename}{\textit{Date}:}
845 \else
846   \newcommand{\datename}{Received by the editors}
847 \fi

```

```

848 <*amsart | amsproc>
849 \def\@settitle{\begin{center}%
850 <amsart> \baselineskip14\p@\relax
851 <amsproc> \Large
852   \bfseries
853 <amsart>\uppercase\@title
854   \@title
855   \end{center}%
856 }
857 </amsart | amsproc>

```

For multiple authors we need to combine the author names into a list of the form Author One, Author Two, . . . , and Author Last. Change line breaking penalties to avoid a line break in the middle of an author name if there are a lot of authors. This should probably better be done by changing spaces within author names to `\nolinebreak[3]\space`, but that would take more work. [mjd,2000/12/27]

```

858 <*amsart | amsproc>
859 \def\author@andify{%
860   \xandlist {\unskip ,\penalty-1 \space\ignorespaces}%
861     {\unskip } \@and~}%
862   {\unskip ,\penalty-2 \space \@and~}%
863 }

864 \def\@setauthors{%
865   \begingroup
866   \def\thanks{\protect\thanks@warning}%
867   \trivlist
868 <amsart> \centering\footnotesize \@topsep30\p@\relax
869 <amsart> \advance\@topsep by -\baselineskip
870 <amsproc> \centering \fontsize{11}{13\p@}\selectfont\@topsep13\p@\relax
871   \item\relax
872   \author@andify\authors

```

Make `dbl-backslash` robust to prevent problems if/when `\MakeTextUppercase` expansion is applied.

```

873 \def\{\protect\linebreak}%
874 <amsart> \MakeUppercase{\authors}%
875 <amsproc> \authors
876 \ifx\@empty\contribs
877 \else
878   ,\penalty-3 \space \@setcontribs
879   \@closetocontribs

```

```

880 \fi
881 \endtrivlist
882 \endgroup
883 }
884 \def\@closetocontribs{}
885 \def\@setcontribs{%
886 \xcontribs
887 \amsart \MakeUppercase{\xcontribs}%
888 \amsproc \xcontribs
889 }
890 \def\@setdate{\datename \ @date \@addpunct.}
891 \def\@setsubjclass{%
892 {\itshape\subjclassname.}\enspace \@subjclass \@addpunct.}
893 \def\@setkeywords{%
894 {\itshape\keywordsname.}\enspace \@keywords \@addpunct.}
895 \amsart | \amsproc

```

The following definitions suffice for all the AMS document classes.

```

896 \def\@setthanks{\def\thanks##1{\par##1 \@addpunct.}\thankses}

```

‘Abstract’ can be changed to, say, ‘Résumé’ (French) by redefining `\abstractname`. This and other control sequence names (`\refname`, `\contentsname`, `\appendixname`, and so on) are compatible with the `babel` package (the AMS sometimes publishes articles in languages other than English.)

In AMS document classes, the abstract should be placed before `\maketitle` (otherwise the desired ordering of frontmatter elements cannot be ensured in all cases).

```

897 \newbox\abstractbox

```

We start by checking whether `\maketitle` has already been used (in which case it was reset to `\relax`); if so, we give a warning that the abstract should be placed before `\maketitle`.

```

898 \newenvironment{abstract}{%
899 \ifx\maketitle\relax
900 \ClassWarning{\@classname}{Abstract should precede
901 \protect\maketitle\space in AMS document classes; reported}%
902 \fi
903 \global\setbox\abstractbox=\vtop \bgroup
904 \normalfont\Small
905 \list{}{\labelwidth\z@
906 \leftmargin3pc \rightmargin\leftmargin
907 \listparindent\normalparindent \itemindent\z@
908 \parsep\z@ \@plus\p@

```

In order to get equation numbers indented with the rest of the abstract, we have to do this:

```

909 \let\fullwidthdisplay\relax
910 }%
911 \item[\hskip\labelsep\scshape\abstractname.]%
912 }{%
913 \endlist\egroup

```

If the abstract was supposed to be typeset earlier, then `\@setabstract` is now equal to `\relax`, and we had better drop the contents of the abstract box onto the page immediately, to salvage the situation as best we can.

```
914 \ifx\@setabstract\relax \setabstracta \fi
915 }
```

Because the abstract might be postponed until the end of an article, we cannot simply use the fact of a preceding `\maketitle` to tell whether `\endabstract` should immediately put the abstract box contents on the page. So we use an auxiliary function that will be reset to no-op once we have passed the point where the abstract should normally be typeset.

```
916 \def\@setabstract{\@setabstracta \global\let\@setabstract\relax}
917 \def\@setabstracta{%
918   \ifvoid\abstractbox
919   \else
920     \skip@20\p@ \advance\skip@-\lastskip
921     \advance\skip@-\baselineskip \vskip\skip@
922     \box\abstractbox
923     \prevdepth\z@ % because \abstractbox is a vtop
924   \fi
925 }
```

Title page environment does nothing much; information and formatting to be provided by the user.

```
926 \def\titlepage{%
927   \amsbook \cleardoublepage
928   \amsart | \amsproc \clearpage
929   \thispagestyle{empty}\setcounter{page}{0}}
930 \def\endtitlepage{\newpage}
```

## 2.10 Macros for list labels

Through version 1.2, first-level enumerated item labels were formatted with a following period, which is not AMS style. Effective with version 2.0 these labels are formatted with parentheses. Anyone requiring the period style will have to redefine `\labelenumi`.

```
931 \def\labelenumi{(\theenumi)}
932 \def\theenumi{\@arabic\c@enumi}
933 \def\labelenumii{(\theenumii)}
934 \def\theenumii{\@alph\c@enumii}
935 \def\p@enumii{\theenumi}
936 \def\labelenumiii{(\theenumiii)}
937 \def\theenumiii{\@roman\c@enumiii}
938 \def\p@enumiii{\theenumi(\theenumii)}
939 \def\labelenumiv{(\theenumiv)}
940 \def\theenumiv{\@Alph\c@enumiv}
941 \def\p@enumiv{\p@enumiii\theenumiii}

942 \def\labelitemi{\m@th\bullet}
943 \def\labelitemii{\bfseries --}% \upshape already done by \itemize
944 \def\labelitemiii{\m@th\ast}
```

```
945 \def\labelitemiv{$\m@th\cdot$}
```

## 2.11 Verse and quotation environments

```
946 \newenvironment{verse}{\let\\\@centercr
947 \list{}{\itemsep\z@ \itemindent -1.5em\listparindent\itemindent
948 \rightmargin\leftmargin \advance\leftmargin 1.5em}\item[]%
949 }{%
950 \endlist
951 }
952 \let\endverse=\endlist % for efficiency
```

The left/right margins of the quotation environment are supposed to be the same as for the abstract environment.

```
953 \newenvironment{quotation}{\list{}{%
954 \leftmargin3pc \listparindent\normalparindent
955 \itemindent\z@
956 \rightmargin\leftmargin \parsep\z@ \@plus\p@}%
957 \item[]%
958 }{%
959 \endlist
960 }
961 \let\endquotation=\endlist % for efficiency
962 \newenvironment{quote}{%
963 \list{}{\rightmargin\leftmargin}\item[]%
964 }{%
965 \endlist
966 }
967 \let\endquote=\endlist % for efficiency
```

## 2.12 List environments

Changed definition of `\trivlist`, `enumerate`, and `itemize` in order to have `\makelabel` apply `\upshape`.

```
968 \def\trivlist{\parsep\parskip\@nmbrlistfalse
969 \@trivlist \labelwidth\z@ \leftmargin\z@
970 \itemindent\z@
971 \let\@itemlabel\@empty
972 \def\makelabel##1{\upshape##1}}
973 \renewenvironment{enumerate}{%
974 \ifnum \@enumdepth >3 \@toodeep\else
975 \advance\@enumdepth \@ne
976 \edef\@enumctr{enum\romannumeral\the\@enumdepth}\list
977 {\csname label\@enumctr\endcsname}{\usecounter
978 {\@enumctr}\def\makelabel##1{\hss\llap{\upshape##1}}}\fi
979 }{%
980 \endlist
981 }
982 \let\endenumerate=\endlist % for efficiency
983 \renewenvironment{itemize}{%
984 \ifnum\@itemdepth>3 \@toodeep
```

```

985 \else \advance\@itemdepth\@ne
986 \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
987 \list{\csname\@itemitem\endcsname}%
988 {\def\makelabel##1{\hss\llap{\upshape##1}}}%
989 \fi
990 }{%
991 \endlist
992 }
993 \let\enditemize=\endlist % for efficiency

994 \newcommand{\descriptionlabel}[1]{\hspace\labelsep \upshape\bfseries #1:}
995 \newenvironment{description}{\list{}}{%

```

Adjust the indent of the first line to the desired value:

```

996 \advance\leftmargini6\p@ \itemindent-12\p@
997 \labelwidth\z@ \let\makelabel\descriptionlabel}%
998 }{
999 \endlist
1000 }
1001 \let\enddescription=\endlist % for efficiency

```

`\upn` The command `\upn` can be used to force upright font for punctuation or digits in italic text. For example

... as numbered by `\upn{‘A’}` or `\upn{‘1’}` hereafter`\upn{}`

```

1002 \let\upn=\textup

```

Since these margin settings are dependent on the fonts used, we postpone them until `begin-document`. (This means that we cannot use the values directly for calculations before `begin-document`.) Allow for a reasonable maximum value; `13 = xiii = m` should be adequate.

```

1003 \AtBeginDocument{%
1004 \labelsep=5pt\relax
1005 \setcounter{enumi}{13}\setcounter{enumii}{13}%
1006 \setcounter{enumiii}{13}\setcounter{enumiv}{13}%
1007 \settowidth\leftmargini{\labelenumi\hskip\labelsep}%
1008 \advance\leftmargini by \normalparindent
1009 \settowidth\leftmarginii{\labelenumii\hskip\labelsep}%
1010 \settowidth\leftmarginiii{\labelenumiii\hskip\labelsep}%
1011 \settowidth\leftmarginiv{\labelenumiv\hskip\labelsep}%
1012 \setcounter{enumi}{0}\setcounter{enumii}{0}%
1013 \setcounter{enumiii}{0}\setcounter{enumiv}{0}%
1014 \leftmarginiv=10pt \leftmarginvi=\leftmarginiv
1015 \leftmargin=\leftmargini
1016 \labelwidth=\leftmargini \advance\labelwidth-\labelsep
1017 \@listi}

```

In some contexts the space above/below lists needs to be suppressed. So we put it into a variable `\listisep`.

```

1018 \newskip\listisep

```

```

1019 \listisep\smallskipamount
1020 \def\@listI{\leftmargin\leftmarginI \parsep\z@skip
1021 \topsep\listisep \itemsep\z@skip
1022 \listparindent\normalparindent}
1023 \let\@listI\@listI

```

Is it necessary to reset `\parsep`, `\partopsep`, `\itemsep` to their default values (0) in each of the subordinate list setup functions? I don't believe so, but I leave the settings in the `listii` function just in case some unusual nesting of environments might cause trouble. [mjd,1994/09/22]

```

1024 \def\@listii{\leftmargin\leftmarginii
1025 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1026 \topsep\z@skip \parsep\z@skip \partopsep\z@skip \itemsep\z@skip}
1027 \def\@listiii{\leftmargin\leftmarginiii
1028 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep}
1029 \def\@listiv{\leftmargin\leftmarginiv
1030 \labelwidth\leftmarginiv \advance\labelwidth-\labelsep}
1031 \def\@listv{\leftmargin\leftmarginv
1032 \labelwidth\leftmarginv \advance\labelwidth-\labelsep}
1033 \def\@listvi{\leftmargin\leftmarginvi
1034 \labelwidth\leftmarginvi \advance\labelwidth-\labelsep}

```

### 2.13 Fleqn option

```

1035 \@ifclasswith{\@classname}{fleqn}{%
1036 \let\@tempa\@iden
1037 \AtBeginDocument{\mathindent\leftmarginI}%
1038 }{\let\@tempa\@gobble}%
1039 \@ifpackageloaded{amsmath}{\let\@tempa\@gobble}{%
1040 \@ifpackageloaded{amstex}{\let\@tempa\@gobble}{}}%
1041 }
1042 \@tempa{%
1043 \def\[{ \relax
1044 \ifmmode\@badmath
1045 \else
1046 \begin{trivlist}%
1047 \@beginparpenalty\predisplaypenalty
1048 \@endparpenalty\postdisplaypenalty
1049 \item[]\leavevmode
1050 \hbox to\linewidth\bgroup$\displaystyle

```

Note that the `\m@th` should go at the end in `\]` just in case an embedded small math formula inside `\text` occurs in the display.

Why the extra `\bgroup` here? I think it's not needed. [mjd,3-Feb-1994]

```

1051 \hskip\mathindent\bgroup
1052 \fi}%
1053 \def\]{ \relax
1054 \ifmmode
1055 \egroup \m@th$\hfil \egroup
1056 \end{trivlist}%
1057 \else \@badmath

```

```

1058   \fi}%
1059 \renewenvironment{equation}{%
1060   \@beginparpenalty\predisplaypenalty
1061   \@endparpenalty\postdisplaypenalty
1062   \refstepcounter{equation}%
1063   \@topsep\abovedisplayskip \trivlist
1064   \item[]\leavevmode
1065   \hbox to\linewidth\bgroup\hskip\mathindent$\displaystyle
1066 }{%
1067   \m@th$\hfil \displaywidth\linewidth \hbox{\@eqnnum}\egroup
1068   \endtrivlist
1069 }%
1070 \renewenvironment{eqnarray}{%
1071   \stepcounter{equation}\let\@currentlabel\theequation
1072   \global\@eqnswtrue \global\@eqcnt\z@ \tabskip\mathindent
1073   \let\=\@eqnrcr \abovedisplayskip\topsep
1074   \ifvmode \advance\abovedisplayskip\partopsep \fi
1075   \belowdisplayskip\abovedisplayskip
1076   \belowdisplayshortskip\abovedisplayskip
1077   \abovedisplayshortskip\abovedisplayskip
1078   $$\everycr{}\halign to\linewidth\bgroup
1079     \hskip\@centering
1080     $\displaystyle\tabskip\z@skip###\m@th$&%
1081     \@eqnrel \global\@eqcnt\@ne
1082     \hfil$\{#####\}\m@th$\hfil&%
1083     \global\@eqcnt\tw@
1084     $\displaystyle #####\m@th$\hfil\tabskip\@centering&%
1085     \global\@eqcnt\thr@@
1086     \hbox to \z@\bgroup\hss#####\egroup\tabskip\z@skip\cr
1087 }{%
1088   \@@eqnrcr \egroup \global\advance\c@equation\m@ne$$$%
1089   \global\@ignoretrue
1090 }%
1091 \newdimen\mathindent
1092 \mathindent\leftmargini
1093 }

```

## 2.14 Redefined internal sectioning commands

In `amsart.sty` `\@startsection`, `\@sect`, and a couple of other things are redefined to fix a few hard-coded things that would interfere with the desired style. The most noteworthy difference is that all section headings will go into the table of contents (governed by `secnumdepth` as usual), EVEN WHEN THE \* FORM IS USED. The only section heading not listed in the table of contents is the heading for the toc itself.

A second major departure from standard L<sup>A</sup>T<sub>E</sub>X is that when a short form of a section title is given, it is used only for the running heads; the table of contents still gets the full version of the title. This is correct for AMS editorial practice. However if one wants to get a line break into the table of contents it means that the standard L<sup>A</sup>T<sub>E</sub>X method cannot be used. See the `\except` and

`\for` commands.

```

1094 \def\@startsection#1#2#3#4#5#6{%
Section titles, if they are run-in with the following text, are stored in a box
instead of being typeset right away. They will be typeset by \everypar, but if
one section heading follows right after another, this won't happen. So by doing
\leavevmode we force this to happen. (\if@noskipsec is true if the previous
section title has not yet been typeset.)
1095 \if@noskipsec \leavevmode \fi
1096 \par \@tempskipa #4\relax
1097 \afterindenttrue
1098 \ifdim \@tempskipa <\z@ \@tempskipa -\@tempskipa \afterindentfalse\fi
1099 \if@nobreak \everypar{}\else
1100     \addpenalty\@secpenalty\addvspace\@tempskipa\fi
1101 % Don't call \@ssect in the ifstar case:
1102 \@ifstar{\@dblarg{\@sect{#1}{\@m}{#3}{#4}{#5}{#6}}}%
1103     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%
1104 }
```

We add `\textup` here in order to make section numbers always roman, even if the rest of the section head is italic.

```

1105 \def\@secntformat#1{%
1106     \protect\textup{\protect\@secnumfont
1107         \csname the#1\endcsname
1108     \protect\@secnumfont
1109     }%
1110 }
```

Some journals require a different font for section numbers. (As coded here, this option permits only a change of weight, to `\bfseries`.) The `amsart` font is checked again later, when processing `\section`, and if there is no section title, it is made bold.

```

1111 <amsbook | amsproc>\let\@secnumfont\@empty
1112 <amsart>\def\@secnumfont{\mdseries}
```

For reference, here is the argument list for `\@sect`.

```

% #1 #2 #3 #4 #5 #6 #7 #8
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}[SHORTT]{TITLE}
```

```

1113 \def\@sect#1#2#3#4#5#6[#7]#8{%
```

Define `\@toclevel` for `\@tocwrite` (to `\@tochangmeasure`).

```

1114 \edef\@toclevel{\ifnum#2=\@m 0\else\number#2\fi}%
1115 \ifnum #2>\c@secnumdepth \let\@secnumber\@empty
1116 \else \xp\let\@xp\@secnumber\csname the#1\endcsname\fi
```

If the value of `afterskip`  $> 0$ , then this is not a run-in section heading, and we want to suppress final punctuation.

```

1117 \@tempskipa #5\relax
```

`\@svsec` will be the section number plus some formatting if the star form was not used and if the depth of numbering extends to the current level. The user can change `secnumdepth` to control how many levels of sectioning will be numbered.

```
1118 \ifnum #2>\c@secnumdepth
1119 \let\@svsec\@empty
1120 \else
1121 \refstepcounter{#1}%
```

If the section heading is not run-in and there is no title, omit final punctuation and space. If it is run-in, omit extra space.

```
1122 \edef\@secnumpunct{%
1123 \ifdim\@tempskipa>\z@ % not a run-in section heading
1124 \ifnotempty{#8}{.\@nx\enspace}%
1125 \else
1126 \@ifempty{#8}{.}{.\@nx\enspace}%
1127 \fi
1128 }%
```

For `amsart`, if a subsection has no title, make the section number bold; otherwise leave it alone. [bnb; 2004/06/08; per vwa, B-442]

```
1129 <amsart> \@ifempty{#8}{%
1130 <amsart> \ifnum #2=\tw@ \def\@secnumfont{\bfseries}\fi}{%}
```

If the `*`-form was not used (`#2` less than 1000), we add `\sectionname` or whatever as a prefix, separated by a space. We need the `ifundefined` test in order to know whether the space should be added or not. There must be a better way to do this but I haven't thought of it yet.

```
1131 \protected@edef\@svsec{%
1132 \ifnum#2<\@m
1133 \ifundefined{#1name}{}{%
1134 \ignorespaces\csname #1name\endcsname\space
1135 }%
1136 \fi
1137 \@secntformat{#1}%
1138 }%
1139 \fi
1140 \ifdim \@tempskipa>\z@ % then this is not a run-in section heading
1141 \begingroup #6\relax
1142 \@hangfrom{\hskip #3\relax\@svsec}{\interlinepenalty\@M #8\par}%
1143 \endgroup
```

Section headings don't set marks for the running heads in the article style, only in the `amsbook` style. Assumption: `\sectionmark` is defined to call `\@secnumber` as its penultimate argument.

```
1144 <amsbook> \csname #1mark\endcsname{#7}%
```

If `#2` (level) is greater than 1000 then we don't do a table of contents entry. This happens only for the section heading above the table of contents itself.

```
1145 \ifnum#2>\@m \else \@tocwrite{#1}{#8}\fi
1146 \else
```

Otherwise we're doing a run-in heading; it is stored as `\@svsechd`, which will be typeset by `\everypar` as soon as some text is encountered.

```
1147 \def\@svsechd{#6\hskip #3\@svsec
```

To allow for the possibility that the user wants an empty section title, leaving just the section number, we check whether `#8` is nonempty before adding the period.

```
1148 \ifnotempty{#8}{\ignorespaces#8\unskip
```

The following test is to prevent a period being added if the section title ended in a question mark or other punctuation.

```
1149 \addpunct.}%
1150 \ifnum#2>\@m \else \@tocwrite{#1}{#8}\fi
1151 }%
1152 \fi
```

In a previous version of `amsart` `\@nobreaktrue` was added to `\@xsect` for some reason. Let's keep that just in case it was done to prevent a certain kind of bug. [mjd,17-Aug-1994]

```
1153 \global\@nobreaktrue
```

`\@xsect` does some more stuff based on whether this is a run-in heading or not.

```
1154 \@xsect{#5}}
```

Undefine `\@ssect` to save memory; it's not needed in `amsart`.

```
1155 \let\@ssect\relax
```

## 2.15 Chapters and sections

Allocate counters for sectioning commands. Paragraph and subparagraph counters are allocated but normally not used.

```
1156 \newcounter{part}
1157 \langleamsbook\rangle\newcounter{chapter}
1158 \langleamsproc | amsart\rangle\newcounter{section}
1159 \langleamsbook\rangle\newcounter{section}[chapter]
1160 \langleamsbook\rangle\def\thesection{\arabic{section}}
1161 \newcounter{subsection}[section]
1162 \newcounter{subsubsection}[subsection]
1163 \newcounter{paragraph}[subsubsection]
1164 \newcounter{subparagraph}[paragraph]
```

Set numbering style for sectioning commands. In a couple of cases resetting is unnecessary but we include the full list here for completeness.

```
1165 \renewcommand\thepart {\arabic{part}}
1166 \langleamsbook\rangle\renewcommand\thechapter {\arabic{chapter}}
1167 \renewcommand\thesection {\arabic{section}}
1168 \renewcommand\thesubsection {\thesection.\arabic{subsection}}
1169 \renewcommand\thesubsubsection {\thesubsection.\arabic{subsubsection}}
1170 \renewcommand\theparagraph {\thesubsubsection.\arabic{paragraph}}
1171 \renewcommand\thesubparagraph {\theparagraph.\arabic{subparagraph}}
```

Depth of section numbering; if `secnumdepth` were 2 instead of 3, `\subsubsection` would not be numbered.

```
1172 \setcounter{secnumdepth}{3}
```

The arguments of `\@startsection` are given for reference:

```
% #1    #2    #3    #4    #5    #6
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
```

```
1173 \def\partname{Part}
```

```
1174 <*amsart | amsproc
```

```
1175 \def\part{\@startsection{part}{0}%
```

```
1176   \z@\linespacing\@plus\linespacing}{.5\linespacing}%
```

```
1177   {\normalfont\bfseries\raggedright}}
```

```
1178 </amsart | amsproc
```

`Specialsection` correlates to our inhouse Z-head.

```
1179 \def\specialsection{\@startsection{section}{1}%
```

```
1180 <amsart> \z@\linespacing\@plus\linespacing}{.5\linespacing}%
```

```
1181 <amsproc | amsbook> \z@{2\linespacing\@plus\linespacing}{.5\linespacing}%
```

```
1182 <amsart> {\normalfont\centering}}
```

```
1183 <amsproc | amsbook> {\large\scshape\centering}}
```

In the book class `\part` puts the part title on a separate page.

```
1184 <*amsbook>
```

```
1185 \def\part{\cleardoublepage \thispagestyle{empty}%
```

```
1186   \null\vfil \markboth{}{\secdef\@part\@spart}
```

```
1187 %
```

```
1188 \def\@part[#1]#2{%
```

```
1189   \ifnum \c@secnumdepth >-2\relax \refstepcounter{part}%
```

```
1190   \addcontentsline{toc}{part}{\partname\ thepart.}
```

```
1191   \protect\enspace\protect\noindent#1}%
```

```
1192   \else
```

```
1193     \addcontentsline{toc}{part}{#1}\fi
```

```
1194   \begingroup\centering
```

```
1195   \ifnum \c@secnumdepth >-2\relax
```

```
1196     {\fontsize{\@xviipt}{22}\bfseries
```

```
1197     \partname\ thepart} \vskip 20\p@ \fi
```

```
1198   \fontsize{\@xxpt}{25}\bfseries
```

```
1199     #1\vfil\vfil\endgroup \newpage\thispagestyle{empty}}
```

```
1200
```

```
1201 \def\@spart#1{\addcontentsline{toc}{part}{\protect\noindent#1}%
```

```
1202   \begingroup\centering
```

```
1203   \fontsize{\@xxpt}{25}\bfseries
```

```
1204     #1\vfil\vfil\endgroup \newpage\thispagestyle{empty}}
```

```
1205 </amsbook>
```

The arguments of `\partrunhead` are `\partname`, `\thepart`, and the text of the part title. The first two were fully expanded during the marking process. Use of a mere interword space between the first two args makes it possible to apply `\ignorespaces` and `\unskip` as shown here to produce the desired results if one or the other is empty.

```

1206 ⟨*amsbook⟩
1207 \def\partrunhead#1#2#3{%
1208   \@ifnotempty{#2}{\uppercase{\ignorespaces#1 #2\unskip}\@ifnotempty{#3}{. }}%
1209   \def\@tempa{#3}%
1210   \ifx\@empty\@tempa\else
1211     \begingroup \def\{\ \ignorespaces}% defend against questionable usage
1212     \uppercase\@tempa\@tempa
1213     \endgroup
1214   \fi
1215 }
1216 \let\chapterrunhead\partrunhead
1217 \let\sectionrunhead\partrunhead
1218 ⟨/amsbook⟩

```

Section headings in the amsbook style differ from the amsart style in a couple of ways: The ones that aren't centered are indented on the left, instead of flush left; and the first level, `\section`, is not small caps but bold. Cf. the definition of `\appendix`.

```

1219 \def\section{\@startsection{section}{1}%
1220   \z@{.7\linespacing\@plus\linespacing}{.5\linespacing}%
1221   ⟨amsart⟩ {\normalfont\scshape\centering}}
1222   ⟨amsbook|amsproc⟩ {\normalfont\bfseries\centering}}

```

Negative value for #5 is a signal to make a run-in heading instead of doing a vskip after the heading.

```

1223 \def\subsection{\@startsection{subsection}{2}%
1224   ⟨amsart⟩ \z@{.5\linespacing\@plus.7\linespacing}{-.5em}%
1225   ⟨amsbook|amsproc⟩ \normalparindent{.5\linespacing\@plus.7\linespacing}{-.5em}%
1226   {\normalfont\bfseries}}
1227 \def\subsubsection{\@startsection{subsubsection}{3}%
1228   ⟨amsart⟩ \z@{.5\linespacing\@plus.7\linespacing}{-.5em}%
1229   ⟨amsbook|amsproc⟩ \normalparindent\z@{-.5em}%
1230   {\normalfont\itshape}}

```

Fontdimen 2 of the current font is the ideal interword space of the font. Thus the following spec says that the space after the paragraph heading should be a normal interword space (but nonstretching and nonshrinking).

```

1231 \def\paragraph{\@startsection{paragraph}{4}%
1232   ⟨amsart⟩ \z@\z@{-\fontdimen2\font}%
1233   ⟨amsbook|amsproc⟩ \normalparindent\z@{-\fontdimen2\font}%
1234   \normalfont}
1235 \def\subparagraph{\@startsection{subparagraph}{5}%
1236   \z@\z@{-\fontdimen2\font}%
1237   \normalfont}
1238 ⟨*amsart|amsproc⟩
1239 \def\appendix{\par\c@section\z@ \c@subsection\z@
1240   \let\sectionname\appendixname
1241   \def\thesection{\@Alph\c@section}}
1242 \def\appendixname{Appendix}
1243 ⟨/amsart|amsproc⟩

```

A slower, but fully expandable definition of `\@Roman` to avoid the nonexpandable `\uppercase` which is undesirable in certain circumstances.

```

1244 \def\@Roman#1{\@xp\@slowromancap
1245   \romannumeral#1@}%
1246 %
1247 \def\@slowromancap#1{\ifx @#1% then terminate
1248   \else

```

Note: `\if` is required here, not `\ifx`, because `\romannumeral` returns category 12 letters!

```

1249   \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
1250   c#1C\else\if m#1M\else#1\fi\fi\fi\fi\fi\fi
1251   \@xp\@slowromancap
1252   \fi
1253 }

```

## 2.16 Book features

Books (monographs) comprise three distinct sections, `\frontmatter`, `\mainmatter`, and `\backmatter`. The `\frontmatter` would consist of the title page, copyright page, table of contents, preface, etc. The `\mainmatter` would be the body of the book. The `\backmatter` would include the appendix, bibliography, glossary, and index.

```

1254 <*amsbook>
1255 \def\frontmatter{\cleardoublepage\pagenumbering{roman}}
1256 \def\mainmatter{\cleardoublepage\pagenumbering{arabic}}
1257 \def\backmatter{%
1258   \ifopenright\cleardoublepage\else\clearpage\fi
1259   \let\chaptername\relax}
1260 </amsbook>

```

Book proceedings and monographs allow a signature to print at the end of a preface.

```

1261 <!*amsart>
1262 \def\aufm#1{\par\vspace*{12pt}{\flushright #1\par}}
1263 </!amsart>

```

Monographs can have a special exercise environment.

```

1264 <*amsbook>
1265 \newenvironment{xcb}{%
1266   \setcounter{enumi}{0}%
1267   \settowidth{\leftmargini}{\labelenumi\hskip\labelsep}%
1268   \setcounter{enumii}{4}% letter d
1269   \settowidth{\leftmarginii}{\labelenumii\hskip\labelsep}%
1270   \@startsection{section}% counter name; ignored because of the
1271   % * below
1272   {1}% sectioning level
1273   {\z@}% indent to the left of the section title
1274   {18\p@\@plus2\p@}% vertical space above
1275   {1sp}% Space below of 13pt base-to-base, so none needs to be added
1276   % here; but \z@ would cause the following text to be run-in, so we

```

```

1277     % use 1sp instead.
1278   {\bfseries}% The font of the subsection title
1279   *% always unnumbered
1280 }{%
1281   \par
1282 }
1283 \/\amsbook)

```

## 2.17 Book chapters

The `\chapter` command is provided only in the `amsbook` class, not in `amsart` or `amsproc`.

```

1284 \*amsbook)
1285 \def\chapter{%
1286   \if@openright\cleardoublepage\else\clearpage\fi
1287   \thispagestyle{plain}\global\@topnum\z@
1288   \@afterindenttrue \secdef\@chapter\@schapter}

   \@chapter for numbered chapters.

1289 \def\@chapter[#1]#2{\refstepcounter{chapter}%
1290   \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1291   \else \let\@secnumber\thechapter \fi
1292   \typeout{\chaptername\space\@secnumber}%
1293   \def\@toclevel{0}%
1294   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{#2}%
1295   \else \@tocwriteb\tocchapter{chapter}{#2}\fi
1296   \chaptermark{#1}%
1297   \addtocontents{lof}{\protect\advspace{10\p@}}%
1298   \addtocontents{lot}{\protect\advspace{10\p@}}%
1299   \@makechapterhead{#2}\@afterheading}

   \@schapter for unnumbered chapters.

1300 \def\@schapter#1{\typeout{#1}%
1301   \let\@secnumber\@empty
1302   \def\@toclevel{0}%
1303   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{#1}%
1304   \else \@tocwriteb\tocchapter{chapter}{#1}\fi
1305   \chaptermark{#1}%
1306   \addtocontents{lof}{\protect\advspace{10\p@}}%
1307   \addtocontents{lot}{\protect\advspace{10\p@}}%
1308   \@makeschapterhead{#1}\@afterheading}

1309 \newcommand\chaptername{Chapter}
1310 \newcommand\appendixname{Appendix}

1311 \def\@makechapterhead#1{\global\topskip 7.5pc\relax
1312   \begingroup
1313   \fontsize{\@xivpt}{18}\bfseries\centering

```

In order to keep the chapter number “CHAPTER III” from getting in the way of the `\topskip` we put it inside the paragraph containing the main title. Then we have to do some laborious `\rlap`ing and `\hskip`ing to position it correctly.

```

1314 \ifnum\c@secnumdepth>\m@ne
1315 \leavevmode \hskip-\leftskip
1316 \rlap{\vbox to\z@{\vss
1317 \centerline{\normalsize\mdseries
1318 \uppercase\@xp{\chaptername}\enspace\thechapter}
1319 \vskip 3pc}}\hskip\leftskip\fi
1320 #1\par \endgroup
1321 \skip@34\p@ \advance\skip@-\normalbaselineskip
1322 \vskip\skip@ }

1323 \def\@makeschapterhead#1{\global\topskip 7.5pc\relax
1324 \begingroup
1325 \fontsize{\@xivpt}{18}\bfseries\centering
1326 #1\par \endgroup
1327 \skip@34\p@ \advance\skip@-\normalbaselineskip
1328 \vskip\skip@ }
1329 \langle/amsbook\rangle

```

The `\appendix` command, following the L<sup>A</sup>T<sub>E</sub>X book, marks the start of a division after `\mainmatter` and before `\backmatter` that consists of appendixes.

```

1330 \langle*amsbook\rangle
1331 \def\appendix{\par
1332 \c@chapter\z@ \c@section\z@
1333 \let\chaptername\appendixname
1334 \def\thechapter{\@Alph\c@chapter}}
1335 \langle/amsbook\rangle

```

## 2.18 Table of contents macros

`\tableofcontents` is like `\chapter` or `\section` except for no number and no table of contents entry.

`\@pnumwidth` is the maximum width for page numbers in a table of contents. 1.6em allows enough room for three digits.

```
1336 \newcommand{\@pnumwidth}{1.6em}
```

`\@tocrmarg` is `\@pnumwidth` plus the desired minimum space (1em) between page numbers and the preceding text.

```
1337 \newcommand{\@tocrmarg}{2.6em}
```

```
1338 \langleamsart\rangle\setcounter{tocdepth}{2}
```

```
1339 \langleamsbook | amsproc\rangle\setcounter{tocdepth}{1}
```

```
1340 \newswitch{toc}
```

```
1341 \newswitch{lof}
```

```
1342 \newswitch{lot}
```

Since table of contents, list of figures and list of figures are identical in design as far as the chapter heading and other preliminaries go, we redefine `\@starttoc` to do the necessary work, rather than defining a new macro (which would use up another control sequence name).

We define first the article form of `\@starttoc`, then the book form.

Owing to confusion about what font to use for `\contentsname`, a symbolic name has been assigned to provide flexibility.

```

1343 ⟨*amsart | amsproc⟩
1344 \newcommand\contentsnamefont{\scshape}
1345 \def\@starttoc#1#2{\begingroup
1346   \setTrue{#1}%

```

Remove the skip after the abstract so that we can substitute another.

```
1347 \par\removeLastskip\vskip\z@skip
```

The first two arguments of `\@startsection` here are special values that cause different internal branches to be taken.

```

Arguments: {} = name = empty
\@M = no number should be used and no table of contents entry
\z@ = indent amount
12pt + 12pt = vskip before
6pt = vskip after
\centering\contentsnamefont = format
1348 \@startsection{}\@M\z@{\linespacing\@plus\linespacing}%
1349   {.5\linespacing}\centering\contentsnamefont}{#2}%

```

If we have a list of figures or list of tables we want to put them in the main table of contents, but we don't want to put an entry there for the main table of contents itself. So we check to see if argument 2 is `\contentsname` and if it is then we refrain from doing `\addcontentsline`.

```

1350 \ifx\contentsname#2%
1351 \else \addcontentsline{toc}{section}{#2}\fi
1352 \makeatletter
1353 \@input{\jobname.#1}%
1354 \if@filesw
1355   \xp\newwrite\csname tf@#1\endcsname
1356   \immediate\xp\openout\csname tf@#1\endcsname \jobname.#1\relax
1357 \fi
1358 \global\@nobeakfalse \endgroup
1359 \addvspace{32\p@\@plus14\p@}%
1360 \let\tableofcontents\relax
1361 }
1362 ⟨/amsart | amsproc⟩

```

And here is the book form of `\@starttoc`.

```

1363 ⟨*amsbook⟩
1364 \def\@starttoc#1#2{%
1365   \begingroup
1366   \setTrue{#1}%

```

Inside this group we change `\secdef` so that we can call `\chapter` and only get the preliminary part of its definition that we need.

```
1367 \let\secdef\@gobbletwo \chapter
```

If we have a list of figures or list of tables we want to put them in the main table of contents, but we don't want to put an entry there for the main table of contents itself. So we check to see if argument 2 is `\contentsname` and if it is then we refrain from doing `\addcontentsline`.

```

1368 \let\@secnumber\@empty % for \@tocwrite and \chaptermark
1369 \ifx\contentsname#2%
1370 \else \@tocwrite{chapter}{#2}\fi

```

Now we do the equivalent of `\schapter`. Expand #2 so that it will be easy to apply uppercasing to it. (For `\starttoc` we assume that #2 is always a control such as `\contentsname`.)

```

1371 \typeout{#2}\xp\chaptermark\@xp{#2}%
1372 \@makeschapterhead{#2}\@afterheading

```

Protect against document classes that have nonzero `\parskip`.

```

1373 \parskip\z@skip

```

And finally we read in the `.toc` (or whatever) file.

```

1374 \makeatletter
1375 \@input{\jobname.#1}%
1376 \if@filesw
1377   \xp\newwrite\csname tf@#1\endcsname
1378   \immediate\xp\openout\csname tf@#1\endcsname \jobname.#1\relax
1379 \fi
1380 \global\@nobeakfalse \endgroup
1381 \newpage
1382 }
1383 </amsbook>

```

Now it is easy to define `\tableofcontents` and its relatives.

```

1384 \def\contentsname{Contents}
1385 \def\listfigurename{List of Figures}
1386 \def\listtablename{List of Tables}
1387 \def\tableofcontents{%
1388   \@starttoc{toc}\contentsname
1389 }
1390 \def\listoffigures{\@starttoc{lof}\listfigurename}
1391 \def\listoftables{\@starttoc{lot}\listtablename}

```

In order to automatically leave enough space for the ‘number’ part of toc entries, we compute the maximum width of the ‘number’ part for each sectioning level and pass that information to `\tocline` through the `.aux` file.

Init the `tocindent`s if they are not yet set (first run).

```

1392 \AtBeginDocument{%
1393   \for\@tempa:=-1,0,1,2,3\do{%
1394     \ifundefined{r@tocindent\@tempa}{%
1395       \xp\gdef\csname r@tocindent\@tempa\endcsname{0pt}}{}}%
1396   }%
1397 }

```

`\writetocindent` This function writes out the max toc indents to the aux file.

```

1398 \def\writetocindent{%
1399   \begingroup
1400   \for\@tempa:=-1,0,1,2,3\do{%
1401     \immediate\write\@auxout{%

```

```

1402     \string\newlabel{tocindent\@tempa}{%
1403         \csname r@tocindent\@tempa\endcsname}}%
1404     }%
1405     \endgroup}
1406 %
1407 \AtEndDocument{\@writetocindents}

```

`\indentlabel` This function is a no-op except in `\@tocwrite` where it is a pointer to `\@tochangmeasure`.

```
1408 \let\indentlabel\@empty
```

`\@tochangmeasure` This function measures the hangindent part of a toc entry and updates the current max for the given sectioning level, if necessary. The max's at the end of the document will be written in the form of a pseudo-label to the `.aux` file by `\@writetocindents`.

We can assume that `\@tochangmeasure` is already inside a group when called.

```

1409 \def\@tochangmeasure#1{\sbox\z@{#1}%
1410     \ifdim\wd\z@>\csname r@tocindent\@toclevel\endcsname\relax
1411         \exp\xdef\csname r@tocindent\@toclevel\endcsname{\the\wd\z@}%
1412     \fi
1413 }

```

`\@toclevel` Initialize, for the record.

```
1414 \def\@toclevel{0}
```

Since we don't have leader dots, we have `\@tocline` instead of `\@dottedtocline`.

```

1415 \def\@tocline#1#2#3#4#5#6#7{\relax
1416     \ifnum #1>\c@tocdepth % then omit
1417     \else
1418         \par \addpenalty\@secpenalty\addvspace{#2}%
1419         \begingroup \hyphenpenalty\@M
1420         \@ifempty{#4}{%
1421             \@tempdima\csname r@tocindent\@toclevel\endcsname\relax
1422         }{%
1423             \@tempdima#4\relax
1424         }%
1425         \parindent\z@ \leftskip#3\relax \advance\leftskip\@tempdima\relax
1426         \rightskip\@pnumwidth plus4em \parfillskip-\@pnumwidth
1427         #5\leavevmode\hskip-\@tempdima #6\nobreak\relax
1428         \hfil\hbox to\@pnumwidth{\@tocpagenum{#7}}\par
1429         \nobreak
1430         \endgroup
1431     \fi}
1432 \def\@tocpagenum#1{\hss{\mdseries #1}}

```

The function `\@tocwrite` writes the information of a section heading to the `.toc` file in a standard form. It allows different functions to be substituted for `\numberline`, to get greater control of toc formatting.

```

1433 \def\@tocwrite#1{\@xp\@tocwriteb\csname toc#1\endcsname{#1}}
    The \chapter command uses \@tocwriteb directly because of the need to
    write slightly different things to the toc file depending on the current value
    of \chaptername.
1434 \def\@tocwriteb#1#2#3{%
1435   \begingroup
1436     \def\@tocline##1##2##3##4##5##6{%
1437       \ifnum##1>\c@tocdepth
1438         \else \sbox\z@{##5\let\indentlabel\@tochangmeasure##6\fi}%
1439         \csname l@#2\endcsname{#1{\csname#2name\endcsname}{\@secnumber}{}}%
1440       \endgroup
1441       \addcontentsline{toc}{#2}%
1442       {\protect#1{\csname#2name\endcsname}{\@secnumber}{#3}}

```

Specs for monograph toc are as follows (tocdepth is 1, i.e., subsections and lower are not listed in toc).

Part: Space above 12pt plus2pt, indent 0pt, "Part" + wordspace + number + "." + 1em + title (raggedright, no hangindent) + 1em + page number in column 1.6em wide.

Chapter: Space above 8pt, hangindent on ("Chapter 0" + "." + 1em), + title (raggedright) + 1em + page number in column 1.6em wide.

Appendix: Same as Chapter except for epithet "Appendix M"

Section: Space above 0pt, hangindent on (1pc + "0.0" + "." + 1em), + title (raggedright) + 1em + page number in column 1.6em wide.

Specs for article toc are as follows (tocdepth 2):

Section: Same as for monograph.

Subsection: Space above 0pt, hangindent 6pc (number + "." + 1em), + title (raggedright) + 1em + page number in column 1.6em wide.

Subsubsection: Same as subsection but hangindent 8pc

Typical invocation of \l@chapter:

```

\contentsline{chapter}{%
  \tocchapter{Chapter}{3}{Some title stuff}}{103}
-->\l@chapter-->
      #1 #2  #3  #4 #5
\@tocline{0}{8pt}{0pt}{\bfseries}
      #6      6a      6b 6c      #7
  {\tocchapter{Chapter}{3}{Some title stuff}}{103}

```

The \tocchapter is a slightly more useful form than \numberline that allows control for optionally omitting strings like 'Chapter' or changing fonts for sub-components of the toc entry. Note that it is allowed to have appendix and chapter at the same toc level, with \tocappendix instead of \tocchapter written in the .toc file.

The arguments of `\@tocline` are as follows:

```
\@tocline{LEVEL}{VSPACE}{INDENT}{NUMBERWIDTH}{EXTRA}%
{TEXT}{PAGENUM}
```

where ‘numberwidth’ is the width of the box allotted to contain the section number, including any preceding word like ‘Chapter’ or ‘Part’. If this width arg is empty then an automatically computed width (max over TOC of the numberwidths for this level) is used. The ‘extra’ argument is formatting such as font changes. The ‘text’ argument contains a section-command specific function like `\tocsection` or `\tocchapter` which takes in turn three arguments: epithet, number, topic.

```
1443 \def\l@section{\@tocline{1}{0pt}{1pc}{}}
    *** The indents do not agree between in-house and distributed ***
    ** versions; no changes; get specs before revising. ***
    The use of \ignorespaces in \tocsection and its relatives means that if #1 is
    empty, the following space will be also removed.
1444 \newcommand{\tocsection}[3]{%
1445   \indentlabel{\@ifnotempty{#2}{\ignorespaces#1 #2.\quad}}#3}
1446 \def\l@subsection{\@tocline{2}{0pt}{1pc}{5pc}}
1447 \let\tocsubsection\tocsection
1448 \def\l@subsubsection{\@tocline{3}{0pt}{1pc}{7pc}}
1449 \let\tocsubsubsection\tocsection
1450 \let\l@paragraph\l@subsubsection
1451 \let\tocparagraph\tocsection
1452 \let\l@subparagraph\l@subsubsection
1453 \let\tocsubparagraph\tocsection
1454 %
1455 \def\l@part{\@tocline{-1}{12pt plus2pt}{0pt}}{\bfseries}
1456 \let\tocpart\tocsection
1457 \def\l@chapter{\@tocline{0}{8pt plus1pt}{0pt}}{}
1458 \let\tocchapter\tocsection
```

In this case we are pretty sure the word ”Appendix” or similar is present, so only check if arg 2 is empty:

```
1459 \newcommand{\tocappendix}[3]{%
1460   \indentlabel{#1\@ifnotempty{#2}{ #2}.\quad}#3}
1461 \def\l@figure{\@tocline{0}{3pt plus2pt}{0pt}{1.5pc}}
1462 \let\l@table=\l@figure
```

## 2.19 Bibliography section or chapter

Following the `babel` package, we use `\refname` in articles and `\bibname` in books.

```
1463 \def\refname{References}
1464 \def\bibname{Bibliography}
```

Restudy the following code; `\bibsetup` isn’t used anywhere although it’s defined for the three different `bibstyles`. Because some publications have different default label styles, separate that out for easy tailoring of packages.

```

1465 \def\@defaultbiblabelstyle#1{#1.}
1466 \def\bibliographystyle#1{%
1467   \if@files\immediate\write\@auxout{\string\bibstyle{#1}}\fi
1468   \def\@tempa{#1}%
1469   \def\@tempb{amsplain}%
1470   \def\@tempc{}%
1471   \ifx\@tempa\@tempb
1472     \def\@biblabel##1{\@defaultbiblabelstyle{##1}}%
1473     \def\bibsetup{}%
1474   \else
1475     \def\bibsetup{\labelsep6\p@}%
1476     \ifx\@tempa\@tempc
1477       \def\@biblabel##1{}%
1478       \def\bibsetup{\labelwidth\z@ \leftmargin24\p@
1479         \itemindent-\leftmargin
1480         \labelsep\z@ }%
1481     \fi
1482   \fi}

    Permit easy change of font size for unusual purpose, e.g., for an author's
    "life list" in collected works. [bnb, 2004/04/01]
1483 \newcommand{\bibliofont}{\footnotesize}

    thebibliography differs in some author packages only in the shape of the
    title; make this easy to change. [bnb, 2004/05/22]
1484 \newcommand{\@bibtitlestyle}{%
1485 <amsart|amsproc> \@xp\section\@xp*\@xp{\refname}%
1486 <amsbook> \@xp\chapter\@xp*\@xp{\bibname}%
1487 }
1488 \newenvironment{thebibliography}[1]{%
1489   \@bibtitlestyle
1490   \normalfont\bibliofont\labelsep .5em\relax
1491   \renewcommand\theenumiv{\arabic{enumiv}}\let\p@enumiv\@empty
1492   \list{\@biblabel{\theenumiv}}{\settowidth\labelwidth{\@biblabel{#1}}%
1493     \leftmargin\labelwidth \advance\leftmargin\labelsep
1494     \usecounter{enumiv}}%
1495   \sloppy \clubpenalty\@M \widowpenalty\clubpenalty
1496   \sfcode'\.=\@m
1497 }{%

    Change error for empty list (no items) to warning, to allow authors to leave
    their bibliography temporarily empty during writing:
1498   \def\@noitemerr{\@latex@warning{Empty 'thebibliography' environment}}%
1499   \endlist
1500 }

    The \bysame command prints a horizontal dash indicating repetition of the
    author's name in consecutive bibliography entries.
1501 \def\bysame{\leavevmode\hbox to3em{\hrulefill}\thinspace}

    We define \newblock even though it's not needed for AMS publication style, just
    to avoid error messages when a non-AMS .bst file is used. This is a convenience

```

for users; use of `\newblock` is not recommended for submissions to the AMS.

```
1502 \def\newblock{}
```

`\MR` Provide an MR number for a bibliography item. At the moment [mjd,1995/08/07] `\MRhref` this only prints the MR number, but later we expect to extend it to write an HTML `\special` to the `.dvi` file.

The presentation of the MR number has been simplified (from using a bold volume number) coincident with the change in the MathSciNet database to a 7-digit reference number from the volume:number form.

Ensure that an old-style MR number does not break across lines if it contains a space; editorial request. [bnb; 2004/04/01] Countermanded, to permit break between reference number and an old-style number following in parentheses. [bnb; 2004/06/11]

```
1503 \newcommand\MR[1]{\relax\ifhmode\unskip\spacefactor3000 \space\fi
```

```
1504 MR~\MRhref{#1}{#1}}
```

```
1505 \let\MRhref@gobble
```

`\URL` Allows sticking in an arbitrary URL in a bibliography. Leading “http” is not `\URLhref` assumed. Call `\verb` to ensure that special characters in the URL don’t cause `\@URL` trouble.

```
1506 \newcommand\URL{\begingroup
```

```
1507 \def\@sverb##1{%
```

```
1508 \def\@tempa####1##1{\@URL{####1}\egroup\endgroup}%
```

```
1509 \@tempa}%
```

```
1510 \verb}
```

```
1511 \let\URLhref@gobble
```

```
1512 \def\@URL#1{\URLhref{#1}#1}
```

## 2.20 Index section or chapter

Chapter or section heading for an index. Index is set up to be two columns.

```
1513 \newif\if@restonecol
```

```
1514 \amsbook
```

```
1515 \def\indexchap#1{\global\topskip 7.5pc\relax
```

```
1516 \twocolumn[{\fontsize{\xivpt}{18}\bfseries\centering
```

```
1517 \vskip\topskip\hbox{ }\vskip-\baselineskip% adjust top space
```

```
1518 #1\par
```

After `\twocolumn` finishes operating, the top material is left in an insert register, and `\topskip` will now be applied above the following material. So we should set it to the normal after-chapter-title space (34pt)—cf. `\@makeschapterhead`.

```
1519 \global\topskip 34\p@\relax
```

```
1520 \ifx\@empty\indexintro
```

```
1521 \else
```

```
1522 \begingroup \normalsize
```

```
1523 \skip@\topskip \advance\skip@ -\baselineskip
```

```
1524 \vskip\skip@
```

```

1525     \parbox[t]{24pc}{\normalfont\indexintro\par}%
1526     \endgroup
1527     \global\topskip 24\p@\relax
1528     \fi
1529   }]%
1530 }
1531 \newcommand{\indexintro}{}
1532 </amsbook>

    theindex differs in some author packages only in the shape of the title; make
    this easy to change. [bnb, 2004/05/22]
1533 \newcommand{\@indextitlestyle}{%
1534 <*amsbook>
1535   \let\@makeschapterhead\indexchap
1536   \@xp\chapter\@xp*\@xp{\indexname}%
1537 </amsbook>
1538 <amsart|amsproc> \twocolumn[\@xp\section\@xp*\@xp{\indexname}]%
1539 }
1540 \def\theindex{\@restonecoltrue\if@twocolumn\@restonecolfalse\fi
1541   \columnseprule\z@ \columnsep 35\p@
1542   \@indextitlestyle
1543   \thispagestyle{plain}%
1544   \let\item\@idxitem
1545   \parindent\z@ \parskip\z@\@plus.3\p@\relax
1546   \raggedright
1547   \hyphenpenalty\@M
1548   \footnotesize}
1549 \def\indexname{Index}

1550 \def\@idxitem{\par\hangindent 2em}
1551 \def\@subitem{\par\hangindent 2em\hspace*{1em}}
1552 \def\@subsubitem{\par\hangindent 3em\hspace*{2em}}
1553 \def\endtheindex{\if@restonecol\onecolumn\else\clearpage\fi}
1554 \def\indexspace{\par\bigskip}

```

## 2.21 Footnotes

In books the footnote counter should reset to 0 at the beginning of each chapter:

```
1555 <amsbook>\@addtoreset{footnote}{chapter}
```

Rule above footnotes is 5 picas wide.

```

1556 \def\footnoterule{\kern-.4\p@
1557     \hrule\@width 5pc\kern11\p@\kern-\footnotesep}

```

A simple superscript doesn't work here; it fails on a minipage, where `\itshape` (which is invalid in math mode) is used for the footnote numbers. Cf. the definition of `\textprime`.

```

1558 \def\@makefnmark{%
1559   \leavevmode
1560   \raise.9ex\hbox{\fontsize\sf@size\z@\normalfont\@thefnmark}%
1561 }
1562 %

```

```
1563 \def\@makefntext{\indent\@makefnmark}
```

Add `\normalfont` before `\footnotesize` so that fonts will come out properly using the new font selection scheme.

```
1564 \long\def\@footnotetext#1{%
1565   \insert\footins{%
1566     \normalfont\footnotesize
1567     \interlinepenalty\interfootnotelinepenalty
1568     \splittopskip\footnotesep \splitmaxdepth \dp\strutbox
1569     \floatingpenalty\@MM \hsize\columnwidth
```

Mostly `\@parboxrestore` does what we want; but not with respect to `\parindent` and `\tolerance`.

```
1570   \@parboxrestore \parindent\normalparindent \sloppy
1571   \protected@edef\@currentlabel{%
1572     \csname p@footnote\endcsname\@thefnmark}%
1573   \@makefntext{%
1574     \rule\z@\footnotesep\ignorespaces#1\unskip\strut\par}}}
```

We change `\sloppy` to keep it from overriding our normal value of 1pt for `\hfuzz` and `\vfuzz` with a LESS sloppy value (.5pt).

```
1575 \hfuzz=1pt \vfuzz=\hfuzz
1576 \def\sloppy{\tolerance9999 \emergencystretch 3em\relax}
```

## 2.22 Float placement parameters

These control the placing of floating objects like tables and figures. We use much more tolerant values than the L<sup>A</sup>T<sub>E</sub>X defaults; the L<sup>A</sup>T<sub>E</sub>X defaults are geared to fussier page breaks, at a price of requiring more manual intervention to deal with difficult page breaking problems.

When using L<sup>A</sup>T<sub>E</sub>X's twocolumn option, 'page' really means 'column', for the parameters that don't have a dbl prefix: that is, `topnumber` is then the maximum number of top figures allowed in each column, and so forth.

**topnumber** maximum number of top figures allowed per page

**bottomnumber** maximum number of bottom figures allowed per page

**totalnumber** maximum number of figures (top and bottom) allowed per page

**dbltopnumber** same as `topnumber`, but for two-column wide figures, when double-column format is used

```
1577 \setcounter{topnumber}{4}
1578 \setcounter{bottomnumber}{4}
1579 \setcounter{totalnumber}{4}
1580 \setcounter{dbltopnumber}{4}
```

Float fraction parameters.

`\topfraction` maximum part of the page allowed for top figures, expressed as a decimal fraction. The value of .97 means roughly 'accept pages that have only two lines of text, and the rest figures'.

`\bottomfraction` same as `\topfraction`, but for bottom figures

`\textfraction` *minimum* part of the page that must be occupied by text, if the page is to have any text at all. If this value cannot be achieved, L<sup>A</sup>T<sub>E</sub>X will turn the current figure or figures into a “float page”, i.e., a page of figures without any text.

`\floatpagefraction` minimum amount (that is, total combined height) of figures needed before L<sup>A</sup>T<sub>E</sub>X will make a float page. This is expressed as a fraction of the normal page height.

`\dbltopfraction` like `\topfraction`, but applies only to figures that are two columns wide, when double-column format is used.

`\dblfloatpagefraction` minimum amount of double-column figure material needed before L<sup>A</sup>T<sub>E</sub>X will make a two-column wide “float page”

```
1581 \renewcommand{\topfraction}{.97}
1582 \renewcommand{\bottomfraction}{.97}
1583 \renewcommand{\textfraction}{.03}
1584 \renewcommand{\floatpagefraction}{.9}
1585 \renewcommand{\dbltopfraction}{.97}
1586 \renewcommand{\dblfloatpagefraction}{.9}
```

We also modify the default values for spacing around floating figures: (A) so that figures on a float page will not be vertically centered on the total page height but will be flush at the top of the page, and (B) so that there will be slightly more stretchability around figures, to help find better page breaks in difficult situations.

`\floatsep` Space between consecutive figures

`\textfloatsep` Space between text and top or bottom figures

`\intextsep` Space above and below a figure in the middle of the text (i.e., placed with the `[h]` option)

`\dblfloatsep` Space between consecutive figures that are two columns wide (when two-column format is used)

`\dbltextfloatsep` Space between double-column figures and text

`@fptop` Space above the first figure on a float page

`@fpsep` Space between figures on a float page

`@fpbot` Space below the last figure on a float page

`@dblfpptop` Space above the first double-column figure on a two-column wide float page

`@dblfpptop` Space between double-column figures on a two-column wide float page

`@dblfpbot` Space below the last double-column figure on a two-column wide float page

```
1587 \setlength{\floatsep}{12pt plus 6pt minus 4pt}
1588 \setlength{\textfloatsep}{15pt plus 8pt minus 5pt}
1589 \setlength{\intextsep}{12pt plus 6pt minus 4pt}
1590 \setlength{\dblfloatsep}{12pt plus 6pt minus 4pt}
1591 \setlength{\dbltextfloatsep}{15pt plus 8pt minus 5pt}
```

```

1592 \setlength{\@fptop}{0pt}% removed "plus 1fil"
1593 \setlength{\@fpsep}{8pt}% removed "plus 2fil"
1594 \setlength{\@fpbot}{0pt plus 1fil}
1595 \setlength{\@dblftop}{0pt}% removed "plus 1fil"
1596 \setlength{\@dblfpsep}{8pt}% removed "plus 2fil"
1597 \setlength{\@dblfpbot}{0pt plus 1fil}

```

`\fps@figure`, `\fps@table`: placement specifications for `figure` and `table` environments. ‘`tbp`’ means that a figure will be placed at the top or bottom of a page, or on a separate page with no text. This might be changed to ‘`tp`’, for example, if you never want figures to appear at the bottom of a page.

```

1598 \newcommand{\fps@figure}{tbp}
1599 \newcommand{\fps@table}{tbp}

```

Some more setup for figures.

```

1600 <amsart | amsproc>\newcounter{figure}
1601 <amsbook>\newcounter{figure}[chapter]
1602 \def\@captionheadfont{\scshape}
1603 \def\@captionfont{\normalfont}
1604 \def\ftype@figure{1}
1605 \def\ext@figure{lof}
1606 \def\fnm@figure{\figurename\ \thefigure}
1607 \def\figurename{Figure}
1608 \newenvironment{figure}{%
1609   \@float{figure}%
1610 }{%
1611   \end@float
1612 }
1613 \newenvironment{figure*}{%
1614   \@dblfloat{figure}%
1615 }{%
1616   \end@dblfloat
1617 }

```

And similar for tables.

```

1618 <amsart | amsproc>\newcounter{table}
1619 <amsbook>\newcounter{table}[chapter]
1620 \def\ftype@table{2}
1621 \def\ext@table{lot}
1622 \def\fnm@table{\tablename\ \thetable}
1623 \def\tablename{Table}
1624 \newenvironment{table}{%
1625   \@float{table}%
1626 }{%
1627   \end@float
1628 }
1629 \newenvironment{table*}{%
1630   \@dblfloat{table}%
1631 }{%
1632   \end@dblfloat
1633 }

```

Change `\@floatboxreset` to add `\centering`. Centering is always applied to tables and figures in AMS publications. It should not be necessary to throw in `\begin{center} ... \end{center}` in every instance to achieve this.

```
1634 \def\@floatboxreset{\global\@minipagefalse \centering}
```

This is what we want `\@makecaption` to do: If the total width is greater than normal `columnwidth` we want to break the caption into lines using a line width of  $W = (\text{columnwidth} - 6\text{pc})$ , and center the resulting block between the margins. Otherwise we want to set the caption as a single line, centered between the margins.

To do this we set the caption as a vbox with line width  $W$ , except that we allow the last line (which may be the only line) to have width up to full `columnwidth` by adding a kern of  $-6\text{pc}$ . If the result is a single hbox (i.e., a single line) we need to unpack the hbox, remove `rightskip`, `parfillskip`, and the  $-6\text{pc}$  kern, and center the remaining material. If the caption is more than one line, then box 1 contains the last line, which we need to unpack in the same way, and run through the paragraphing process again (because this last line may be up to 6 picas wider than the desired width).

In practice this procedure tends to fail if there are any potential breakpoints near the end of the first line (in the window between short-width and full-width). Then  $\text{T}_{\text{E}}\text{X}$  tends to choose a break (depending on `spaceskip`, `tolerance`, etc) at the last acceptable breakpoint before short-width is exceeded, *without considering any later material*—in particular, the negative kern. [This was pointed out by Donald Arseneau, May 2000.] Unfortunately, setting `parfillskip` to a negative value does not work either. I guess this is a special case of the existing limitation on `parshape`: you cannot specify a `parshape` in terms of number of lines from the bottom of the paragraph. (We would like to specify a `parshape` where the last line is 6 picas longer than the others.)

Finally, if the caption is for a figure, it will be set below the figure, so the separating space goes above the caption; otherwise the separating space goes below the caption.

```
1635 \long\def\@makecaption#1#2{%
```

Measure the contents of the caption. If `#2` is not empty, then we must add a period and an en-space before typesetting it. The `\@caption` macro adds an extra `\ignorespaces` at the beginning of `#2`, so in order to find out if the user-typed portion is empty we use `\@cdr` to pull off the `\ignorespaces`.

```
1636 %    Use color-safe commands
1637 \setbox\@tempboxa\vbox{\color@setgroup
1638 \advance\hsize-2\captionindent\noindent
1639 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
1640 {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
1641 \unskip\kern-2\captionindent\par
1642 \global\setbox\@ne\lastbox\color@endgroup}%
1643 \ifhbox\@ne % the normal case
1644 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
1645 \fi
```

If `\@tempboxa` is not empty at this point then the caption was more than one line long or there was extra vertical mode material, maybe a `\write` (from `\index` or something). Interestingly, we can't use `\ifvoid` to see if `\@tempboxa` is empty, because empty is not the same thing as void (as far as the `\ifvoid` test is concerned). So instead we measure the width of `\@tempboxa` to see if it's zero; this should suffice for non-bizarre cases.

```

1646 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
1647   \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
1648   \else % tempboxa contained more than one line
1649     \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
1650       \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
1651   \fi
1652 \ifnum\@tempcnta<64 % if the float IS a figure...
1653   \addvspace\abovecaptionskip
1654   \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
1655 \else % if the float IS NOT a figure...
1656   \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
1657   \nobreak
1658   \vskip\belowcaptionskip
1659 \fi
1660 \relax
1661 }
```

Allocate the skip registers for above and below caption space.

```

1662 \newskip\abovecaptionskip \abovecaptionskip=12pt \relax
1663 \newskip\belowcaptionskip \belowcaptionskip=12pt \relax
1664 \newdimen\captionindent \captionindent=3pc
```

## 2.23 Miscellaneous

`\nonbreakingspace` Change `~` to be more forgiving of accidental adjacent spaces. Note that this means multiple `~~~...` cannot be used to get multiple spaces in the output.

```

1665 \def\nonbreakingspace{\unskip\nobreak\ \ignorespaces}
1666 \def~{\protect\nonbreakingspace}
```

Redefine `\@biblabel` to do nothing if the argument is empty. We don't really care what the previous definition was so we don't check it.

```

1667 \def\@biblabel#1{\ifnotempty{#1}{[#1]}}
```

Changed `\@cite` to always use roman/upright, nonbold font, even in italic or bold text (following AMS style). Turn off `\mathsurround` just in case there are subscripts in the cite numbers.

```

1668 \def\@citestyle{\m@th\upshape\mdseries}
1669 \langleamsart\rangle\let\citeform\@firstofone
1670 \langleamsbook|amsproc\rangle\def\citeform#1{\bfseries#1}
1671 \def\@cite#1#2{\%
1672   \@citestyle[\citeform{#1}\if@tempswa, #2\fi]}
```

Make `\cite` robust if it isn't already. Too many unsuspecting users get problems from this in a figure or table caption.

```

1673 \@ifundefined{cite }{%
1674   \expandafter\let\csname cite \endcsname\cite
1675   \edef\cite{\@nx\protect\@xp\@nx\csname cite \endcsname}%
1676 }{}

```

`\fullwidthdisplay` The function `\fullwidthdisplay` makes a displayed equation take up the full column width even if the current context is an indented list.

```

1677 \def\fullwidthdisplay{\displayindent\z@ \displaywidth\columnwidth}

```

And we insert the `\fullwidthdisplay` function at the beginning of `\everydisplay` just in case any later code in `\everydisplay` needs to use the values of `\displayindent` or `\displaywidth`.

```

1678 \edef\@tempa{\noexpand\fullwidthdisplay\the\everydisplay}
1679 \everydisplay\expandafter{\@tempa}

```

A few odds and ends for indexes, based on `makeindex`. The definition of `\see` as “see also” is unfortunate, but of long standing, and cannot be changed without destroying backward compatibility, so an alternate command, `\seeonly`, is provided to cover the basic situation.

```

1680 \newcommand*\seeonlyname{see}
1681 \newcommand*\seename{see also}
1682 \newcommand*\alsoname{see also}
1683 \newcommand*\seeonly[2]{\emph{\seeonlyname} #1}
1684 \newcommand*\see[2]{\emph{\seename} #1}
1685 \newcommand*\seealso[2]{\emph{\alsoname} #1}
1686 \newcommand\printindex{\@input{\jobname.ind}}

```

`\textprime` A text prime symbol, for applying primes to numbers such as list numbers or equation numbers that are not really math. Furthermore Cyrillic myagkii znak, or soft sign, is represented by a prime symbol in Russian names when they are transliterated into English.

`\textprime` uses the prime symbol from `math`, but because it’s intended specifically for nonmath use, we avoid going through math mode with `$. . . $`. We must therefore call `\check@mathfonts` to ensure that `scriptfont2` is actually defined. Otherwise, if `\textprime` were used in a document before the first math formula, there would be no adequate assignment yet for `\scriptfont2`.

The raise value of `.8ex` is just a reasonable guess at making the bottom of the prime symbol fall near the top of a preceding lowercase letter but still not fall too low on an uppercase letter. We could look up the `\fontdimen` values used in math mode for superscripts but I don’t think it’s worth the bother.

Cf. also the definition of `\@makefnmark`. The prime symbol here is not raised quite so high because I think that is desirable for the soft-sign usage.

```

1687 \DeclareRobustCommand\textprime{\leavevmode
1688   \raise.8ex\hbox{\check@mathfonts\the\scriptfont2 \char48 }}

```

## 2.24 Book style variations

Here is the layout for a `\maketitle` in the `amsbook` class.

```

1689 ⟨*amsbook⟩
1690 \def\maketitle{\par
1691   \@topnum\z@ % this prevents figures from falling at the top of page 1
1692   \begingroup
1693   \@maketitle
1694   \endgroup
1695   \c@footnote\z@
1696   \def\do##1{\let##1\relax}%
1697   \do\maketitle \do\@maketitle \do\title \do\@xtitle \do\@title
1698   \do\author \do\@xauthor \do\address \do\@xaddress
1699   \do\email \do\@xemail \do\curraddr \do\@xcurraddr
1700   \do\dedicatory \do\@dedicatory \do\thanks \do\thankses
1701   \do\keywords \do\@keywords \do\subjclass \do\@subjclass
1702 }

1703 \def\@maketitle{%
1704   \cleardoublepage \thispagestyle{empty}%
1705   \begingroup \topskip\z@skip
1706   \null\vfil
1707   \begingroup
1708   \LARGE\bfseries \centering
1709   \openup\medskipamount
1710   \@title\par\vspace{24pt}%
1711   \def\and{\par\medskip}\centering
1712   \mdseries\authors\par\bigskip
1713   \endgroup
1714   \vfil
1715   \ifx\@empty\addresses \else \@setaddresses \fi
1716   \vfil
1717   \ifx\@empty\@dedicatory
1718   \else \begingroup
1719     \centering{\footnotesize\itshape\@dedicatory\@par}%
1720     \endgroup
1721   \fi
1722   \vfill
1723   \newpage\thispagestyle{empty}
1724   \begin{center}
1725     \ifx\@empty\@subjclass\else\@setsubjclass\fi
1726     \ifx\@empty\@keywords\else\@setkeywords\fi
1727     \ifx\@empty\@translators\else\vfil\@settranslators\fi
1728     \ifx\@empty\thankses\else\vfil\@setthanks\fi
1729   \end{center}
1730   \vfil
1731   \@setabstract
1732   \endgroup}

Define the desired form for translator names.

1733 \def\@settranslators{\par
1734   \begingroup
1735   \translname: \andify\@translators \uppercase\@translators
1736   \@translators \@par

```

```

1737 \endgroup}
1738 \def\@setdate{\par\smallskip\@date\par\smallskip}
1739 \def\@setsubjclass{\par\smallskip
1740   {\itshape\subjclassname.}\enspace\@subjclass\par\smallskip}
1741 \def\@setkeywords{\par\smallskip
1742   {\itshape \keywordsname.}\enspace \@keywords\par\smallskip}
1743 \</amsbook>
1744 \</classes>

```

## 2.25 Setup for theorems, definitions, remarks, proofs

### Intended usage

Here are some examples showing the kinds of theorem environment declarations that are possible.

```

\newtheorem{prop}{Proposition}
\newtheorem{thm}{Theorem}[section]
\newtheorem{lem}[thm]{Lemma}
\newtheorem*{Zorn}{Zorn's Lemma}

\theoremstyle{definition}
\newtheorem{dfn}{Definition}

\theoremstyle{remark}
\newtheorem*{rmk}{Remark}

```

The first four statements all define environments using the default theorem style ('plain'), since there is no prefatory `\theoremstyle` declaration. The first statement defines an automatically numbered `prop` environment whose headings will look like this: Proposition 1, Proposition 2, and so forth. The second statement defines an environment `thm` with numbers subordinate to section numbers, so the headings will look like this: Theorem 1.1, Theorem 1.2, Theorem 1.3, . . . , (in section 2:) Theorem 2.1, Theorem 2.2, and so forth. The third statement defines a `lem` environment whose numbers will interleave in sequence with the theorem numbers: Theorem 1.3, Lemma 1.4, Lemma 1.5, Theorem 1.6, and so forth. The fourth statement defines a special unnumbered lemma named 'Zorn's Lemma'. The remaining two `\newtheorem` statements have no special features except for the `\theoremstyle` declarations that cause the `dfn` and `rmk` environments to have some differences in appearance.

There are three basic styles provided: The 'plain' style produces bold headings and italic body text; the 'definition' style produces bold headings and normal body text; the 'remark' style produces italic headings and normal body text.

A `\swapnumbers` command allows theorem numbers to be swapped to the front of the theorem headings. Putting `\swapnumbers` in your document preamble will cause *all following* `\newtheorem` statements to produce number-first headings. (To provide maximum control, `\swapnumbers` is designed so that it can be used more than once; each time it is used, theorem numbers will be swapped to the opposite side for all following `\newtheorem` statements. But rarely will it need to be invoked more than once per document.)

### Custom theorem styles

There is a `\newtheoremstyle` command provided to make the creation of custom theoremstyles fairly easy.

Usage:

```

#1
\newtheoremstyle{NAME}%
#2 #3 #4
{ABOVESPACE}{BELOWSPACE}{BODYFONT}%
#5 #6 #7 #8
{INDENT}{HEADFONT}{HEADPUNCT}{HEADSPACE}%
#9
{CUSTOM-HEAD-SPEC}

```

Leaving the ‘indent’ argument empty is equivalent to entering `Opt`. The ‘head-punct’ and ‘headspace’ arguments are for the punctuation and horizontal space between the theorem head and the following text. There are two special values that may be used for ‘headspace’: a single space means that a normal interword space should be used; “`\newline`” means that there should be a line break after the head instead of horizontal space. The ‘custom-head-spec’ argument follows a special convention: it is interpreted as the replacement text for an internal three-argument function `\thmhead`, i.e., as if you were defining

```
\renewcommand{\thmhead}[3]{...#1...#2...#3...}
```

but omitting the initial `\renewcommand{\thmhead}[3]`. The three arguments that will be supplied to `\thmhead` are the name, number, and optional note components. Within the replacement text you can (and normally will want to) use other special functions `\thmname`, `\thmnumber`, and `\thmnote`. These will print their argument if and only if the corresponding argument of `\thmhead` is nonempty. For example

```
{\thmname{#1}\thmnumber{ #2}\thmnote{ (#3)}}
```

This would cause the theorem note `#3` to be printed with a preceding space and enclosing parentheses, if it is present, and if it is absent, the space and parentheses will be omitted because they are inside the argument of `\thmnote`.

Finally, if you have an extra bit of arbitrary code that you want to slip in somewhere, the best place to do it is in the ‘body font’ argument.

The `\newtheoremstyle` command is designed to provide, through a relatively simple interface, control over the style aspects that are most commonly changed. More complex requirements must be addressed through a separate  $\text{\LaTeX}$  package.

When you set up custom theorem styles with `\newtheoremstyle` you should not use `\swapnumbers`. You have full control of the ordering of elements in the theorem head, just place them where you want. Or, if you do use `\swapnumbers`, you must look at the definition of `\swappedhead` and change it appropriately.

### Implementation

The `\theoremstyle` command is very simple except for the need to warn about an unknown theoremstyle.

```

1745 ⟨*amsthm | classes⟩
1746 \newcommand{\theoremstyle}[1]{%
1747   \@ifundefined{th@#1}{%
1748     \PackageWarning{amsthm}{Unknown theoremstyle ‘#1’}%
1749     \thm@style{plain}%
1750   }{%
1751     \thm@style{#1}%
1752   }%
1753 }

1754 \newtoks\thm@style
1755 \thm@style{plain}

```

What’s really needed is a full-fledged systematic approach for specifying the desired order and formatting of the three identified parts of a theorem head (name, number, note).

```

1756 \newtoks\thm@bodyfont \thm@bodyfont{\itshape}
1757 \newtoks\thm@headfont \thm@headfont{\bfseries}
1758 \newtoks\thm@notefont \thm@notefont{}
1759 \newtoks\thm@headpunct \thm@headpunct{.}

```

Vertical spacing: initialize to current value of `\topsep`. If a document class loads the `amsthm` package it should take care to set these variables explicitly, if current `\topsep` is not the desired value.

```

1760 \newskip\thm@preskip \newskip\thm@postskip
1761 ⟨*classes⟩
1762 \def\thm@space@setup{%
1763   \thm@preskip=.5\baselineskip\@plus.2\baselineskip
1764   \@minus.2\baselineskip
1765   \thm@postskip=\thm@preskip
1766 }
1767 ⟨/classes⟩
1768 ⟨*amsthm⟩
1769 \def\thm@space@setup{%
1770   \thm@preskip=\topsep \thm@postskip=\thm@preskip
1771 }
1772 ⟨/amsthm⟩

```

Modify `\newtheorem` to add `*` option. If a `*` is found, pass it on to `\@xnthm` as the first argument. (This information enables us to handle two different possibilities in a single function `\@xnthm` instead of needing two separate functions.)

```

1773 \renewcommand{\newtheorem}{\@ifstar{\@xnthm *}{\@xnthm \relax}}

```

Check to see if an optional arg is present after the first mandatory arg (which is `#2` at the moment since the `*` or non-`*` is `#1`).

```

1774 \def\@xnthm#1#2{%
1775   \let\@tempa\relax
1776   \@xp\@ifdefinable\csname #2\endcsname{%
1777     \global\@xp\let\csname end#2\endcsname\@endtheorem
1778     \ifx *#1% unnumbered, need to get one more mandatory arg
1779       \edef\@tempa##1{%

```

```

1780     \gdef\@xp\@nx\curname#2\endcurname{%
1781         \@nx\@thm{\@xp\@nx\curname th@the\thm@style\endcurname}%
1782         }{##1}}}%
1783     \else % numbered theorem, need to check for optional arg
1784         \def\@tempa{\@oparg{\@ynthm{#2}} []}%
1785     \fi
1786 }%
1787 \@tempa
1788 }

```

Environment numbered together with a previously defined environment.

Arg1: env name, e.g. ‘thm’

Arg2: optional sibling counter

Arg3: heading text e.g. ‘Theorem’

```

1789 \def\@ynthm#1[#2]#3{%

```

If optional arg #2 is empty, call \@xthm to look for a possible optional arg in terminal position. Note that the two optional args are mutually exclusive. As #2 is a counter name and must be processed by \curname anyway, we can use a simpler test instead of \@ifempty.

```

1790     \ifx\relax#2\relax
1791         \def\@tempa{\@oparg{\@xthm{#1}{#3}} []}%
1792     \else
1793         \@ifundefined{c@#2}{%
1794             \def\@tempa{\@nocounterr{#2}}%
1795         }{%
1796             \@xp\xdef\curname the#1\endcurname{\@xp\@nx\curname the#2\endcurname}%
1797             \toks@{#3}%
1798             \@xp\xdef\curname#1\endcurname{%
1799                 \@nx\@thm{%
1800                     \let\@nx\thm@swap
1801                     \if S\thm@swap\@nx\@firstoftwo\else\@nx\@gobble\fi
1802                     \@xp\@nx\curname th@the\thm@style\endcurname}%
1803                     {#2}{\the\toks@}}%
1804             \let\@tempa\relax
1805         }%
1806     \fi
1807 \@tempa
1808 }

```

Environment numbered relative to the counter given as #3. This function should really be named \@znthm but we’re trying to save a bit of hash table and string pool by reusing one of the command names rendered obsolete by the amsthm package.

Arg1: env name e.g. ‘thm’; Arg2: heading text e.g. ‘Theorem’; Arg3: parent counter e.g. section.

```

1809 \def\@xthm#1#2[#3]{%

```

Set up the counter c@#1 and optionally add it to the reset list of counter #3. As #3 is a counter name and must be processed by \curname anyway, we can use a

simpler test instead of `\@ifempty`.

```

1810 \ifx\relax#3\relax
1811   \newcounter{#1}%
1812 \else
1813   \newcounter{#1}[#3]%

```

Define `\thexxx` to be `\theyyy`. `\arabic{xxx}` (assuming default values of punctuation and numbering style). The use of `\xdef` here is inherited from the old L<sup>A</sup>T<sub>E</sub>X code, I'm not sure it's a good idea in general, but there should not be any problems unless someone changes the value of `\@thmcountersep` or `\@thmcounter`.

```

1814   \@xp\xdef\csname the#1\endcsname{\@xp\@nx\csname the#3\endcsname
1815     \@thmcountersep\@thmcounter{#1}}%
1816 \fi
1817 \toks@{#2}%
1818 \@xp\xdef\csname#1\endcsname{%
1819   \@nx\@thm{
1820     \let\@nx\thm@swap
1821     \if S\thm@swap\@nx\@firstoftwo\else\@nx\@gobble\fi
1822     \@xp\@nx\csname th@\the\thm@style\endcsname}%
1823     {#1}{\the\toks@}}%
1824 }

```

[mjd,1999/10/13] The following code doesn't handle the case where an equation is immediately followed by a theorem with no intervening `\par`—then the `spacefactor` is 1000.

```

1825 % % \def\thm@check@break{%
1826 % %   \ifhmode \unskip \unskip
1827 % %     \edef\pre@thm@spacefactor{\the\spacefactor}\par
1828 % %     \edef\thm@topbreak{%
1829 % %       \ifnum\pre@thm@spacefactor<\sfcode'\!\relax
1830 % %         % preceding text line did not end with end-of-sentence punctuation
1831 % %         \ifnum\prevgraf=\@ne \penalty\widowpenalty
1832 % %         \else \penalty9999\relax
1833 % %         \fi
1834 % %       \else
1835 % %         \@nx\addpenalty{\@beginparpenalty}%
1836 % %       \fi
1837 % %     }%
1838 % %   \else
1839 % %     \def\thm@topbreak{\addpenalty\@beginparpenalty}%
1840 % %   \fi
1841 % % }

```

If arg #2 is empty, this is an unnumbered environment; otherwise #2 is the name of a counter. #3 is descriptive name such as “Theorem” or “Lemma”. Arg #1 is the style function, for example `\th@plain`.

```

1842 \def\@thm#1#2#3{%
1843   \ifhmode\unskip\unskip\par\fi
1844   \normalfont

```

1845 `\trivlist`

Explicitly set plain style here, then override parts as necessary in the function provided as #1.

```

1846 \let\thmheadnl\relax
1847 \let\thm@swap\@gobble
1848 <amsart> \let\thm@indent\noindent % no indent
1849 <amsart> \thm@headfont{\bfseries}% heading font bold
1850 <amsbook|amsproc> \let\thm@indent\indent % indent
1851 <amsbook|amsproc> \thm@headfont{\scshape}% heading font small caps
1852 \thm@notefont{\fontseries\mdefault\upshape}%
1853 \thm@headpunct{.}% add period after heading
1854 \thm@headsep 5\p@ plus\p@ minus\p@\relax
1855 \thm@space@setup
1856 #1% style overrides
1857 \@topsep \thm@preskip % used by thm head
1858 \@topsepadd \thm@postskip % used by \@endparenv
1859 \def\@tempa{#2}\ifx\@empty\@tempa
1860 \def\@tempa{\@oparg{\@begintheorem{#3}-{}}{}}%
1861 \else
1862 \refstepcounter{#2}%
1863 \def\@tempa{\@oparg{\@begintheorem{#3}-{\csname the#2\endcsname}}{}}%
1864 \fi
1865 \@tempa
1866 }

```

The internal function `\@restorelabelsep` starts out as a no-op. I don't think this is needed any more [mjd,2000/10/26].

1867 `\def\@restorelabelsep{\relax}`

This variation of the `\@thm` command is no longer needed. The variation `\@xthm` was commandeered for `\newtheorem` use.

1868 `\let\@ythm\relax`

Init `\thmname` etc.

1869 `\let\thmname\@iden \let\thmnote\@iden \let\thmnumber\@iden`

`\@upn` The function `\@upn` is used to force theorem numbers and similar elements to be upright even when the current font is italic. If a suitable italic font with upright numbers and punctuation is available, this function should be redefined to be a no-op.

1870 `\providecommand\@upn{\textup}`

Definitions for theorem heads.

1871 `\def\thmhead@plain#1#2#3{%`

To allow for the case where the `thmname` part is empty and the heading consists only of a number (don't laugh, we have examples from real mathematical manuscripts), we don't add the space at the beginning of `thmnumber` unless #1 is nonempty.

1872 `\thmname{#1}\thmnumber{\@ifnotempty{#1}{ }\@upn{#2}}%`

In `thmnote` we always add a leading space, i.e., assuming that at least one of the preceding parts will always be present.

```
1873 \thmnote{ {\the\thm@notefont(#3)}}}
1874 \let\thmhead\thmhead@plain
```

`Swappedhead` is for the case where the number precedes the word "Theorem". When the numbers fall on the left, like section numbers, AMS journal style calls for them to be lightface. We get this by invoking `\@secnumfont`.

In handling the punctuation after the number we have to step delicately if we want to successfully handle cases where the theorem name is empty (this is very rare, yet it sometimes arises in practice).

```
1875 \def\swappedhead#1#2#3{%
1876   \amsthm \thmnumber{#2}%
1877   \amsthm \thmname{\@ifnotempty{#2}{~}#1}%
1878   \classes \thmnumber{\@upn{\@secnumfont#2\@ifnotempty{#1}{.~}}}%
1879   \classes \thmname{#1}%
1880   \thmnote{ {\the\thm@notefont(#3)}}}
```

A customized definition of `\th@plain` written for version 1.x of the `amsthm` package might refer to `\swappedhead@plain`; this gives it a definition for backward compatibility.

```
1881 \let\swappedhead@plain=\swappedhead
```

In `\@begintheorem \thmheadnl` is called after the theorem head: maybe a newline, otherwise a no-op.

```
1882 \let\thmheadnl\relax
1883 \let\thm@indent\noindent
1884 \let\thm@swap@gobble
```

If argument #2 is empty, then this is an unnumbered environment. Otherwise #2 is a numbering command such as `\thexyz`. We use `\deferred@thm@head` instead of `\item` in order to allow line breaking in the optional note argument.

```
1885 \def\@begintheorem#1#2[#3]{%
1886   \deferred@thm@head{\the\thm@headfont \thm@indent
```

Changes to `\thmnumber` and `\thmnote` are local to the containing box.

```
1887   \@ifempty{#1}{\let\thmname@gobble}{\let\thmname@iden}%
1888   \@ifempty{#2}{\let\thmnumber@gobble}{\let\thmnumber@iden}%
1889   \@ifempty{#3}{\let\thmnote@gobble}{\let\thmnote@iden}%
```

The `\thm@swap` function selects either `\swappedhead` or `\thmhead`.

```
1890   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
```

I can't think of any example where the after-head punctuation should be omitted so it seems correct not to use `\@addpunct` here.

```
1891   \the\thm@headpunct
```

If this *is* a newline (from a `\newtheoremstyle`), it gets lost if there isn't any text following the heading, since `\deferred@thm@head` packs the heading into an `hbox` with `\sbox\@labels`. Attempting to move the `\thmheadnl` outside the scope of `\deferred@thm@head` (just outside the ending brace below) results in

no improvement if no text follows the heading, and where there is text, runs it in. Inserting a space following such a heading results in an extra blank line. [bnb, 2004/06/30]

```

1892   \thmheadnl % possibly a newline.
1893   \hskip\thm@headsep
1894 }%
1895 \ignorespaces}

1896 \newskip\thm@headsep
1897 \thm@headsep=5pt plus1pt minus1pt\relax

1898 \let\adjust@parskip@nbreak=\@nbitem

1899 \newtoks\dth@everypar
1900 \dth@everypar={%
1901   \@minipagefalse \global\@newlistfalse
1902   \@noparitemfalse
1903   \if@inlabel
1904     \global\@inlabelfalse
1905     \begingroup \setbox\z@\lastbox
1906     \ifvoid\z@ \kern-\itemindent \fi
1907     \endgroup
1908     \unhbox\@labels
1909   \fi
1910   \if@nbreak \@nbreakfalse \clubpenalty\@M
1911   \else \clubpenalty\@clubpenalty \everypar{}%
1912   \fi
1913 }%

1914 \def\deferred@thm@head#1{%
1915   \if@inlabel \indent \par \fi % eject a section head if one is pending
1916   \if@nbreak

```

This case normally arises when a theorem follows immediately after a section head. Then we leave the below-section-head space instead of adding above-theorem space; but some adjustment of parskip is needed.

```

1917   \adjust@parskip@nbreak
1918   \else
1919     \addpenalty\@beginparpenalty
1920     \addvspace\@topsep
1921     \addvspace{-\parskip}%
1922   \fi
1923   \global\@inlabeltrue
1924   \everypar\dth@everypar
1925   \sbox\@labels{\normalfont#1}%
1926   \ignorespaces
1927 }

```

`\nonslanted` The `\nonslanted` command changes the current font to `\upshape` if it is `\itshape` or `\slshape`. This is used for document structure numbers that should be consistently upright in all contexts.

```

1928 \def\nonslanted{\relax

```

Can't do a direct `\ifx` between `\f@shape` and `\itdefault` because the latter is `\long` (grumble grumble).

```
1929 \exp\let\@xp\@tempa\cshape shape\endcshape
1930 \ifx\@tempa\itshape\upshape
1931 \else\ifx\@tempa\slshape\upshape\fi\fi}
```

`\swapnumbers` The `\swapnumbers` command sets a switch `\thm@swap` that is used by `\newtheorem`. To conserve hash table we load `\thm@swap` with two uses; the first one is needed only in `\newtheorem` declarations and the second one is needed only in typesetting theorem environments.

```
1932 \def\swapnumbers{\edef\thm@swap{\if S\thm@swap N\else S\fi}}
1933 \def\thm@swap{N}%
```

`\@opargbegintheorem` not needed, `\@oparg` utility serves instead.

```
1934 \let\@opargbegintheorem\relax
```

Except for the body font, default values are built into `\@thm`.

```
1935 \def\th@plain{%
1936 %% \let\thm@indent\noindent % no indent
1937 %% \thm@headfont{\bfseries}% heading font is bold
1938 %% \thm@notefont{ }% same as heading font
1939 %% \thm@headpunct{.}% add period after heading
1940 %% \let\thm@swap\@gobble
1941 %% \thm@preskip\topsep
1942 %% \thm@postskip\theorempreskipamount
1943 \itshape % body font
1944 }
```

Theorem style 'definition' is the same as 'plain' except for the body font.

```
1945 \def\th@definition{%
1946 \normalfont % body font
1947 }
```

Theorem style 'remark' differs from 'plain' in head font and body font. Also smaller spacing above and below for AMS classes only.

```
1948 \def\th@remark{%
1949 <amsart|amsthm> \thm@headfont{\itshape}%
1950 \normalfont % body font
1951 <*amsthm>
1952 \thm@preskip\topsep \divide\thm@preskip\tw@
1953 \thm@postskip\thm@preskip
1954 </amsthm>
1955 }
```

The standard definition of `\@endtheorem` is just `\endtrivlist`, but that doesn't automatically start a new paragraph, so we add `\@endpfalse` in order to ensure a new paragraph. This differs from the basic L<sup>A</sup>T<sub>E</sub>X behavior.

```
1956 \def\@endtheorem{\endtrivlist\@endpfalse }
```

`\newtheoremstyle` An easy way to make a not too complicated variant theorem style. Usage:

```

#1
\newtheoremstyle{NAME}%
#2 #3 #4
{ABOVESPACE}{BELOWSPACE}{BODYFONT}%
#5 #6 #7 #8
{INDENT}{HEADFONT}{HEADPUNCT}{HEADSPACE}%
#9
{CUSTOM-HEAD-SPEC}
1957 \newcommand{\newtheoremstyle}[9]{%
Empty or Opt for #5 is translated to \noindent.
1958 \ifempty{#5}{\dimen@z@skip}{\dimen@#5\relax}%
1959 \ifdim\dimen@=\z@
#4 is body font. Extra code could be included there if necessary.
1960 \toks@{#4\let\thm@indent\noindent}%
1961 \else
1962 \toks@{#4\def\thm@indent{\noindent\hbox to#5{}}}%
1963 \fi
Arg #8 is a glue spec for the space after the head. As a proper glue spec for
‘normal interword space’ is rather hard to write, we recognize an argument of { }
as a special case and translate internally to the necessary fontdimen equivalent.
Furthermore, if #8 consists entirely of \newline, then we will perform a line
break after the theorem head instead of adding horizontal space. At the moment
[1995/01/23] this is not perfectly well implemented because of complications
with the way LATEX’s \item adds a heading to the vertical list; for best results
there should not be anything (not even a blank line) after the \begin{xxx}
command.
1964 \def@tempa{#8}\ifx\space\tempa
Notice that we disregard stretch and shrink for labelsep = interwordspace.
1965 \toks@\@xp{\the\toks@ \thm@headsep\fontdimen\tw@\font\relax}%
1966 \else
1967 \def@tempb{\newline}%
1968 \ifx\tempb\tempa
1969 \toks@\@xp{\the\toks@ \thm@headsep\z@skip
1970 \def\thmheadnl{\newline}}%
1971 \else
1972 \toks@\@xp{\the\toks@ \thm@headsep#8\relax}%
1973 \fi
1974 \fi
1975 \begingroup
1976 \thm@space@setup
1977 \@defaultunits\@tempskipa#2\thm@preskip\relax\@nnil
1978 \@defaultunits\@tempskipb#3\thm@postskip\relax\@nnil
1979 \xdef@gtempa{\thm@preskip\the\@tempskipa
1980 \thm@postskip\the\@tempskipb\relax}%

```

```

1981 \endgroup
1982 \@temptokena\@xp{\@gtempa
1983   \thm@headfont{#6}\thm@headpunct{#7}}%
1984 }%
1985 \@ifempty{#9}{%
1986   \let\thmhead\thmhead@plain
1987 }{%
1988   \@namedef{thmhead@#1}##1##2##3{#9}%
1989   \@temptokena\@xp{\the\@temptokena
1990     \@xp\let\@xp\thmhead\curname thmhead@#1\endcsname}%
1991 }%
1992 \@xp\xdef\curname th@#1\endcsname{\the\toks@ \the\@temptokena}%
1993 }

```

`\qed` Define `\qed` for end of proof. This command might occur in math mode, in a displayed equation, but it should never occur in inner math mode in ordinary paragraph text.

```

1994 \DeclareRobustCommand{\qed}{%
1995   \ifmmode \mathqed
1996   \else
1997     \leavevmode\unskip\penalty9999 \hbox{} \nobreak\hfill

```

The `\hbox` is to prevent a line break within the `\qedsymbol` if it is defined to be something composite— e.g., things like (Corollary 1.2) `\openbox` as are occasionally done.

```

1998   \quad\hbox{\qedsymbol}%
1999   \fi
2000 }

```

Setup for QED stack. We use a dedicated iterator macro `\qed@elt` instead of the generic `\@elt` because the  $\LaTeX$  output routine is not safe against changes in `\@elt`. Therefore it is not safe to use `\@elt` for any processing that might trigger the output routine. Although this does not seem very likely when adding a QED symbol to a horizontal list, we did in fact get a bug report for this kind of failure.

```

2001 \let\QED@stack\@empty
2002 \let\qed@elt\relax

```

Puts a QED symbol on the stack:

```

2003 \newcommand{\pushQED}[1]{%
2004   \toks@\{\qed@elt{#1}\}\@temptokena\expandafter{\QED@stack}%
2005   \xdef\QED@stack{\the\toks@\the\@temptokena}%
2006 }

```

Pops the QED stack and prints the result.

```

2007 \newcommand{\popQED}{%
2008   \begingroup\let\qed@elt\popQED@elt \QED@stack\relax\relax\endgroup
2009 }
2010 \def\popQED@elt#1#2\relax{#1\gdef\QED@stack{#2}}

```

Prints the current QED symbol and replaces the top entry on the stack with a null entry.

```
2011 \newcommand{\qedhere}{%
2012   \begingroup \let\mathqed\math@qedhere
2013   \let\qed@elt\setQED@elt \QED@stack\relax\relax \endgroup
2014 }
```

In case the `amsmath` or `amstext` packages are loaded, we need the following two tests. (Redundantly declaring them is harmless.)

```
2015 \newif\ifmeasuring@
2016 \newif\iffirstchoice@ \firstchoice@true
2017 \def\setQED@elt#1#2\relax{%
2018   \ifmeasuring@
2019   \else \iffirstchoice@ \gdef\QED@stack{\qed@elt{#2}}\fi
2020   \fi
2021   #1%
2022 }
```

`\mathqed` When a QED occurs inside a math formula, well, it is presumably a displayed equation. In order to find out where to place the QED symbol, we need to check what kind of equation structure we are in.

```
2023 \def\qed@warning{%
2024   \PackageWarning{amsthm}{The \@nx\qedhere command may not work
2025     correctly here}%
2026 }
2027 \newcommand{\mathqed}{\quad\hbox{\qedsymbol}}
```

The `\linebox@qed` function comes into play with the `fleqn` option.

```
2028 \def\linebox@qed{\hfil\hbox{\qedsymbol}\hfilneg}
```

Two large sections of code follow. One for `amsmath` 2.0 and one for plain `LATEX`.

```
2029 \@ifpackageloaded{amsmath}{%
2030   \def\math@qedhere{%
2031     \ifundefined{\@currenvir @qed}{%
2032       \qed@warning\quad\hbox{\qedsymbol}%
2033     }{%
2034       \@xp\aftergroup\csname\@currenvir @qed\endcsname
2035     }%
2036   }
2037   \def\displaymath@qed{%
2038     \relax
2039     \ifmmode
2040       \ifinner \aftergroup\linebox@qed
2041     \else
2042       \eqno
2043     \let\eqno\relax \let\leqno\relax \let\veqno\relax
```

```

2044     \hbox{\qedsymbol}%
2045     \fi
2046   \else
2047     \aftergroup\linebox@qed
2048     \fi
2049   }
2050 \exp\let\csname equation*@qed\endcsname\displaymath@qed

```

If the equation has both an equation number and a qed, we've got trouble, but we can provide half-way decent for the simple cases.

```

2051 \def\equation@qed{%
2052   \iftagsleft@
2053     \hbox{\phantom{\quad\qedsymbol}}%
2054     \gdef\alt@tag{%
2055       \rlap{\hbox to\displaywidth{\hfil\qedsymbol}}%
2056       \global\let\alt@tag\@empty
2057     }%
2058   \else
2059     \gdef\alt@tag{%
2060       \global\let\alt@tag\@empty
2061       \vtop{\ialign{\hfil####\cr
2062         \tagform@\theequation\cr
2063         \qedsymbol\cr}}%
2064       \setbox\z@
2065     }%
2066     \fi
2067   }
2068 \def\qed@tag{%
2069   \global\tag@true \nonumber
2070   &\omit\setboxz@h {\strut@ \qedsymbol}\tagsleft@false
2071   \place@tag@gather
2072   \kern-\tabskip
2073   \ifst@rred \else \global\@eqnswtrue \fi \global\advance\row@\@ne \cr
2074 }
2075 \def\split@qed{%
2076   \def\endsplit{\crr\egroup \egroup \ctagsplit@false \rendsplit@
2077     \aftergroup\align@qed
2078   }%
2079 }
2080 \def\align@qed{%
2081   \ifmeasuring@ \tag*{\qedsymbol}%
2082   \else \let\math@cr@@\qed@tag
2083   \fi
2084 }
2085 \exp\let\csname align*@qed\endcsname\align@qed
2086 \exp\let\csname gather*@qed\endcsname\align@qed
2087 %% Needs some patching up for amsmath 1.2
2088 }{% end of amsmath branch, start plain LaTeX branch

```

The `\qedhere` handling for generic L<sup>A</sup>T<sub>E</sub>X classes (`article`, `book`) with the `amsthm` package (without `amsmath`) is fairly sketchy. When a `qed` symbol and an equation number are both applied to a single display, the results may not be entirely satisfactory, particularly when the `fleqn` and/or `leqno` options are used. [mjd,2000/01/17]

As for `\math@qedhere`, it is expected to occur only via `\qedhere`, where the `\aftergroup` makes sense.

```
2089 \def\math@qedhere{%
2090   \ifundefined{\@currentvir @qed}{%
2091     \qed@warning \aftergroup\displaymath@qed
2092   }{%
2093     \xp\aftergroup\csname\@currentvir @qed\endcsname
2094   }%
2095 }
```

The `\ifmmode \ifinner` case is expected to happen with the `fleqn` option, where we have something like:

```
\hbox to\linewidth{\hbox{$.$.}\hfil}
```

In order to counteract the `\hfil` we must jump out two grouping levels.

```
2096 \def\displaymath@qed{%
2097   \relax
2098   \ifmmode
2099     \ifinner \aftergroup\aftergroup\aftergroup\linebox@qed
2100   \else
2101     \eqno \def\@badmath{$$$}%
2102     \let\eqno\relax \let\leqno\relax \let\veqno\relax
2103     \hbox{\qedsymbol}%
2104   \fi
2105 \else
2106   \aftergroup\linebox@qed
2107 \fi
2108 }
```

This definition is a fallback definition that places the `qed` and then the equation number, on the right-hand side. For `leqno`, not so good; but then

```
2109 \ifundefined{ver@leqno.clo}{%
2110   \def\equation@qed{\displaymath@qed \quad}%
2111 }{%
2112   \def\equation@qed{\displaymath@qed}%
2113 }
```

If `amsmath` is not loaded, then we need to do some surgery on the `\[` macro. Normally it looks like this:

```
\[=macro:
->\relax \ifmmode \@badmath \else
\ifvmode \nointerlineskip \makebox [.6\linewidth ]\fi $$$\fi
```

If arg 2 is `\m@th` when we make this test it indicates that the `fleqn` option is in effect. Perhaps try to do something there.

```

2114 \def\@tempa#1$#2#3\@nil#4{%
2115     \def#4{#1$#2\def\@currenvir{displaymath}#3}%
2116 }%

```

If `\[` has already been made robust (by `fixltx2e` or a L<sup>A</sup>T<sub>E</sub>X format from 2015 or later), patch the internal macro `\[<space>` instead.

```

2117 \expandafter\ifx\csname[ \endcsname\relax
2118     \expandafter\@tempa\[\@nil\[%
2119 \else
2120     \expandafter\expandafter\expandafter\@tempa\csname[
2121     \expandafter\endcsname\expandafter\@nil
2122     \csname[ \endcsname
2123 \fi
2124 }

```

If an older version of `amsmath` is in use, we need to fall back to a simpler definition of `\math@qedhere`.

```

2125 \ifpackageloaded{amstex}{%
2126     \def\@tempa{TT}%
2127 }{%
2128     \ifpackageloaded{amsmath}{%
2129         \def\@tempb#1 v#2.#3\@nil{#2}%
2130         \ifnum\@xp\@xp\@xp\@tempb\csname ver@amsmath.sty\endcsname v0.0\@nil
2131             <\tw@
2132             \def\@tempa{TT}%
2133         \else
2134             \def\@tempa{TF}%
2135         \fi
2136     }{%
2137         \def\@tempa{TF}
2138     }%
2139 }
2140 \if\@tempa
2141     \renewcommand{\math@qedhere}{\quad\hbox{\qedsymbol}}%
2142 \fi

```

The reason that we do not simply use the `\square` symbol from `msam` for the open-box qed symbol is that we want to avoid requiring users to have the `AMSFonTS` font package. And the `lasy` `\Box` is too large.

```

2143 \newcommand{\openbox}{\leavevmode

```

I think I got these numbers from measuring `msam`'s `\square` but I forgot to make notes at the time. [mjd,1995/01/25]

```

2144 \hbox to.77778em{%
2145     \hfil\vrule
2146     \vbox to.675em{\hrule width.6em\vfil\hrule}%
2147     \vrule\hfil}}
2148 \DeclareRobustCommand{\textsquare}{%
2149     \begingroup \usefont{U}{msa}{m}{n}\thr@@\endgroup
2150 }

```

```

2151 <*classes>
2152 \@ifclasswith{\@classname}{noamsfonts}{%
2153   \providecommand{\qedsymbol}{\openbox}%
2154 }{}
2155 \providecommand{\qedsymbol}{\textsquare}
2156 </classes>
2157 <amsthm>\providecommand{\qedsymbol}{\openbox}

```

The proof environment is never numbered, and has a `\qed` at the end, which makes it inconvenient to use `\newtheorem` for defining it. Also authors frequently need to substitute an alternative heading text (e.g. ‘Proof of Lemma 4.3’) instead of the default ‘Proof’. For all these reasons we define the proof environment here instead of leaving it for authors to define. Text after the end of a proof, like that after the end of a theorem, begins a new paragraph. This differs from basic L<sup>A</sup>T<sub>E</sub>X behavior. [bnb, 1999/09/27]

```

2158 \newenvironment{proof}[1][\proofname]{\par
2159   \pushQED{\qed}%
2160   \normalfont \topsep6\p@\@plus6\p@\relax
2161   \trivlist
2162 <amsbook|amsproc> \itemindent\normalparindent
2163   \item[\hskip\labelsep
2164 <amsbook|amsproc> \scshape
2165 <amsart|amsthm> \itshape
2166   #1\@addpunct{.}]\ignorespaces
2167 }{}
2168 \popQED\endtrivlist\@endpefalse
2169 }

```

Default for `\proofname`:

```

2170 \providecommand{\proofname}{Proof}

```

For reference:

```

From: tycchow@math.mit.edu (Timothy Y. Chow)
Subject: Suppressing theorem numbering in LaTeX
Message-ID: <1994Aug11.234754.22523@galois.mit.edu>
Date: Thu, 11 Aug 94 23:47:54 GMT
To: tex-news@SHSU.EDU

```

A friend of mine wants numbering of theorems, conjectures, and so on suppressed if there is only one of them in his article. In other words he wants "Conjecture 1" to appear as simply "Conjecture" if there is no Conjecture 2. What is the best way to go about doing this?

...

Maybe something clever can be done to make the desired behavior happen automatically. Note that this would seem to be a general numbering problem rather than a theorem-specific one, because similar behavior would be desirable for appendices: according to standard publishing practice, if there's only one it is titled just ‘Appendix’, and if there are more than one they are titled ‘Appendix A’, ‘Appendix B’, and so on.

```
2171 </amsthm | classes>
2172 <*classes>
```

## 2.26 Utility commands used with AMS author packages

`\bb@skip` Skip to result in base-to-base distance from previous to next box.

```
2173 \def\bb@skip#1{%
2174   \skip@#1\relax \advance\skip@-\prevdepth \advance\skip@-\baselineskip
2175   \vskip\skip@}
```

`\markleft` Basic L<sup>A</sup>T<sub>E</sub>X has `\markright` and `\markboth`, but sometimes it's necessary to change just the left running head. This macro completes the set. From the `alttext.sty` module of Klaus Lagally's `arabtex`. [bnb, 2004/03/25]

```
2176 \def\markleft#1{{\let\protect\noexpand
2177   \let\label\relax \let\index\relax \let\glossary\relax
2178   \expandafter\@markleft\@themark{#1}%
2179   \markf\@themark}}%
2180   \if@nobreak\ifvmode\nobreak\fi\fi}
2181 \def\@markleft#1#2#3{\gdef\@themark{#3}{#2}}
```

`\DH` The Icelandic thorn and eth and the Croatian barred D are part of the T1 font encoding, but aren't available in OT1. However, they are needed (rarely) in the author names or bibliographies. Provide emulations, using the thorn in `msbm` or D's barred with a macron. The lowercase eth requires an `\edef` to access `msbm` properly. Define it separately to permit checking for small caps. [bnb, 2004/04/05]

```
2182 \ifundefined{symAMSb}{%
2183   \def\@dh{dh}%
2184   \ClassWarning{\@classname}{\string\dh\space unavailable without amsfonts;
2185     replaced by 'dh'}%
2186 }{%
2187   \edef\@dh{\noexpand\mathhexbox{\hexnumber@\symAMSb}67}%
2188 }
2189 \DeclareTextCommand{\dh}{OT1}{%
2190   \edef\@tempb{\scdefault}%
2191   \ifx\f@shape\@tempb
2192     \leavevmode
2193     \raisebox{- .8ex}{\makebox[\z@][l]{\hskip-.08em\accent"16\hss}}d%
2194   \else
2195     \@dh
2196   \fi
2197 }
2198 \DeclareTextCommand{\DH}{OT1}{%
2199   \leavevmode\raisebox{- .5ex}{\makebox[\z@][l]{\hskip-.07em\accent"16\hss}}D}
2200 \DeclareTextCommand{\DJ}{OT1}{%
2201   \leavevmode\raisebox{- .5ex}{\makebox[\z@][l]{\hskip-.07em\accent"16\hss}}D}
2202 \DeclareTextCommand{\dj}{OT1}{%
2203   \edef\@tempa{\f@shape}\edef\@tempb{\scdefault}%
2204   \ifx\@tempa\@tempb
2205     \leavevmode
```

```

2206   \raisebox{-.75ex}{\makebox[\z@][l]{\hskip-.08em\accent"16\hss}}d%
2207   \else
2208   \leavevmode\raisebox{.02ex}{\makebox[\z@][l]{\hskip.1em\accent"16\hss}}d%
2209   \fi}

```

## 2.27 Hyphenation exceptions

Some common hyphenation exceptions, based on the listing in TUGboat vol 10, no 3, November 1989, pp. 336–341. Many words from the TUGboat list that seemed less likely to occur in mathematical text have been omitted because hyphenation exceptions use up memory (most versions of T<sub>E</sub>X currently have a limit of 307 on hyphenation exceptions; each added hyphenation exception uses up something like 2 extra words of main memory as well). The list of hyphenations for proper names could be expanded forever if room permitted; we restrict ourselves to a rather short, extremely arbitrary list. Note that the hyphenation rules of British English differ in some particulars from the US rules, so some of the hyphenations given below may need to be overridden for proper UK hyphenation.

```

2210 \hyphenation{acad-e-my acad-e-mies af-ter-thought anom-aly anom-alies
2211 an-ti-deriv-a-tive an-tin-o-my an-tin-o-mies apoth-e-o-ses
2212 apoth-e-o-sis ap-pen-dix ar-che-typ-al as-sign-a-ble as-sist-ant-ship
2213 as-ymp-tot-ic asyn-chro-nous at-trib-uted at-trib-ut-able bank-rupt
2214 bank-rupt-cy bi-dif-fer-en-tial blue-print busier busiest
2215 cat-a-stroph-ic cat-a-stroph-i-cally con-gress cross-hatched data-base
2216 de-fin-i-tive de-riv-a-tive dis-trib-ute dri-ver dri-vers eco-nom-ics
2217 econ-o-mist elit-ist equi-vari-ant ex-quis-ite ex-tra-or-di-nary
2218 flow-chart for-mi-da-ble forth-right friv-o-lous ge-o-des-ic
2219 ge-o-det-ic geo-met-ric griev-ance griev-ous griev-ous-ly
2220 hexa-dec-i-mal ho-lo-no-my ho-mo-thetic ideals idio-syn-crazy
2221 in-fin-ite-ly in-fin-i-tes-i-mal ir-rev-o-ca-ble key-stroke
2222 lam-en-ta-ble light-weight mal-a-prop-ism man-u-script mar-gin-al
2223 meta-bol-ic me-tab-o-lism meta-lan-guage me-trop-o-lis
2224 met-ro-pol-i-tan mi-nut-est mol-e-cule mono-chrome mono-pole
2225 mo-nop-oly mono-spline mo-not-o-nous mul-ti-fac-et-ed mul-ti-plic-able
2226 non-euclid-ean non-iso-mor-phic non-smooth par-a-digm par-a-bol-ic
2227 pa-rab-o-loid pa-ram-e-trize para-mount pen-ta-gon phe-nom-e-non
2228 post-script pre-am-ble pro-ce-dur-al pro-hib-i-tive pro-hib-i-tive-ly
2229 pseu-do-dif-fer-en-tial pseu-do-fi-nite pseu-do-nym qua-drat-ic
2230 quad-ra-ture qua-si-smooth qua-si-sta-tion-ary qua-si-tri-an-gu-lar
2231 quin-tes-sence quin-tes-sen-tial re-arrange-ment rec-tan-gle
2232 ret-ri-bu-tion retro-fit retro-fit-ted right-eous right-eous-ness
2233 ro-bot ro-bot-ics sched-ul-ing se-mes-ter semi-def-i-nite
2234 semi-ho-mo-thet-ic set-up se-vere-ly side-step sov-er-eign spe-cious
2235 spher-oid spher-oid-al star-tling star-tling-ly sta-tis-tics
2236 sto-chas-tic straight-est strange-ness strat-a-gem strong-hold
2237 sum-ma-ble symp-to-matic syn-chro-nous topo-graph-i-cal tra-vers-a-ble
2238 tra-ver-sal tra-ver-sals treach-ery turn-around un-at-tached
2239 un-err-ing-ly white-space wide-spread wing-spread wretch-ed
2240 wretch-ed-ly Eng-lish Euler-ian Feb-ru-ary Gauss-ian

```

2241 Hamil-ton-ian Her-mit-ian Jan-u-ary Japan-ese Kor-te-weg  
 2242 Le-gendre Mar-kov-ian Noe-ther-ian No-vem-ber Rie-mann-ian Sep-tem-ber}

## 2.28 Initialization

We define a function to do the normal calculations that we want for `\textheight` and `\textwidth`

`\calclayout` Subtract the height of the running heads:

```
2243 \def\calclayout{\advance\textheight -\headheight
2244 \advance\textheight -\headsep
```

We set `\oddsidemargin` and `\evensidemargin` to center the text on the page.

```
2245 \oddsidemargin\paperwidth
2246 \advance\oddsidemargin -\textwidth
2247 \divide\oddsidemargin\tw@
```

Now we subtract the default margin provided by standard DVI drivers. But first we make sure that the final margin will be at least .5 inch.

```
2248 \ifdim\oddsidemargin<.5truein \oddsidemargin.5truein \fi
2249 \advance\oddsidemargin -1truein
2250 \evensidemargin\oddsidemargin
```

And we set `\topmargin` to get vertical centering as well.

```
2251 \topmargin\paperheight \advance\topmargin -\textheight
2252 \advance\topmargin -\headheight \advance\topmargin -\headsep
```

Height of running foot ignored: not present.

```
2253 \divide\topmargin\tw@
```

We provide a minimum of .5in (after compensating for the default margin—see next step).

```
2254 \ifdim\topmargin<.5truein \topmargin.5truein \fi
```

Now subtract the default margin provided by standard DVI drivers.

```
2255 \advance\topmargin -1truein\relax
2256 }
```

Initialize the dimensions, page numbering, etc. For inhouse use, administrative stuff is isolated in separate files.

```
2257 <amsart>\InputIfFileExists{amsart.cfg}{-}{%
2258 <amsproc>\InputIfFileExists{amsproc.cfg}{-}{%
2259 <amsbook>\InputIfFileExists{amsbook.cfg}{-}{%
2260 \calclayout % initialize
2261 \pagenumbering{arabic}%
2262 \pagestyle{headings}%
2263 \thispagestyle{plain}%
2264 }
```

If we are in compatibility mode, add some backward compatibility stuff below. Otherwise quit here.

```
2265 \if@compatibility \else\endinput\fi
```

Compensate for changed meaning of `\tiny`:

```
2266 \def\tiny{\Tiny}
```

The macro `\defaultfont` was provided in version 1.1 of `amsart`; retained for compatibility as a synonym of `\normalfont`. Resets everything except for size.

```
2267 \def\defaultfont{\normalfont}
```

Macro for making non-slanted numbers and punctuation in italic or slanted text. This is to avoid visual inconsistencies between numbers or parentheses in math and adjacent numbers or parentheses in text.

```
2268 \def\rom{\textup}
```

For backward compatibility with version 1.1 of `amsart`, we define `pf`, `pf*` environments. And undefine `\proof` just in case an existing document contains a `\newenvironment` or `\newcommand` for it, as that would now cause an error.

```
2269 \let\@newpf\proof \let\proof\relax \let\endproof\relax
```

```
2270 \newenvironment{pf}{\@newpf[\proofname]}\popQED\endtrivlist}
```

```
2271 \newenvironment{pf*}[1]{\@newpf[#1]}\popQED\endtrivlist}
```

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
2272 \endinput
```

```
2273 </classes>
```