TEXsis 2.16
12 November 1994

TEXsis

TEX Macros for Physicists

Eric Myers
Department of Physics and Astronomy
Vassar College
Poughkeepsie, New York

Frank E. Paige
Physics Department
Brookhaven National Laboratory
Upton, New York

TEXsis
TEX Macros for Physicists

Eric Myers

Department of Physics and Astronomy
Vassar College
Poughkeepsie, New York 12601 USA

and

Frank E. Paige

Physics Department
Brookhaven National Laboratory
Upton, New York 11973 USA

TEXsis is a collection of TEX macros for typesetting many kinds of physics documents, including research papers, preprints and "e-prints," conference proceedings, theses, books, referee reports, letters, and memoranda. TEXsis macros provide automatic numbering of equations, automatic numbering and formatting of references, double column formatting, macros for making tables and figures, with or without captions, and including tables with horizontal and vertical rules. TEXsis supports a wide variety of type sizes and a number of specialized document formats, and it even includes macros for making form letters for job applications or letters of recommendation. This document describes version 2.16 of TEXsis, and also serves as an example of its use.

TEXsis: TEX Macros for Physicists

Source code and documentation for TEXsis are available via anonymous ftp over the Internet from lifshitz.ph.utexas.edu. To copy the files, run ftp on your computer and connect to lifshitz.ph.utexas.edu (or 128.83.131.57). Give the login name "anonymous" and your e-mail address as the password. The TEXsis files for Unix machines are in a (possibly compressed) tar file in the subdirectory `texsis`, or in shar bundles in the subdirectory `texsis/Unix`. For Vax/VMS machines the files are available in DCL bundles (.COM files which unpack themselves into many files) in the subdirectory `texsis/VMS`. The code is the same in any case, it's just packed differently. See the `README` file there for more details.

Small improvements and corrections are distributed as patches in the patch file `TXSpatch.tex`. If there is such a file in the `/texsis` subdirectory on lifshitz you should copy it as well.

If you cannot obtain the source files for TEXsis by anonymous ftp, they are available by e-mail from the authors via the Internet, or we can try to make arrangements to make copies on magnetic tape or floppy disks (you provide the media).

We would be grateful to hear of any errors found in this documentation or in the software, and we also invite suggestions for improvements in both. We try to correct errors as quickly as possible, and we try to improve TEXsis as our time allows. When reporting an error in the code it would help a great deal if you would send us a copy of the relevant sections of the manuscript file, the log file, and if possible a sample of any output generated. It would be even better if you could break your example down into a short file that demonstrates the error on a single page.

To contact us send e-mail to `texsis@lifshitz.ph.utexas.edu` or:

U.S.Mail:	Eric Myers Department of Physics Box 539 Vassar College Poughkeepsie, NY 12601 USA	Frank E. Paige Physics Department Box 510A Brookhaven National Laboratory Upton, NY 11973 USA
Internet Electronic Mail:	myers@vassar.edu	paige@bnlux1.bnl.gov
Bitnet Electronic Mail:	ERMYERS@VAXSAR	PAIGE@BNLCL1

TEX is a trademark of the American Mathematical Society.

UNIX is a trademark of AT&T Bell Laboratories.

Copyright © 1990, 1991, 1992, 1994 by Eric A. Myers and Frank E. Paige.
Permission is granted to use this software by any member the academic community in conjunction with basic research, but this software may not be used or distributed for profit without the explicit permission of the authors.

TEXsis is distributed WITHOUT ANY WARRANTY; without even the implied warranty of merchantability or fitness for a particular purpose.

1. Overview

TEX is a computer typesetting language developed by Donald Knuth at Stanford University and described in his book, *The TEXbook*.[1] The great strength of TEX is its ability to typeset technical material and mathematical equations using a syntax based on content; TEX automatically handles the actual placement of the characters on the page. TEX also has a powerful macro capability, giving it great flexibility. This has been exploited by many authors to write macro packages, or "formats," providing features not contained in the basic language (which is sometimes referred to as "plain" TEX.)

TEXsis is a TEX macro package which is designed primarily for typesetting physics papers and preprints, conference proceedings, and related documents. It has been developed from the TechRpt format described by W. Groppe[2] and from TEXsis 1.01, the TEX thesis format by E. Myers.[3] The general scope of TEXsis is similar to the SLAC format called PHYZZX[4], but it offers some significant advantages, such as a more powerful and flexible way of referring to references, equations, and other "numbered" objects. It is functionally a superset of plain TEX, so that everything described in *The TEXbook* will work in TEXsis (although a few of the plain TEX macros have actually been modified). We should mention that LATEX,[5] , another popular TEX-based macro package, is not compatible with plain TEX, so it is not possible to use TEXsis with LATEX or vice versa. Since the REVTEX macro package from the American Physical Society is based on LATEX, it is also incompatible with both TEXsis and plain TEX.

This document explains how to use TEXsis. Since it is a user's guide, it makes no attempt to describe how the various macros work; for this information one should consult the source code, which is extensively commented. However, we do assume that the reader is already familiar with plain TEX. New users should therefore first see either *The TEXbook*, especially chapters 1-10 and 16-19, or *A Gentle Introduction to TEX* by Michael Doob,[6] or the manual by Spivak[7] which accompanies PCTEX (which is an implementation of TEX for the IBM PC).

The main topics covered in the sections of this manual are:

- Section 2 – Getting Started. How to run and initialize TEXsis, and the "master file" concept.
- Section 3 – Selecting sizes and fonts of type, choosing single, double, or triple spacing, and automatically handling quotation marks.
- Section 4 – Automatic numbering of equations, automatic sizing of parentheses and brackets, and special symbols for physics equations.
- Section 5 – Defining new references, for which numbers are assigned automatically, citing references already so defined, and making a list of references.
- Section 6 – Labeling things like section numbers, figures, items in an outline, or page numbers, and referring to them in other parts of the document.

- Section 7 – Producing a complete paper, including the title page, chapters, sections, subsections, and appendices, a table of contents, running headlines, and an index.

- Section 8 – Displaying text in special ways using "environments," including those for centered lines, lists of things, listings of computer programs, and TEX examples.

- Section 9 – Producing automatically numbered tables and figures and producing ruled tables.

- Section 10 – Using specialized layouts for journals and conference proceedings, and creating new ones.

- Section 11 – Typing letters, form letters, memos and referee reports.

- Section 12 – Producing double column output.

- Section 13 – Miscellaneous macros for ignoring text and doing various other useful things, and a checkpoint/restart feature for printing very long documents.

Several appendices are also included to deal with more technical matters. Appendix A explains how to install TEXsis on Unix computers, on a VAX/VMS system, and on an IBM PC. It also explains how to customize TEXsis for your own installation. Appendix B lists changes that have been made to TEXsis in recent revisions. Appendix C lists the fonts usually available with TEXsis.

Two examples of TEXsis documents are included in the standard distribution. The first, in the file `Example.tex`, is a short (3 pages) example of a physics paper typeset with TEXsis. The second is this manual itself, which was also created with TEXsis. If you would like to look at the files used to create the Manual, the master file is called `TXSdoc.doc`, and the various sections of the document are in files with names of the form `TXS`*xxxx*`.doc`. For those who are comfortable programming in TEX it is possible to extract specific macros from the TEXsis source code or to modify or even rewrite these macros. The source code is contained in files with names of the form `TXS`*xxxx*`.tex`. We have tried wherever possible to include at least somewhat illuminating comments in the source code.

1.1. Some Philosophy

In designing the TEXsis macros we have tried to follow some basic guidelines which we feel make TEXsis a more useful tool. It may help you while learning about TEXsis to know what these guidelines are, especially if you are already familiar with other TEX macro packages, though if you like you can skip this section without missing anything. The guidlines we try to follow are:

- Most importantly, TEXsis is a superset of plain TEX, so that anything you can't do in TEXsis (and even those that you can do) can be done with plain TEX instead. This also means that TEXsis is compatible with most plain TEX macro packages, such as `epsf.tex` and `index.tex`, or other specialized macro sets.† Even if some of the plain macros are actually redefine in TEXsis they are required to still behave as described in *The TEXbook*.
- When a TEXsis macro is similar to a plain TEX macro it should, when possible, have a similar name and syntax. Thus `\Footnote` in TEXsis is used exactly like `\footnote` in plain TEX, and in ruled tables "`&`" and "`\cr`" can be used exactly as they are in an `\halign`.
- Macro names will be short and simple but descriptive. Almost all plain TEX macro names are in lower case for ease of typing, so macros with names containing uppercase letters can immediately be recognized as being a part of TEXsis (instead of plain TEX). As a general rule though we also try to use lowercase for frequently used macro names for ease of typing.
- There are two different ways in which arguments may be passed to macros, depending on the length of the argument. Short arguments are simply enclosed in curly brackets, the way TEX usually handles macro arguments. For example, the title of a section is created by the `\section` command with,

  ```
  \section{The nonlinear $\sigma$-model}
  ```

  Longer amounts of text (many lines), and text which is considered to be enclosed in special environments, is enclosed between commands of the form "`\thing`" and "`\endthing`." An abstract, for example, is typed as,

  ```
  \abstract
  A mechanism for total confinement of quarks, similar to that
  of Schwinger, is defined which requires the existence of
  Abelian or non-Abelian gauge fields.  It is shown how ...
  \endabstract
  ```

  This is similar to the `\begin{thing}` ... `\end{thing}` construction in LATEX but is slightly easier to type.* The reason for preferring this way of delimiting text or arguments is that for very long pieces of material it's much easier to look for something like `\thing` and `\endthing` than it is to count curly brackets.

† We cannot completely guarantee this because other macro packages might define macros with the same names as used by TEXsis, although we have taken several steps to prevent this kind of conflict.

* And, more importantly, is compatible with plain TEX, since `\end` signals the end of a document in plain TEX. However, if you really want to type `\begin{thing}` and `\end{thing}` you can do so, and TEXsis will understand it to mean `\thing` and `\endthing`.

- The fundamental purpose of TEXsis is to typeset printed words and equations on a physical piece of paper. In other "document markup" languages, including SMGL and LATEX, the primary goal is to create in the computer an abstract representation of a document which is independent of the way the document is printed. It can be useful to treat a document this way, and so we sometimes use simliar ideas in TEXsis, but this approach can also conflict with the goal of typesetting the document the way an author or editor would choose. Whenever there is such a conflict we will err on the side of "typesetting" rather than on the "abstract representation" of the document.
- The source code defining the macros should be available to the users, so that those who want to see how something is done can do so, and those who want to make their own modifications can do so. The macro definitions should include sufficient comments that (usually) someone with only a little bit of experience with TEX can understand what is going on. Also, macros which perform similar functions and work together should be kept in a single file, but separate parts of the whole package should be kept in separate files. When possible these individual files should be usable on their own with plain TEX. For example, it's possible to use just the double column macros of TEXsis by saying `\input TXSdcol.tex`, and the ruled table macros can be used in plain TEX by saying `\input TXSruled.tex`.

2. Getting Started

Assuming that TEXsis has been properly installed on your computer, using TEXsis is as simple as using plain TEX. To run TEX with the TEXsis format to process a manuscript file called `myfile.tex` you simply give the command:

```
texsis myfile
```

As usual, the output from TEX is put in the file `myfile.dvi`, which you may print on a laser printer or other output device.

Since TEXsis is a superset of plain TEX, any file that you process with plain TEX can be processed with TEXsis, and you will get the same results. In particular, plain TEX defaults to 10 pt. type and single spacing between lines, and so does TEXsis unless you initialize its special features. But since you are probably using TEXsis because of these special features, you should begin your manuscript file with the initialization command

```
\texsis
```

One of the things this command does is switch the output to 12 pt. type, double spaced, which generally looks better on a small laser printer. Commands that allow you to change the type size or line spacing yourself are described in Section 3. More specialized initialization commands may be used in place of (or in addition to) the `\texsis` command when you are using a pre-defined document layout. These are described in Section 10.

Note also that just as in plain TEX there is a difference between uppercase and lowercase letters in control word names. Thus if you say `\TeXsis` instead of `\texsis` you won't initialize any special features but you will print out "TEXsis."

2.1. The Master File

An important and useful concept for typesetting a paper or any moderately long document is that of the *Master File*. The idea is to create one small file which initializes TEXsis with `\texsis`, makes any special definitions or sets parameters, and then reads in the actual text of the document in sections using the `\input` command. As a simple example consider a manuscript called MUONS. The master file is called `MUONS.tex`, and contains the following:

```
\texsis              % initialize TeXsis
\input MyMacros      % special macros for this paper
\input MUONSr        % references for the paper
\input MUONS0        % title page for the paper
\input MUONS1        % first section of manuscript
\input MUONS2        % second section of manuscript
\ListReferences      % print the reference list
\end
```

The file `MyMacros.tex` contains any special macro definitions used in this paper. (If there are none, leave this line out.) The file `MUONSr.tex` contains the references used in this paper, beginning with `\referencelist` and ending with `\endreferencelist`, as described in Section 5. The file `MUONS0.tex` contains the commands for producing the title page of the paper, as described in Section 10. The actual text of the manuscript is divided

into two parts, which are kept in the files `MUONS1.tex` and `MUONS2.tex`. To process the entire document one simply runs TEXsis on the master file with the command

```
texsis MUONS
```

If you are familiar with computer programming you may think of the master file as the main program and the other files as subroutines called by the main program.

One of the great advantages of using a master file, aside from keeping the manuscript organized, is that it lets you work on one section of the paper at a time. For example, suppose you want to run off a draft of just the first section of the paper called MUONS above. You don't need to process the references, the title page, or the second section of the paper. Recall that the percent sign (`%`) is the comment character in TEX — anything appearing on a line after a `%` is ignored by TEX. Thus we simply edit the master file and put a `%` in front of certain lines, like this:

```
\texsis                 % initialize TeXsis
\draft                  % optional draft mode
\input MyMacros         % special macros for this paper
% \input MUONSr         % references for the paper
% \input MUONS0         % title page for the paper
\input MUONS1           % first section of manuscript
% \input MUONS2         % section section of manuscript
% \ListReferences       % print the references
\end
```

Now when you run TEXsis on this file only the first section of the paper will be processed, but it will use the macros defined in `MyMacros.tex`, as well as any other definitions that you might add to the master file. Once the first section is error free you can repeat the process on the second section of the paper, without rerunning the first section. After both sections are correct you can reinstate all lines in the master file and process the whole document.

3. Type Sizes, Type Styles, and Line Spacing

Plain TEX is set up to use 10 pt type, with smaller sizes for superscripts and subscripts. Unfortunately 10 pt. type is too small to print clearly on most small laser printers. TEXsis expands the available type sizes to include 11, 12, 14, and 16 pt type, and TEXsis makes it easy to change from one size of type to another. Type sizes are selected with the following commands:

`\ninepoint`	Gives 9 pt type (not illustrated).
`\tenpoint`	Gives 10 pt type, like this.
`\elevenpoint`	Gives 11 pt type. (not illustrated).
`\twelvepoint`	Gives 12 pt type, like this.
`\fourteenpoint`	Gives 14 pt type, like this.
`\sixteenpoint`	Gives 16 pt type, like this.
`\twentypoint`	Gives 20 pt type (not illustrated).
`\twentyfourpoint`	Gives 24 pt type (not illustrated).

In many implementations of TEX there is insufficient room to load all of these fonts, so fonts are loaded only when they are first used, and not all of them can be illustrated here. TEXsis starts out with `\tenpoint`, just like Plain TEX, but switches to `\twelvepoint` once you say `\texsis`.

In Plain TEX you can switch between Roman, Bold, or Italic typestyles with the commands `\rm`, `\bf`, and `\it`. The same is true with TEXsis. These commands change the style of the type, but not the size. The complete set of commands to change typestyles is:

`\rm`	Gives Roman type. Available in all sizes.
`\it`	Gives *italic type* for emphasis. Available in all sizes.
`\sl`	Gives *slanted type* for book titles, etc. Available in all sizes.
`\bf`	Gives **bold face type** for emphasis. Available in all sizes.
`\tt`	Gives `typewriter type` for listing computer programs, TEX examples, etc. Available in 10, 11, and 12 pt.
`\mib`	Gives math italic bold in math mode (`$...$`) for titles. Available in 10, 11, 12, 14, 16, 20, and 24 pt base sizes, but the minimum size for subscripts and superscripts is 10 pt.
`\scr`	Gives script capital letters (using Ralph Smith's Formal Script fonts, the `rsfs` fonts, if they are available on your computer). Use `\scr` as you would use `\cal`.

TEXsis includes the mathematics fonts in all sizes; these are used automatically when you type equations between the math quotes, `$...$`. Limited sans-serif fonts are also available using the commands `\tenss` and `\twelvess`.

Two special symbols, `\Tbf` and `\tbf`, are used to produce "Title bold face" and "small title bold face." The first is used to print the title of a paper and titles of the chapters of a paper, and the second is used to print the titles of sections of a paper. You are free to change the definitions of these control words, but if you do not they default to 14 pt boldface for `\Tbf` and 12 pt boldface for `\tbf`.

Math italic bold (`\mib`) is listed with the other fonts but is actually different: it substitutes boldfaced versions of the math italic and math symbol fonts for the normal ones *everywhere* in math mode (`$...$`). These fonts are available only with with Computer Modern fonts and only in 10 pt and larger sizes; the standard math fonts are used for smaller superscripts. Since `\mib` effectively redefines `$..$`, it must be placed before the math material (i.e., outside the math quotes) and the whole formula will be bold. For example,

```
{\tbf\mib Observation of $Z^0 \to \mu^+\mu^-$}
```

will produce the indicated text in `\tbf` size with a bold formula, provided you have the right fonts. The easiest way to find out is to try it. Since the `\mib` fonts are not available everywhere, they are never loaded or used unless you explicitly ask for them.

Plain TEX includes a calligraphic script font known as `\cal`, but the letters in this font are not quite what most physicists are accustomed to using for things like Lagrangian densities or functional measures. Ralph Smith of U.C. San Diego has created a new "formal script" font with much better letters for such purposes, and TEXsis supports this font provided that it is available on your computer. To get this new font you simply use `\scr` in place of `\cal`. An example is:

```
$$
   {\scr L} = - \del^\mu \phi^\ast \del_\mu \phi + m^2 \phi^2
$$
```

Unfortunately we don't know if your computer has the `rsfs` fonts so we won't try to show you the output from this example. The quickest way to determine whether the fonts are available on your system is to try this example and verify that TEX does not complain about the font. If your system does not have the font we have made it available via anonymous ftp from lifshitz.ph.utexas.edu in the directory `texsis/rsfs`, both in the form of "pk" font files (which are binary files) and in the form of MetaFont source files.

3.1. Interline Spacing

When you use the macros `\tenpoint`, `\twelvepoint`, etc., the style of the type is reset to Roman, as if you'd also typed `\rm`. The baseline skip (the distance between lines) is also set to be suitable for the type size. Several commands are provided that vary the interline spacing, both in the text and in auxiliary material such as tables. These are:

`\doublespaced` makes the baseline skip 1.5 times its normal value. True double spacing looks too wide with typesetting.

`\singlespaced` resets the baseline skip to its normal value.

`\TrueDoubleSpacing` makes the baselineskip twice its normal value. This usually looks too wide for a final document, but it is desirable in a manuscript being submitted to journals such as Physical Review.

`\triplespaced` makes the baseline skip 3 times its normal value. This should only be used for drafts of papers to leave room for notes or corrections.

`\widenspacing` increases the current spacing by a factor of 1.25. With single spacing this is the analog of $1\frac{1}{2}$ spacing on a typewriter. Repeating `\widenspacing` increases the spacing by 1.25 times whatever it already was; it does not reset the spacing to singlespaced first.

When TEXsis is first started the default is `\singlespaced`, but once TEXsis is initialized with the `\texsis` command (or `\preprint` or `\paper`) the spacing is set to `\doublespaced`.

3.2. Footnotes

Even when a paper is double spaced, you may want to have the footnotes single spaced. Also, if the text is in 12 pt type you may nevertheless want to have the footnotes in 10 pt type. In TEXsis you can do this easily using `\Footnote` in place of `\footnote`, as in this example:†

```
... as in this example:\Footnote\dag{This is an example of
the use of {\tt\bs Footnote} in \TeXsis.}
```

`\Footnote` also automatically puts the reference mark (in this case the dagger symbol `\dag`) in a superscript.* The typestyle used for the text of the footnote is controlled by `\FootFont`, which defaults to:

† This is an example of the use of `\Footnote` in TEXsis.

* The footnote character is typeset in math mode, as if it were between two `$`'s.

```
\def\FootFont{\tenpoint}
```

If you want to change this you can do so somewhere near the beginning of your document. The spacing between multiple paragraphs in a `\Footnote` (if you are prone to being so verbose) is set to zero, regardless of the `\parskip` in the main part of the document, but you can also change this by adding `\parskip=`⟨amount⟩ to your new definition of `\FootFont`. If you want a singlespaced footnote in the `\FootFont` typestyle but without putting a reference mark in the text you can use `\Vfootnote` just as you would use `\vfootnote` in Plain TEX. And of course if `\Footnote` or `\Vfootnote` cannot do what you need even after changing `\FootFont` then you can always use `\footnote` or `\vfootnote` from Plain TEX.‡

Sometimes it's desirable to have footnotes numbered consecutively. This is easily done with `\NFootnote`, which keeps a count of the footnotes it has created and uses the numbers for the reference marks. Just as with `\Footnote`, the text is single spaced and printed in the `\FootFont` typestyle.[1] The footnote counter is called `\footnum`, so if you want to start over numbering footnotes (at the begining of a new chapter, for example), then simply set this back to zero, by saying

```
\footnum=0
```

3.3. Automatic Quote Balancing

As you may recall from *The TEXbook*, one of the differences between typing and typesetting is the way in which quote marks are handled. On a typewriter you would simply use the non-oriented double-quote mark (") both to begin and end the quoted material. In typesetting, however, quoted text begins with a left-quote mark (“) and ends with a right-quote mark (”). To produce these in TEX you have to type two single-quote marks, either ‘‘ or ’’. This is not very difficult to do, but it is easy to forget and use the double-quote marks instead. With TEXsis that's okay. If you say

```
\quoteon
```

then the next time a double-quote mark comes along it will be translated into a left-quote mark. The double-quote mark after that will become a right-quote mark, and so on, so that material enclosed in double-quote marks will be typeset correctly. All you have to do is remember to use the double-quote marks in pairs.

In some special cases you may not want TEXsis to be so smart about double-quotes. You can turn this feature off by saying

```
\quoteoff
```

The default though, once you have said `\texsis`, is `\quoteon`.

---

‡ A technical footnote: With `\footnote` from plain TEX the group is closed before the footnote is typeset, so any line spacing you set inside the footnote is lost. You can solve this by putting a `\par` or `\vskip 0pt` at the end of your footnote, but then the final strut is put on a separate line. Getting footnotes to work just right is tricky business, which is why we wrote `\Footnote`.

[1] This is an example of a numbered footnote, which was created by saying “`...printed in the {\tt\bs FootFont} typestyle.\NFootnote{This is an example ... }`”

4. Equations

Plain TEX was designed to make it easy to typeset equations and mathematical formulae. TEXsis provides additional features that make the job even easier. Among these are the automatic numbering of equations (with the ability to refer to numbered equations by symbolic labels), automatic sizing of parentheses, and a collection of special symbols which are often used by physicists.

4.1. Equation Numbering

In Plain TEX equation numbers are added to displayed equations with the `\eqno` command, as described in Chapter 19 of *The TEXbook*. For example, typing

```
$$ E=mc^2.  \eqno (47) $$
```

will produce

$$E = mc^2 \tag{47}$$

With TEXsis the procedure for getting automatically numbered equations is similar, except that you use `\EQN` instead of `\eqno`, and what follows this (up to the "`$$`") is not used as the number of the equation, but as an internal *label* for the equation. For example, if you had already used `\EQN` to create three numbered equations, then typing

```
$$ \vec F = -{G_N Mm \over r^2} \, \hat r
   \EQN Newton $$
```

would produce

$$\vec{F} = -\frac{G_N M m}{r^2}\,\hat{r} \tag{4.4}$$

TEXsis automatically assigns the next available equation number to the equation and prints this number on the right, in parentheses. The equations that follow the one above would be numbered 4.5, 4.6, and so on.

The purpose of the "label" after the `\EQN` is to let you refer to the equation in the text without knowing the equation number that has been assigned to it. For example, to refer to Eq. (4.4) above, which has the label "Newton", you might type

```
The dependence of the force on the inverse square of the distance
can be seen in \Eq{Newton}.
```

This would be printed as:

> The dependence of the force on the inverse square of the distance can be seen in Eq. (4.4).

The control sequence `\Eq{`⟨label⟩`}` gives the same result as typing "`Eq.~(`*nn*`)`", where *nn* is the appropriate equation number for the given label. Two similar control words allow you to refer to a range of equations, or to refer to an equation simply by number, without the preceding "`Eq.~`." Typing `\Eqs{`⟨label⟩`}` produces "Eqs. (*nn*)", while `\Ep{`⟨label⟩`}` gives just the equation number associated with the label, enclosed in parentheses. Thus typing

```
The class of metrics defined by \Eqs{cond-1} through
\Ep{cond-5} can all be shown to satisfy inequality
\Ep{triangle}, as well as the condition in \Eq{Killing}.
```

would produce output something like

> The class of metrics defined by Eqs. (14) through (19) can all be shown to satisfy inequality (23), as well as the condition in Eq. (7).

Almost any sequence of characters can be used as a "label" for an equation. This includes letters, numbers, and punctuation marks, but it is best to avoid TEX's special characters, such as `\`, `&`, `}`, `{`, `$`, `_`, and `^`. Spaces may be included but will be ignored, and there *is* a difference between uppercase and lowercase letters. Also, the semicolon (`;`) is reserved for a special purpose, which is described below.

You may want to give equations "names," as in the example called "Newton" above, or you can give the equations labels that are numbers. This is the best thing to do if the equations in the original manuscript already have numbers, but if you do this then you should be sure to remember that the internal label and the number which is printed (the "external" number) are not necessarily the same. If the equation *labeled* "15" is the fifteenth equation in the document then it will also be equation number 15, but if you move the equation around in the file so that it is the sixth equation in the document it will be printed as equation 6, even though it has the label "15".

If you have a heavily edited document and the arrangement of the numbers and labels gets confusing there is a useful aid to help straighten things out. If you say

```
\eqnotracetrue
```

then the equation numbers will be printed as usual, but with the label appearing next to it in the right margin in square brackets. For example, the equation

```
$$
(i\gamma^\mu \partial_\mu - m)\psi (x) = 0
   \EQN Dirac $$
```

when printed with `\eqnotracetrue` would appear as:

$$(i\gamma^\mu \partial_\mu - m)\psi(x) = 0 \tag{4.5}$$ ["Dirac"]

Since the label appears in the margin the spacing of your document remains exactly the same. Obviously you would not print the final copy of the document this way, but it can be very useful for editing. You can turn off this equation tracing feature anytime by saying `\eqnotracefalse`. If you say `\draft` at the beginning of your document then the equation tracing will be turned on automatically (along with some other useful features).

Sometimes when several equations are closely related you will want to give the whole set the same equation number. To distinguish separate formulae in such a group you then generally add a letter to the number. You might, for example, have the equation

$$J^2|j,m\rangle = j(j+1)|j,m\rangle \tag{4.6a}$$

and then a short while later you might have the related equation

$$J_z|j,m\rangle = m|j,m\rangle \tag{4.6b}$$

These may be referred to together as Eqs. (4.6) or may be referred to separately as Eqs. (4.6a) and (4.6b). To get an equation number with a letter in it you still use `\EQN`, but you add the appropriate letter to the end of the label, separated from the label by a semicolon. The equations above were created by typing:

```
$$
J^2\ket{j,m} = j(j+1) \ket{j,m} \EQN eigen;a
$$
```

and

```
$$
J_z\ket{j,m} = m \ket{j,m} \EQN eigen;b
$$
```

The semicolon is not printed — it is just used to separate the letter from the label. If the letter is "a" then the next available equation number is used, as usual, but if the letter is anything else then the equation number is not incremented and instead the number from the previous equation is used. You can use any sequence of letters you like, although it is usually best to use "a,b,c..." for consecutive equations.

To refer to equations that have both numbers and letters you can either use the letter to pick out a specific equation, or simply leave out the letter and use the label to refer to the whole set together. You can use either `\Eq` or `\Eqs` to do this. For example, the sentence after Eq. (4.6b) above was created by typing

```
These may be referred to together as \Eqs{eigen}
or may be referred to separately as
\Eqs{eigen;a} and \Ep{eigen;b}.
```

You will eventually encounter an equation that is so long that it has to be broken up into several lines, or you will want to print several equations together, lined up by their "=" signs. In plain TEX you would use `\eqalign` to display these kinds of equations, or `\eqalignno` to display them with equation numbers. With TEXsis you should use `\EQNalign` in place of `\eqalign` or `\eqalignno` to get automatically assigned equation numbers in aligned equations. As in plain TEX, the alignment of the equations is controlled by `&` and `\cr`. An equation number is assigned to a particular line by putting `\EQN` followed by a label on that line, before the `\cr`. The label is everything from the `\EQN` up to the `\cr`. For example,

```
$$
\EQNalign{
{dp \over dt} &= - {\partial H \over \partial q}
\EQN Hamilton-1 \cr
{dq \over dt} &= {\partial H \over \partial p}
\EQN Hamilton-2 \cr}
$$
```

produces the two aligned and numbered equations

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q} \tag{4.7}$$

$$\frac{dq}{dt} = \frac{\partial H}{\partial p} \tag{4.8}$$

`\EQNalign` can be used both for typing multi-part equations or for splitting an equation which is too long to fit on a single line. However, when splitting a single long equation into several lines it may actually be better to use `$$\eqalign{`⟨formula⟩`}` followed by `\EQN` ⟨label⟩`$$`. For example, in

$$\begin{aligned}\mathcal{L}_{NC} = \frac{-g}{2\cos\theta_W} \sum_i \bar{\psi}_i [(t_{3,i} - 2Q_i \sin^2\theta_W)\gamma_\mu - t_{3,i}\gamma_\mu\gamma_5]\psi_i Z^\mu \\ - e \sum_i Q_i \bar{\psi}_i \gamma_\mu \psi_i A^\mu\end{aligned} \tag{4.9}$$

the equation number is centered between the two lines of the equation, which looks better. This equation was produced using `\eqalign` by typing:

```
$$
\eqalign{
{\cal L}_{NC} &= {-g \over 2\cos\theta_W} \sum_i
   \bar\psi_i [ (t_{3,i}-2Q_i\sin^2\theta_W) \gamma_\mu
   - t_{3,i}\gamma_\mu\gamma_5 ]\psi_i Z^\mu \cr
&\quad - e \sum_i Q_i \bar\psi_i \gamma_\mu \psi_i A^\mu \cr}
\EQN{NC}
$$
```

Closely related equations can be displayed together with the same equation number by following the label with a semicolon and a letter, as already described above. For example, the set of equations,

$$\nabla \cdot \vec{D} = 4\pi\rho \tag{4.10a}$$

$$\nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t} \tag{4.10b}$$

$$\nabla \cdot \vec{B} = 0 \tag{4.10c}$$

$$\nabla \times \vec{H} = \frac{4\pi}{c}\vec{J} + \frac{1}{c}\frac{\partial \vec{D}}{\partial t} \tag{4.10d}$$

is produced by typing:

```
$$
\EQNalign{
\nabla \cdot \vec D    &=  4 \pi \rho \EQN EM;a       \cr
\nabla \times \vec E   &=
        - {1 \over c} {\partial \vec B \over \partial t}
        \EQN EM;b                                     \cr
\nabla \cdot \vec B    &=  0 \EQN EM;c                \cr
\nabla \times \vec H   &=  {4 \pi \over c} \vec J
        + {1 \over c} {\partial \vec D \over \partial t}
        \EQN EM;d                                     \cr}
$$
```

Note that the alignment of the equations is controlled by `&` and `\cr`, not by the arrangement used when the equation was typed into the manuscript file. Still, it is recommended that you arrange long sets of equations like these in some sort of clear pattern when you type them. This will both reduce errors when the equation is typed and make it easier for you to read the file when you are looking at it later. Remember, TEX ignores all spaces and tabs when it reads material for equations or displayed equations.

`\EQNdoublealign` is similar to `\EQNalign` but makes equations with two alignment tabs. For example,

```
$$
\EQNdoublealign{
\sigma_i^2 &= 1 & \EQN Pauli;a \cr
\sigma_i\sigma_j + \sigma_j\sigma_i &= 0 \qquad & (i \ne j)
  \EQN Pauli;b \cr
\sigma_i\sigma_j &= i\sigma_k \qquad & (i,j,k\ {\rm cyclic})
   \EQN Pauli;c \cr}
$$
```

produces

$$\sigma_i^2 = 1 \tag{4.11a}$$

$$\sigma_i\sigma_j + \sigma_j\sigma_i = 0 \qquad (i \ne j) \tag{4.11b}$$

$$\sigma_i\sigma_j = i\sigma_k \qquad (i, j, k \text{ cyclic}) \tag{4.11c}$$

with both the equal signs and the parentheses aligned. The second `&` is required in the first line, because without it the `\EQN` would have aligned the equation number with the parentheses enclosing $i \ne j$.

Plain TEX also lets you display formulae on several lines without any alignment by using `\displaylines`. With TEXsis you can use the similar control sequence `\EQNdisplaylines` to get an automatically assigned equation number for any line in such a display. The pattern is the same as in `\displaylines`, but you simply insert `\EQN`⟨label⟩ in front of the `\cr` on the line you want numbered. `\EQNdisplaylines` produces less space between the equations than would result from typing them separately, and generally looks better for closely related equations.

All of the equation alignment macros are special cases of the plain TEX macro `\halign`. Therefore, if necessary, the vertical spacing can be adjusted by inserting

`\noalign\vskip`⟨dimen⟩

where ⟨dimen⟩ can be either a positive or a negative dimension.

In a long document there are several different ways to number the equations. You could just start at "1" and count up for each new equation, but sometimes it is better to number the equations consecutively within each chapter or section of a document. In TEXsis the default form for an equation number is *cc.ss.nn* where *cc* stands for the chapter number, *ss* is the section number, and *nn* is the equation number within that section. Whenever you move to a new section or chapter (using `\section` or `\chapter`, as described in Section 7.3) the number *ss* or *cc* is incremented by one and the number *nn* is reset to one. If you are not

using chapter divisions in your document then the chapter number *cc* is not made a part of the equation number, and the number *nn* is reset only at the beginning of a section. If you are not using section divisions in your document then the section number *ss* is not made a part of the equation number, and the number *nn* is reset to one only at the beginning of a chapter. If you use neither chapter nor section divisions in your document then neither the chapter number *cc* nor the section number *ss* will be made a part of the equation number, and the equations will simply be numbered consecutively, 1, 2, 3....

Although the system just described for forming equation numbers gives a result that is usually what you will want for your equation numbering, you can change the method by setting either of two switches. If you say `\showchaptIDfalse` then the chapter number *cc* will not be made a part of the equation number even though you may be using chapter level divisions in your document. Similarly, saying `\showsectIDfalse` prevents the section number *ss* from being used in the equation number. The opposites of these two switches are `\showchaptIDtrue` and `\showsectIDtrue`. The equation number *nn* is reset to one only when a section or chapter number that is supposed to be visible is incremented, as you might expect.

4.2. Automatic Parentheses Sizing

Plain TEX provides automatic adjustment of the sizes of parentheses and other delimiters when you use `\left` and `\right`, as in

```
$$ \left( ⟨math material⟩ \right) $$
```

TEXsis provides similar automatic sizing of parentheses and square brackets in displayed equations without the need of typing `\left` and `\right`. You must still use `\left` and `\right` for all other delimiters, and they can be used for parentheses and square brackets if desired. This feature is turned on and off with `\autoparens` and `\offparens` respectively; the default, once you say `\texsis`, is `\autoparens`. Thus, for example,

```
\autoparens
 $$
( a \over b )
 $$
```

produces

$$\left(\frac{a}{b}\right)$$

whereas

```
\offparens
 $$
( a \over b )
 $$
```

produces

$$\frac{(a}{b)}$$

which is not what one normally desires.

The automatic sizing of parentheses works by making `)`, `(`, `]`, and `[` active characters in displayed equations and defining them to be macros which use `\left` and `\right` to size the parentheses. While this normally works properly and simplifies the typing of equations, it will fail if the `\left` and `\right` operators are not properly balanced, such as when a long equation is split between two or more lines with `\eqalign` or `\EQNalign`. Almost all such problems can be solved by using the control sequences `\lparen`, `\rparen`, `\lbrack`, and `\rbrack`, which are taken from Plain TEX and which give just the standard parenthesis and bracket characters. For example,

$$F = \left[\left(\frac{a}{b}\right) + \left(\frac{c}{d}\right) + \left(\frac{e}{f}\right) + \left(\frac{g}{h}\right)\right]$$

can be produced with

```
$$
\EQNalign{
F &= \Biggl\lbrack (a \over b) + (c \over d) \cr
&\quad + (e \over f) + (g \over h) \Biggr\rbrack \cr}
$$
```

Note that in this example the parentheses are balanced on one line and so can be sized automatically, while the brackets span two lines and so must be sized "by hand." The `\vbig` macro described in the next section provides additional flexibility for sizing.

If all else fails, you can use `\offparens` to turn off the automatic sizing of parentheses and put in `\left` and `\right` or explicit sizes yourself. In any case, you must put `\left` and `\right`, if needed, in in-line equations (math expressions which are not displayed equations), because. `\autoparens` only applies to displayed equations, not in-line math text.†

4.3. Special Symbols

TEXsis defines some special symbols and other constructions which are not included in plain TEX and which either are commonly used by physicists or are not trivial to construct. The following macros, which are listed here by related groups, provide these additional special symbols for use in equations:

- `\bra{`⟨text⟩`}` and `\ket{`⟨text⟩`}` produce the notations $\langle text|$ and $|text\rangle$ respectively for quantum mechanical states. `\vev{`⟨text⟩`}` produces $\langle text\rangle$ for a VEV (Vacuum Expectation Value).

† Previous versions of TEXsis used the control sequence `\onparens` to provide automatic parentheses sizing in *all* math-mode expressions. While you can still use this, we do not encourage it since it may cause problems with other uses of these characters.

- `\del`, `\grad`, and `\square` produce the notations ∂, ∇, and $\Box$ for various derivatives. (The first two are just synonyms for `\partial` and `\nabla`.)
- `\degrees` is the degrees symbol. For example, `90\degrees` produces "90°". You can use `\degree` as a synonym. These work equally well in math or text mode.
- `\frac{`⟨numerator⟩`}{`⟨denominator⟩`}` produces the fraction `{`⟨numerator⟩ `\over` ⟨denominator⟩`}`, while `\smallfrac{`⟨numerator⟩`}{`⟨denominator⟩`}` produces the same fraction in the smaller subscript style type. `\half` produces a $\frac{1}{2}$ in a size appropriate to its usage.
- `\simle` and `\simge` make the symbols $\lesssim$ and $\gtrsim$ to indicate "less than about" and "greater than about," respectively. Synonyms for these are `\ltsim` and `\gtsim`.
- `\parenbar` puts a bar in parentheses above a character; this is often used to indicate either a particle or its antiparticle. `\nunubar` and `\ppbar` are special cases. Thus

  ```
  $$ n \parenbar{n} \qquad \nu \nunubar \qquad p \ppbar $$
  ```

  gives

  $$n\overset{(-)}{n} \qquad \nu\overset{(-)}{\nu} \qquad p\overset{(-)}{p}$$

- `\subrightarrow{`⟨text⟩`}` puts the text under a right arrow of the appropriate length. Thus

  ```
  $$ f(x) \subrightarrow{x \to 0} f(0) $$
  ```

  gives

  $$f(x) \underset{x\to 0}{\longrightarrow} f(0)$$

- `\buildchar{`⟨#1⟩`}{`⟨#2⟩`}{`⟨#3⟩`}` gives character `#1` with `#2` above it and `#3` below it, allowing the construction of a variety of special symbols. `\overcirc` is a special case of this which puts a small circle above a character, as in $\mathring{R}$. This is often used in general relativity and related fields.
- `\vbigl{`⟨delim⟩`}{`⟨size⟩`}` and `\vbigr{`⟨delim⟩`}{`⟨size⟩`}` produce very (or variably) big delimiters, generalizing the Plain TEX macros `\bigl`, `\Bigr`, etc. They produce, respectively, the left and right delimiters ⟨delim⟩ with height ⟨size⟩, which can be given in pt., cm., in., or any other valid units. Any valid delimiter may be used, but you must be careful if you use parentheses and automatic parenthesis sizing is also in effect (Section 4.2), because these conflict. You must either turn off automatic parenthesis sizing (with `\offparens`) or use `\lparen` or `\rparen` in place of '`(`' and '`)`'.
- `\Leftcases{`⟨size⟩`}` and `\Rightcases{`⟨size⟩`}` give left and right braces of size ⟨size⟩ smashed to zero height; these can be used in place of the `\cases` of Plain TEX when it is desired to put an equation number on each line. For example,

  ```
  $$\EQNalign{
  m_e    &= ~.511~\MeV \EQN lep;a \cr
  m_\mu  &= ~105~\MeV
            \quad \Rightcases{50pt} \quad {\rm Leptons}
            \EQN lep;b \cr
  m_\tau &= 1784~\MeV \EQN lep;c \cr}
  $$
  ```

gives

$$\left.\begin{aligned} m_e &= .511 \text{ MeV} \quad (4.12a)\\ m_\mu &= 105 \text{ MeV} \quad (4.12b)\\ m_\tau &= 1784 \text{ MeV} \quad (4.12c)\end{aligned}\right\} \text{ Leptons}$$

Note that `\Leftcases` and `\Rightcases` do not provide automatic spacing, which must be adjusted by the user as in the previous example. There seems to be no simple way to obtain automatic spacing and to allow an equation number on each line.

- `\sterling` produces the British currency mark £.
- `\dimensions{`⟨units⟩`}{`⟨power⟩`}` is used for dimensional analysis. The units appear in square brackets, raised to the given power. For example to produce $[\text{mass}]^{-3}$ you would type `\dimensions{mass}{-3}`.
- `\slashchar{`⟨letter⟩`}` puts a slash through a character, the notation for the contraction of a vector with Dirac γ matrices in Feynman graphs. For example

```
$$ {\cal L} =
    \bar\psi[i\slashchar{\partial} - e\slashchar{A}-m]\psi $$
```

gives

$$\mathcal{L} = \bar{\psi}\,[i\partial\!\!\!/ - eA\!\!\!/ - m]\,\psi\,.$$

The results produced by `\slashchar` are not as good as one would like. You can do somewhat better by adjusting the kerning of the `/` by hand for each character. For example,

```
\def\Aslash{A\kern -0.47em/}
```

produces "$A\!\!\!/$". But note that the angle of the slash is still not quite what one would like for a slashed "A." To really get good looking slashed characters someone will have to create a new slashed character math font using MetaFont. Meanwhile you can get by with `\slashchar` or define your own using kerning.

Abbreviations for units are normally set in Roman type even when used in formulas. Special function (e.g., sine, cosine and logarithm) or operator names are also set in Roman type, and TEXsis adds some new names to go with the old standards.

- `\Tr` and `\tr` give "Tr" and "tr" respectively for the trace of an operator or of a matrix.
- `\real` and `\imag` give the symbols Re and Im for the real and imaginary parts of an expression; the spacing is like that for any other function or operator.
- `\eV`, `\KeV`, `\MeV`, `\GeV`, and `\TeV` give the abbreviations for energy units eV, KeV, MeV, GeV, and TeV respectively.
- `\mb`, `\mub`, `\nb`, and `\pb` give the abbreviations for cross section units mb, μb, nb, and pb respectively, while `\cmsec` gives $\text{cm}^{-2}\text{s}^{-1}$, the standard units for luminosity in colliding beam accelerators.

- `\units{`⟨name⟩`}` is a generic macro to print in Roman type the name of any units not covered by the macros above. For example, you can say `\units{furlongs}` and `\units{fortnights}` if you are using that particular system of units.

Note that all of the abbreviations for units above are defined without any spacing; to produce

$$\mathcal{L} = 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}, \qquad p_\perp = 8.3\,\mathrm{GeV}$$

you should type

```
$$
{\cal L} = 10^{33}\,\cmsec, \qquad p_\perp = 8.3\,\GeV
$$
```

with a `\,` to make a thin space between the number and the units.

5. References and Citations

TEXsis includes a set of macros for making citations to the literature in the text of a document. References are processed in the standard format used by many American physics journals: they are automatically assigned consecutive numbers as they appear in the text and are listed in numerical order at the end of the document. In the text a citation appears as a superscript number corresponding to the reference cited, although it is possible to change this so that the reference number appears in-line in the text in square brackets, as is done in some European journals (and now in the Physical Review as well). When each reference is defined it is given a label, which is associated with the number assigned to the reference. Subsequent citations to the same reference are made using this label, so there is no need to know the particular number assigned to the reference; Indeed this number may change as the document goes through editing and revision.

TEXsis is flexible enough that the text of the references may be entered in two different ways, either all references at once at the beginning of the document or each one individually at the point in the text where the work is first cited. In either case, TEXsis stores the text of all of the references in numerical order until it is asked to print the list at the end of the document, by saying `\ListReferences`.

To make a citation to a new reference and to enter the text of the reference, use the control sequence `\reference{`⟨label⟩`}`, where ⟨label⟩ is the label to be used for the number referring to this reference. This is followed by the text of the reference, which is terminated by `\endreference`. An example is:

```
To this end we will follow Feynman\reference{Feynman, 1949}
R.P.~Feynman, Phys.\ Rev.\ \vol{76}, 749 (1949)
\endreference
in developing the propagator approach.
```

This would appear in the text of the document as,

> To this end we will follow Feynman[8] in developing the propagator approach.

and the text of the reference would be saved until the list of references is printed at the end of the document. If you look at the end of this manual you will see it.

There are several things to take note of in the example above. Most importantly, there is no space between the word in the text where the citation is made ("Feynman") and the `\reference`. This is so the superscript reference number will appear in the text right after the word to which the citation is attached, with no intervening space. Note also in the reference text the tie (`~`) between the initials and the name, the control spaces (`\ `) after the abbreviations, and the use of the `\vol` macro for the volume number; `\vol{`⟨number⟩`}` prints the volume ⟨number⟩ with an underscore below it (this is easier to read than bold-faced volume numbers on small laser printers, but if you prefer something else you can simply redefine the `\vol` macro). Similarly, `\booktitle{`⟨title⟩`}` will print the title of a book, in slanted type. Finally, note that there is no punctuation at the end of the reference text. This is because TEXsis automatically supplies the appropriate punctuation when it prints the list of references. Normally this will simply be a period (.), but it is possible to

have multiple references assigned to the same reference number, in which case the references are separated by semi-colons, with the last reference terminated by a period. How to do this is described below.

Since typing the text of a journal citation is somewhat complicated, the `\journal` macro was created to make things a little easier. The format is

`\journal` ⟨journal name⟩;⟨volume⟩,⟨page number⟩(⟨year⟩)

This constructs the journal citation for you, with periods automatically replaced by ".␣" in abbreviations and the volume number underlined for you by `\vol`. It is important that the field separators (the semi-colon, the comma and the parentheses), all be present and in the right order or TEX will look as far forward as it takes to find the next one it wants. You will find that you don't want any spaces between anything.

Besides making it a little easier to type a citation to a journal article, using `\journal` lets you switch from the American form of citation to the European form very easily. In the American form the page number follows the volume, followed by the year in parentheses. In the European form the year follows the volume number and the the page number comes after that. Normally TEXsis uses the American form, but you can change to the European form simply by saying

```
\Eurostyletrue
```

The arguments to `\journal` are always in the American order.

To enter several references with the same reference number simply begin the next `\reference` immediately after the previous `\endreference`, and begin the labels of all but the first reference with an asterisk "*", as in the following example:

```
The vacuum polarization term can be regularized
by introducing a cut-off at high
frequencies.\reference{Pauli and Villars, 1949}
W.~Pauli and F.~Villars, \journal Rev. Mod. Phys.;21,434(1949)
\endreference
\reference{*Schwinger, 1948}
J.~Schwinger, \journal Phys. Rev.;74,1439(1948)
\endreference
```

This produces the single citation

> The vacuum polarization term can be regularized by introducing a cut-off at high frequencies.[9]

The "*" at the beginning of the second label tells TEXsis not to assign a new reference number to this reference, but instead just to add it to the text of the previous one. Of course you could type the text of both references between one `\reference` and `\endreference`. An advantage to following the example above is that the second reference can be given a reference number of its own simply by deleting the "*" at the beginning of the label. Similarly, two references with different reference numbers can be lumped together into one citation by inserting a "*" in front of the label of the second reference. Another advantage to keeping the references separate is that it is easy to move references around, either within

a document that is being revised or between documents. It is therefore recommended that each reference be entered separately, although this is obviously not required.

Instead of putting the `\reference ...\endreference` in the text it may be more convenient to type them all together, especially if they already appear that way in the manuscript. This can be done without producing any unwanted printed numbers by using

```
\referencelist
\reference{1} text 1... \endreference
\reference{2} text 2... \endreference
...
\reference{last} last text... \endreference
\endreferencelist
```

The `\referencelist` must appear at the beginning of the document, so that the reference labels are defined before they are used. This is best put into a separate file which is `\input` by the master file. Of course the list of references is not printed until the end of the document, when `\ListReferences` is called.

If `\reference ...\endreference` is used in the text of the document, then the reference number is printed there. The default format is a superscript (like this[137]). The command `\superrefsfalse` will cause the reference numbers to be printed in square brackets (like this [137]), the style used in most European journals.‡ The `\endreference` macro looks ahead to see if the next token is another `\reference`; if so, then a range of numbers (like this[137–140]) is printed. If a citation appears at the end of a sentence, it looks best if it comes after the punctuation for superscript references but before it for references in brackets. You will have to arrange for that yourself.

All subsequent citations of a previously defined reference are made using its label. If a `\referencelist` is used, then all citations appearing in the text will be made in this way, since then the initial reference definitions print nothing. The construction `\cite{`⟨label⟩`}` prints the reference number corresponding to the label as a superscript (or in square brackets if `\superrefsfalse` is selected.) For example,

```
A good discussion of this is given by Doob\cite{Doob}
in section 7.
```

produces

A good discussion of this is given by Doob[6] in section 7.

since Michael Doob's book *A Gentle Introduction to TEX* is in fact Ref. 6 of this document and has the label "`Doob`." Note that some extra space is inserted after the reference number, just as it usually is at the end of a sentence. If several `\cite`'s are given in succession then the corresponding numbers are printed separated by commas. Unlike `\reference`, which defines the reference numbers, `\cite` does not know if references are sequential, so it cannot replace a sequence of numbers by a range. To avoid a long sequence of reference numbers

‡ Physical Review has also recently adopted this style, but the default behavior of TEXsis is still to use a superscript reference number, at least for this version (2.16).

use instead `\citerange{`⟨first label⟩`}{`⟨last label⟩`}` to print the numbers corresponding to the first and last labels as a range.

The construction `\Ref{`⟨label⟩`}` produces "Ref. *nn*" in the text, with *nn* the number of the reference corresponding to the label. For example, the "Ref. 6" in the previous paragraph was typed into the manuscript tfile as `\Ref{Doob}`. `\Ref` is simply shorthand for `Ref.~\use{Ref.`⟨label⟩`}`, and you can change what is printed simply by changing the definition of `\Ref`. For example, if you said

```
\def\Ref#1{reference~\use{Ref.#1}}
```

then `\Ref{Doob}` would produce "reference 6" instead. For more on how to use `\use` see Section 6.

After the text of the document is complete a printed list of the references can be obtained by saying `\ListReferences`. This prints the numbers at the left hand edge of the page with the text indented by `\refindent`, which is chosen so that the equation numbers line up, and it puts a `\refskip` (initially equal to a `\smallskip`) between references. The reference format can be changed by redefining the command `\refFormat`, which is executed just before the references are printed and which initially is simply `\relax`. For example, to have the reference list printed with a ragged right margin, with no vertical skip in between references, one could say

```
\def\refFormat{% format for the list of references
  \raggedright                      % ragged right aargin
  \def\refskip{\vskip0pt plus 2pt}}% no space, but some stretch

\ListReferences         % now do it
```

No heading is printed for the reference list — you have to do this yourself in whatever format is appropriate. Usually you will want to use `\nosechead`, but sometime you may want to use `\chapter` or `\section` from Section 7.3 instead.

6. Labels and Tags

It has already been noted that one of the most useful features of TEXsis is that it automatically numbers things like equations and references and lets you refer to these things by means of symbolic names or "labels." This section explains how to get the most use out of this labeling mechanism.

Here is a rough outline of how labeling or "tagging" works. TEXsis keeps track of numbers for equations, figures, tables, and other useful things in counters. Whenever you create a new equation, figure, table or reference the appropriate counter is incremented. At the same time, a new macro instruction is defined, with the label incorporated into the name, and the definition of this macro is simply the (new) value of the counter. Thus if you create a displayed equation and label it "Newton" then a macro called "`\Eq.Newton`" is created and it is automatically defined to be whatever the next number is for the equation.

[The actual mechanism is somewhat more subtle. Normally you can only use letters in macro names, but TEXsis uses `\csname ... \endcsname` to allow numbers and punctuation to be used in labels. Also, an extra `@` is added at the beginning and end of each macro name. If this were not done then, for example, using "`end`" as a label could redefine `\end`, and you would not be able to get out of TEX!]

Once a label macro is defined you can use it to refer to the thing it labels, and it will yield the proper number. In the case of the equation called "Newton," saying `\Eq{Newton}` is really saying `Eq.~(\Eq.Newton)`. (Again, since punctuation is not allowed in names, TEXsis actually uses `\csname ... \endcsname` in the definition.) References, equations, figures, tables, etc., all produce their numbers in this way.

Each of the TEXsis macros which uses labels attaches to the label a unique prefix identifying the kind of object labeled. These objects and the corresponding prefixes are shown in Table 6.1. Because of these prefixes the same label can be used for different kinds of objects. For example, the name "Witten" can be used as a label for an equation and as a reference, since their internal labels are `Eq.Witten` and `Ref.Witten`, respectively. The prefixes are automatically inserted by TEXsis macros like `\Eq` and `\cite`.

At the most primitive level you can always associate a label with some value by saying `\tag{`⟨label⟩`}{`⟨value⟩`}`. You can then reference the value, whatever it is, by it's name using the command `\use{`⟨label⟩`}`. In this case if there is to be a prefix in the label you must put it in explicitly. Thus to say just the number of the equation labeled "Newton" without any parentheses or anything you would say `\use{Eq.Newton}`. To get the number of the reference labeled "Knuth" you would say `\use{Ref.Knuth}`, which produces "1" because "Knuth" is the label of the first document we referenced in this manual and "Ref" is the prefix used for references.

One particularly useful way to use `\tag` and `\use` is to refer to some topic which is discussed on some other page in the document. On the page you want to reference you could define the tag "foo" to be the page number by saying `\tag{foo}{\folio}`. Then somewhere else in the document if you want to refer to the information on that page you could do so with `\use{foo}`, as in "This difficulty is discussed in more detail on page `\use{foo}`." Then even if the document is rearranged so that the page numbers change the page number you referred to will come out correct.

Table 6.1: Prefixes for labels and the corresponding counters for various objects.

Entity	Tag Prefix	Counter
Corollary	`Cor.`	`\corollarynum`
Definition	`Df.`	`\definitionnum`
Equation	`Eq.`	`\eqnum`
Figure	`Fg.`	`\fignum`
Lemma	`Lem.`	`\lemmanum`
Reference	`Ref.`	`\refnum`
Table	`Tb.`	`\tabnum`
Theorem	`Thm.`	`\theoremnum`

Knowing about `\use` and the prefixes in Table 6.1 also lets you change how the macros that refer to these objects work. For example, to refer to figure 9.1 in section 9.1 you would normally say `\Fig{Penguin}`. This produces "Fig. 9.1," but suppose you want it to say the full word "figure" instead of just "Fig." You can make this change throughout the document simply by changing the definition of `\Fig` to:

```
\def\Fig#1{figure~\use{Fg.#1}}
```

Then "`\Fig{Penguin}`" would produce "figure 9.1."

Whenever you create any of the objects listed in Table 6.1 a label is automatically created. However, chapters, sections, and subsections are not automatically labeled because their values are not ordinarily changed. If you want to remember the number of a chapter or section of your document to be able to refer to it somewhere else, put `\label{`⟨label⟩`}` in the text of the title of the chapter or section. `\label{`⟨label⟩`}` is the same as `\tag{`⟨label⟩`}{`⟨counter⟩`}`, where ⟨counter⟩ is the current section or chapter number, as appropriate. For example you might have:

```
\section{Supersymmetry and Supergravity \label{sect.SUSY}}
```

To refer to that section later on you would say `\use{sect.SUSY}`. Note that `\label` does not insert any prefix in front of the label, so you must include it explicitly if you want one. We suggest that you follow the example above and use "sect".

6.1. Forward References

If you label an equation or a figure or a section you can refer to the number of that object later in the text, but what if you want to refer to the thing before it has been tagged? Then the label won't exist yet. This is called making a forward reference. To allow for this TEXsis also writes the definition of every label to an auxiliary file, which has a name ending in `.aux`. The label will still be undefined the first time you make the forward reference, but the next time you run the document through TEXsis it will automatically read in the auxiliary file at the beginning of the job and define the label. Thus on the second pass the label will already be defined when it is called for. Having to make a second pass may sound like a lot of extra trouble, but when you are editing a document and putting the finishing touches on it you will find you end up making the extra pass without thinking about it. Just don't delete the `.aux` file between runs.

If you try to reference a label that does not exist you'll be told so at the end of the run (when you say `\bye`) and another pass will be suggested. Undefined references are also noted in the log file when they occur, and the label itself is inserted into the output text in boldface to make it easier for you to track down the problem. In `\draft` mode the mark "`UNDEF`" is also put in the right margin to help you find the problem.

7. Typesetting Complete Papers

The commands described so far are useful for creating the body of a simple physics paper, but they are not enough to typeset a complete paper. A physics paper generally has a title, one or more authors and their addresses or professional affiliations, and perhaps an abstract. Longer documents may be divided into sections, or even chapters, and they might even have a table of contents or an index. This section describes the commands for creating the basic structural elements of a complete paper: the title material and divisions into chapters, sections, and subsections. It also includes a discussion of more advanced topics, such as how to create a table of contents or an index, and how to control the running headlines at the top of the page. Later, Section 10 discusses more specialized document layouts for conference proceedings or for papers submitted to specific journals, while Section 11 describes the commands for producing letters, memos, referee reports, and similar documents.

It is often convenient to be able to typeset the same document in more than one format. Therefore, all complete papers in TEXsis use the same general structure, and one can select the particular document format to be used by changing only one or two commands at the begining of the manuscript file. The commands to typeset the title page and break the paper up into sections then behave slightly differently according to the particular format selected. For example, if you give the `\paper` command then the title page material will appear at the top of the first page, and the text of the paper will also begin on that page. If you simply change `\paper` to `\preprint` the title page material will appear on a page by itself, with a banner (defined by `\banner`) across the top of the page, and the text will begin on the next page. Using any of the other document formats described in Section 10 will cause the title page material to be formatted slightly differently.

At this point an example may be more instructive than more discussion. Here is how a famous paper[10] would have begun had it been typeset with TEXsis:

```
\texsis
\preprint
\pubdate{July 8, 1957}
\pubcode{hep-cats/yymmnnn}
\titlepage
\title
Theory of Superconductivity
\endtitle
\author
J.~Bardeen
Department of Physics
University of Illinois
Urbana, Illinois 61801
\endauthor
\and
\author
L.~N.~Cooper
Department of Physics and Astronomy
The Ohio State University
Columbus, OH 43210
\endauthor
```

```
\and
\author
J.~R.~Schrieffer
Department of Theoretical Physics
University of Birmingham
Birmingham, England B15 2TT
\endauthor
\abstract
A theory of superconductivity is presented, based on
the fact that...

\endabstract
\toappear{Physical Review}
\endtitlepage

\section{Introduction}
The main facts which a theory of superconductivity
must explain are...
```

A more complete example paper is given in the file `Example.tex`, which is included in the standard distribution of TEXsis. In the rest of this section each of the elements in these examples and several others related to the overall structure of a complete paper will be explained.

7.1. Document Types

A complete TEXsis paper should start with the `\texsis` macro described in Section 2. (While this is not required if you also use one of the macros listed below, it does serve to indicate unambiguously that the document is to be typeset with TEXsis.) The `\texsis` macro can be followed by a command to specify whether the document should be typeset as a simple paper or conference proceeding, a preprint, a book, etc. This macro will provide sensible defaults for the layout, although additional commands can be given to override them. The following basic document types are included in TEXsis:

`\paper` Produces a paper with the title and abstract at the top of the first page, such as for submission to a conference proceeding. The text begins on the first page, below the title and abstract.

`\preprint` Produces a paper in a preprint format, with the title material by itself on the first page. A `\banner` is put acros the top of the title page, and the text begins on the page following.

`\Manuscript` Similar to `\preprint`, but with true double spacing for manuscripts submitted for publication in journals such as the *Physical Review*, which require this.

`\book` A book or other long document with a separate title page, running headlines, a table of contents, etc. Page numbers are printed in opposite corners on left and right handed pages.

After the command defining the type of document you can add optional commands to modify the default layout. For example, you might want to use

```
\texsis
\paper
\hsize=⟨dimen⟩
\vsize=⟨dimen⟩
\singlespaced
```

to produce a single-spaced conference proceeding of a specified size. Macros to create more specialized documents in some of the common formats used by physics journals and conferences are described in Section 10. Since TEXsis obviously cannot include all the possible formats, that section also contains some hints about how to modify a format to fit your needs.

7.2. The Title Page

A complete paper almost always starts with a title. Most probably it will also include the names of the authors, and often it will include an abstract. Sometimes the title, author's name(s) and abstract appear on a separate page, while sometimes they simply appear at the top of the first page of the paper. In any case we will refer to all this material together as the "title page" material.

TEXsis has a standardized method for entering the title page material, whether it is to appear on a page by itself or not. You begin the title page material with `\titlepage` and end it with `\endtitlepage` (see the example at the beginning of this section). Between these two control sequences you can use any of the commands listed below to enter the various parts of the title page material:

`\title` Begins the title of the paper. All lines up to a following `\endtitle` are centered and printed in `\Tbf` type. Line endings in the input file *are* respected, just as with `\obeylines`.

`\author` Begins an entry for an author's name and address, which should appear on the lines that follow. Line endings in the input file *are* respected. The first line to follow is the author's name, which is printed centered in `\bf` type. All subsequent lines are centered in normal type. End with `\endauthor`.

`\and` Prints the word "and," centered and with appropriate vertical separation. It can be used either between sets of `\author` ... `\endauthor` or between multiple addresses for the same author.

`\authors` An alternative to `\author` for papers with many authors. Here line endings are not respected; instead, the list of authors is divided into lines as evenly as possible, and each line is centered and printed in Roman type. End with `\endauthors`.

`\institution{`⟨symbol⟩`}{`⟨address⟩`}`
Prints the symbol and the institution address centered on a line. A `\medskip` is placed before the first address to separate the names and addresses. `\institution` should be placed after the author list but before `\endauthors` to work correctly.

`\abstract` Begins the abstract. The word "ABSTRACT" is centered above the abstract. The left and right margins are brought in relative to the text of the paper. End with `\endabstract`.

`\pacs{`⟨codes⟩`}`
Prints the PACS (Physics and Astronomy Classification Scheme) codes on the title page.

`\submitted{`⟨Journal⟩`}`
Prints the phrase "Submitted to *Journal*" centered on the page, with the name of the journal printed in `\sl` type.

`\toappear{`⟨Journal⟩`}`
Similar to `\submitted`, this prints the phrase "To appear in *Journal*" on the title page, with the name of the journal printed in `\sl` type.

`\disclaimer{`⟨contract⟩`}`
Prints a standard DOE disclaimer for contract number ⟨contract⟩, printed in 10 pt. type, as a footnote.

The advantage to entering the title page material this way is that if you change the overall document format, either from `\paper` to `\preprint` or to one of the more specialized formats described in Section 10, then the title page material will automatically be changed to match the document format selected.

The `\banner` macro is called in the `\preprint` document format to put the date, a preprint code number, and the name of your institution or organization at the top of the title page. The name of the institution can be changed by saying

`\def\ORGANIZATION{`⟨name⟩`}`

and this change can be made permanent by putting it into the file `TXSsite.tex` when TEXsis is being installed. The default at this installation is

VASSAR COLLEGE

The preprint date and document number can be specified by putting

```
\pubdate{⟨date⟩}
\pubcode{⟨code-number⟩}
```

before the `\titlepage`. If these are omitted, the date defaults to the current month and year and the document number defaults either to something set as the default when TEXsis was installed or to the current TEXsis version number. The default at this installation is "VC-PHY ". If you really want to customize the look of your paper you can create your own `\banner` macro.

If you are submitting your paper to one of the preprint mail servers such as `hep-lat`, `hep-ph`, or `hep-th` you can have the server automatically insert the preprint number it assigns your document when it is submitted. To do this put

```
\pubcode{hep-lat/yymmnnn}
```

near the top of the document (change "hep-lat" as appropriate to the server you are using). When you submit a paper to a server with the "put" command it will translate any occurrence in the input file of the sequence "`hep-lat/yymmnnn`" to "`hep-lat/`" followed by the 7 digit paper number. You must use the exact string of characters "`yymmnnn`" for this to work. When someone else gets your paper from the server and prints a copy it will have the assigned document code in the banner at the top of the title page.

The standard `\footnote` macro cannot be used to make footnotes to titles and authors's names, because these special fields use internal vertical mode, where footnotes are not allowed. You should instead use the `\vfootnote` macro of Plain TEX for such purposes. In this case you must put the reference mark in the title or author list yourself (for example, with `${}^*$`), and then say `\vfootnote*{The footnote}` somewhere else on the page, outside of the internal vertical mode material (the author or title blocks).

"Footnotes" identifying institutions in a long list of authors are often centered on lines immediately below the author list. `\authors` and `\institution` can be used for this. For example, the author list of one well known paper[11] could be typed as follows:

```
\authors
E.D.~Bloom,$^a$ M.~Breidenback,$^b$ D.H.~Coward,$^a$
H.~DeStaebler,$^a$ J.~Drees,$^a$ J.I.~Friedman,$^b$
G.C.~Hartmann,$^{b,c}$ H.W.~Kendall,$^b$ G.~Miller,$^a$
L.W.~Mo,$^a$ R.E.~Taylor$^a$
\institution{a}{Stanford Linear Accelerator Center,
   Stanford, CA 94305}
\institution{b}{Department of Physics and Laboratory for Nuclear
Science, Massachusetts Institute of Technology,
   Cambridge MA 02139}
\institution{c}{Xerox Corporation, Rochester, NY}
\endauthors
```

This produces

E.D. Bloom,[a] M. Breidenback,[b] D.H. Coward,[a] H. DeStaebler,[a] J. Drees,[a]
J.I. Friedman,[b] G.C. Hartmann,[b,c] H.W. Kendall,[b] G. Miller,[a] L.W. Mo,[a] R.E. Taylor[a]

[a] Stanford Linear Accelerator Center, Stanford, CA 94305

[b] Department of Physics and Laboratory for Nuclear Science,
Massachusetts Institute of Technology, Cambridge MA 02139

[c] Xerox Corporation, Rochester, NY

Note that `\institution` must come *before* the `\endauthors`.

7.3. Chapters and Sections

The following commands can be used to create and automatically number the divisions of a document, including chapters, sections, and appendices:

`\chapter{`⟨title⟩`}`
: Begins a chapter with ⟨title⟩ printed in `\Tbf` type on a new page.

`\section{`⟨title⟩`}`
: Begins a section with the ⟨title⟩ printed in `\tbf` type. This does *not* begin on a new page (although you certainly can start a section on a new page by saying `\vfill\supereject` first.)

`\subsection{`⟨title⟩`}`
: Begins a subsection with the ⟨title⟩ printed in `\bf` type.

`\subsubsection{`⟨title⟩`}`
: Begins a sub-subsection entitled ⟨title⟩. This is a lower level than even a subsection and should only be used when truly appropriate.

`\Appendix{`⟨label⟩`}{`⟨title⟩`}`
: Begins a chapter-like appendix labeled by the letter ⟨label⟩ (generally A, B, C, ... or I, II, III, ...). The ⟨title⟩ is printed in `\Tbf` type at the top of a new page.

`\appendix{`⟨label⟩`}{`⟨title⟩`}`
: Begins a section-like appendix labeled by the letter ⟨label⟩ (generally A, B, C, ... or I, II, III, ...). The ⟨title⟩ is printed in `\tbf` type.

`\nosechead{`⟨title⟩`}`
: The ⟨title⟩ is typeset like a subsection headline, but without a subsection number. This is suitable for acknowledgments, lists of references and such.

The title appears at the top of the section or subsection, left justified in the appropriate typestyle for the document type and with an appropriate amount of vertical spacing below. These and other details of the layout vary with the document format. In general `\chapter` is appropriate mainly for books or theses; the manual you are reading now uses only `\section` and `\subsection` divisions.

By default chapter, section, and subsection numbers are printed with the titles, and the chapter and section numbers are also attached to equation, figure and table numbers in the form "*cc.ss.nn*", where *cc* is the chapter number, *ss* the section number, and *nn* the equation number. The inclusion of the chapter and section numbers can be turned on or off with the commands

`\showchaptIDtrue`	Display chapter numbers.
`\showchaptIDfalse`	Do not display chapter numbers.
`\showsectIDtrue`	Display section numbers.
`\showsectIDfalse`	Do not display section numbers.

`\Appendix` and `\appendix` give chapter-like and section-like appendices, respectively. The chapter or section number is replaced by a letter, which is given as the first argument, and the title, which is the second argument, is printed in the same format as a chapter or section title. For example,

```
\appendix{A}{Review of Functional Integration}
```

The letter labeling the appendix is attached to the equation numbers if `\showchaptIDtrue` or `\showsectIDtrue` is used. You can omit the letter by using an empty first argument, as in

```
\appendix{}{Leading logarithm approximation}
```

in which case the result is the same as setting `\showchaptIDfalse` or `\showsectIDfalse`.

A skip called `\sectionskip` is used before the title of any section-like division (i.e., a `\section` or an `\appendix`); a skip called `\subsectionskip` is used for any lower-level divison, that is, a `\subsection`, `\subsubsection`, or a `\nosechead`. Of course these are all removed by TEX if the section starts on a new page. Furthermore, any of these divisions will start on a new page if less than a minimum length given by `\sectionminspace` remains on the current page. The defaults for these parameters are

```
\sectionskip=2cm plus 8pt minus 8pt     % skip before \section
\subsectionskip=1cm plus 4pt minus 4pt  % skip before \subsection
\sectionminspace=0.25\vsize             % new page if less left
```

but they can be changed to any legal TEX skip or dimension either throughout the document or in a local group. For example, you could use

```
\begingroup
\subsectionskip=\sectionskip
\fourteenpoint
\nosechead{Acknowledgments}
\endgroup
```

to make a `\nosechead` which looks like a `\section`.

Ordinarily the order of appearance of the sections of a document are not changed, so there is no automatic mechanism for labeling the sections. Any of the divisions described above can be labeled by inserting `\label{`⟨label⟩`}` into the title text. The ⟨label⟩ should have the form ⟨type.name⟩, where ⟨type⟩ is `chap`, `sect`, etc., to avoid possible conflicts with other kinds of labels. The corresponding numbers can then be referred to with `\use{`⟨type.name⟩`}`, as described in Section 6.

If the title of a chapter or section is too long, a break in the line can be inserted with the `\n` command. (This will also break the line in the title when it appears in the table of contents.) If running headlines are being used (see Section 7.5) then only the first part of the title, up to the `\n`, will be used for the running headline. You can always set the headline by hand with `\setHeadline`, as is described in Section 7.5.

You can customize the behavior of the section-making macros by adding your own definitions for the macros

`\everychapter`, `\everysection`, `\everysubsection`, `\everysubsubsection`,

which are called immediately before making the corresponding division of the document, or for

`\afterchapter`, `\aftersection`, `\aftersubsection`, `\aftersubsubsection`,

which are called immediately afterwards. By default all of these macros do nothing. As an example, if you wanted every section to begin on a new page, and you wanted numbered footnotes created by `\NFootnote` to start over counting from one, you could define:

```
\def\everysection{\vfill\supereject \footnum=0}
```

Another way you can control the results produced by `\chapter` and `\section` is to change the way the chapter or section number is printed, by changing the definitions of the macros `\ChapterStyle` or `\SectionStyle` or `\SubsectionStyle`. These macros take one argument, the chapter or section number, and by default they simply return that argument. By substituting your own definition you can do almost anything you like to the section number. For example, it is often desired that the section number be printed as a roman numeral rather than an arabic numeral. The following definition will convert the section number to an upper-case roman numeral:

```
\def\SectionStyle#1{\uppercase\expandafter{\romannumeral #1}}
```

Even more customized behaviour of the section-making macros is possible. For example, `\Tbf` and `\tbf` can also be defined as macros which take a single argument, which is the title of the section or chapter. To see how this works it may be best to look at the definitions of `\chapter` and `\section` and the like in the source file `TXSsects.tex`. For very specialized applications you could even copy the definitions in this file to the TXSmods.tex file and then alter them however you like.

7.4. Table of Contents

The section-making macros just described automatically make an entry in a table of contents for the document being prepared. The title of the chapter or section and the page number are stored in a table of contents file (which ends with the extension ".toc") until needed, then the table of contents is created simply by saying `\Contents`. If your document is divided into chapters, like a book, the table of contents should appear on a page with a Roman numeral page number (which is obtained by setting `\pageno` to a negative value). For example:

```
\book                       % document format for a book
\input Chapt1               % input chapters...
...
\vfill\supereject           % eject last page, including figures
\pageno=-3                  % print on pages iii, iv, ...
\nosechead{Contents}        % label TOC without section #
\Contents                   % generate table of contents
```

The table of contents is usually printed at the *end* of the document so that all the page numbers are correct. You then simply move these pages to the front of the document by hand. Alternately, you can say `\Contents` at the begining of your document to read in the table of contents file produced by a previous run of TEXsis, although then you must process the document twice to be sure that the page numbers are correct.

It is important to note that nothing is written to the table of contents file unless `\Contentstrue` is set. The default for most documents is `\Contentsfalse`, since one usually does not want a table of contents for a short paper or a memo or a letter. The exceptions are `\book` and `\thesis`, which automatically set `\Contentstrue`. In any case, if you want a table of contents you can explicitly state `\Contentstrue`.

While an entry in the table of contents is created by `\chapter` or `\section`, no entry is made for `\nosechead`. In this case, or in any case, you can insert an entry into the table of contents "by hand" by saying

`\addTOC{`⟨level⟩`}{`⟨title⟩`}{`⟨page⟩`}`

where ⟨level⟩ is the level of division for the entry being made, with 0 for chapters, 1 for sections and 2 for subsections. This controls how much vertical space is put around the table of contents entry and how much the entry is indented in the list. The ⟨title⟩ is the text of the title as it should appear in the table of contents. Text which is too long for one line will be split over several lines, but you can force a line break where you want one with `\n`. The ⟨page⟩ is the page number for the entry you are making. Generally you'll use `\folio` here to just use the current page number, whatever that may be. Thus to make a section level entry in the table of contents having the title "Acknowledgments" you would simply say

```
\addTOC{1}{Acknowledgments}{\folio}
```

Entries made in the table of contents by `\chapter`, `\section`, etc. will optionally begin with the chapter or section number, depending upon whether `\showchaptIDtrue` and `\showsectIDtrue` are selected. The decision whether or not to show the chapter or section number is made when the table of contents is printed by `\Contents`, not when the `\chapter` or `\section` is begun. Thus it is possible to have the section numbers showing at the beginning of the section, but to then have no section numbers in the table of contents by saying `\showsectIDfalse` immediately before saying `\Contents`.

It is possible to write commands or other text to the table of contents file while it is being created, using the command `\TOCwrite{`⟨stuff⟩`}`. For example, the heading above the appendicies in the table of contents of this Manual was created by saying

```
\TOCwrite{\bigskip\noexpand\nosechead{Appendices:}}
```

before the appendices were created. The `\noexpand` prevents the `\nosechead` from being-expanded until later, after the table of contents file is read back in.

What if you want the table of contents to appear in the table of contents itself!? Naturally it should appear first (even though it may be printed at the end of the document). You can use the tagging mechanism described in Section 6 to make a "forward reference" to the page number. At the beginning of the document you can create the table of contents entry by saying

```
\addTOC{0}{Table of Contents}{\noexpand\use{Pg.tabcontents}}
```

The `\noexpand` prevents the `\use` from being expanded until later, after the table of contents file is read back in. Later, when you actually print the table of contents you would first tag the page number, then invoke `\Contents`, like so:

```
\nosechead{Table of Contents}
\tag{Pg.tabcontents}{\folio}
\Contents
```

7.5. Running Headlines and Footlines

In Plain TEX whenever a page is output an optional headline and footline are added at the top and bottom of the page. In TEXsis the footline is normally blank, while the headline contains the page number in the upper right corner. The `\book` and `\thesis` formats also display a running headline, centered in 10 pt. italic type, containing the chapter or section title. Running headlines and page numbers are not printed on any page containing a `\title` or a new `\chapter`. For an example of running headlines look at the top of any page in this Manual.

`\nopagenumbers` turns off the running headlines and footlines, just as it does in Plain TEX. In TEXsis saying `\pagenumbers` restores the standard headlines and footlines. `\bookpagenumbers` puts odd page numbers in the upper right corner and even page numbers in the upper left corner, the normal convention for books. This is the default for `\book`.

`\RunningHeadstrue`, the default for `\book` (and `\thesis`), displays the running headline text `\HeadText`, as well as the page number. `\RunningHeadsfalse` turns off the running headlines but leaves the page numbers. `\HeadText` is defined by `\chapter` to be the chapter title in 10 pt. italic type; if no chapters are used, then it is defined by `\section` to be the section title. You can change the running headline yourself with the command `\setHeadline{`⟨text⟩`}`. For example, the headline at the top of this page could have been created by saying

```
\setHeadline{Typesetting Complete Papers }
```

The headline text appears in typestyle `\HeadFont`, which you can change to suit your tastes. The default is

```
\def\HeadFont{\tenit}
```

7.6. How to Make an Index

The traditional way to make an index for a book was to write key words for the index on 3×5 cards, and to then write on these card the numbers of the pages on which the key words occured. The cards were then sorted by hand and their contents were typed up in order as the index. This was a tedious and time consuming task. Fortunately this whole process has been modernized so that an index can be made almost automatically. Still, it takes a certain amount of care to create a useful, well constructed index. An important part of designing a good index is deciding which key words will be useful in the index and which key words will not. A computer cannot make these decisions for you.

Making an index with TEX still involves collecting the index entries, sorting them (and removing duplicate references), and then producing the final index. The first and last steps are accomplished with a set of Plain TEX macros known as `index.tex`, while the sorting and collating of the index entries is done by a separate computer program called `makeindex`. This is how we made the index for this Manual.

To create an entry in the index you can say `\index{`⟨key⟩`}`, where ⟨key⟩ is the key word to be entered into the index, or more simply you can say `\idx{`⟨word⟩`}`, which both includes the word in the text and makes an index entry for it; saying "`\idx{frisbee}`" is the same as saying "`frisbee\index{frisbee}`". Either of these indexing commands writes the index entry to a file with a name which ends with the extension "`.idx`". Once your document has been processed by TEX this file will contain all of the "raw" index entries.

The entries in the raw index file are not sorted, and there may be many duplications. The `makeindex` program will sort the index entries and reformat them in a form appropriate for printing the index. The output from the `makeindex` program is a file with the extension "`.ind`". So if your manuscript file is called `myfile.tex` and you have run it through TEXsis you should find a file called `myfile.idx` containing the raw index entries. You would run the `makeindex` program like so,

```
makeindex myfile.idx
```

to produce the processed index file `myfile.ind`. (You can leave off the `.idx` extension for the raw file and `makeindex` will assume it.)

To print the index you can either include it in another pass of the full document, by saying

```
\input \jobname.ind
```

in the manuscript file, or you can print it separately by adding "`\input index`" to the begining of the `jobname.ind` file and running it through TEX or TEXsis.

If you set `\markindextrue` then indexed words selected with `\idx` will appear in the printed output inside a ruled box (which does not interfere with or change the word or line spacing), like so: frisbee. These "proof marks" are useful for draft copies when you are putting together the index, but obviously you do not want them to appear in the final copy, so the default is `\markindexfalse`.

The `index.tex` macros are distributed with TEXsis, but `makeindex` is not. If you do not have the `makeindex` program you can get the C source code for it from the TEXsis anonymous ftp site, `lifshitz.ph.utexas.edu`. It is a well-known program and is also available from many other sites on the net. Note that the `index.tex` macros work with Plain TEX as well as TEXsis, and they work with `makeindex` in it's default mode, even though that default mode is set up for LATEX code.

For more information about how to make full use of `makeindex` see the file `index.tex`, the Unix man page for makeindex(1), and the original paper[12] by Chen and Harrison describing how they designed the `makeindex` program.

8. Environments

TEX ordinarily provides a layout appropriate for straight text and equations. The TEXsis macros described below create environments which bypass the normal layout to format text in special ways, to create different types of lists, to display theorems and proofs, to display computer code, and even to print TEX examples. With the exception of some of the specialized TEX example macros, the syntax for all of these is generally of the form "`\thing`..." to begin the environment and "`\endthing`" to end it.

8.1. Flush or Centered Text

There are three environments in TEXsis which center lines, set them flush left, or set them flush right, respectively:

`\center`	Centers all lines which follow, up to `\endcenter`.
`\flushleft`	The lines which follow, up to `\endflushleft`, are made to line up flush to the left margin.
`\flushright`	The lines which follow, up to `\endflushright`, are made to line up flush to the right margin.

In all of these environments line endings in the input file *are* obeyed, as with `\obeylines`. For example,

```
\center
A Centered Line
and Another
\endcenter
```

produces

A Centered Line
and Another

There is also a `\raggedcenter` environment which does not respect line breaks but instead produces centered lines each of which is made as long as possible:

`\raggedcenter`	Center lines without respecting line endings. Terminate with `\endraggedcenter`.

For example

```
\raggedcenter
In the ragged centering environment each line
is made as long as possible, and
the successive lines are centered.
\endraggedcenter
```

produces

> In the ragged centering environment each line is made
> as long as possible, and the successive lines are centered.

Note that `\raggedcenter` does not hyphenate words.

8.2. List Environments

When you want to make a list of things there are several ways to do it. The items in the list can be marked by a simple marker (like a bullet: •), by a number or numbers and letters (as in an outline), or by some sort of descriptive text in the margin. In Plain TEX you would make all of these kinds of lists using the `\item` macro. The "list environments" described below are generalizations of `\item` for making these various kinds of lists:

`\itemize`	List of paragraphs marked by a bullet (•) or other symbol.
`\enumerate`	List of paragraphs labeled by numbers or by combinations of numbers and letters, as for an outline.
`\description`	List of items and paragraphs describing them.

In each list environment `\itm` is used to introduce each individual item. The environments are terminated with `\enditemize`, `\endenumerate` and `\enddescription`, respectively. The three different kinds of environments can be nested to produce a list with several levels. The indentation of and spacing above and below the list environments can be easily controlled. This is described in Section 8.2.4.

8.2.1. Itemized lists

`\itemize` produces paragraphs marked by bullets (•) or some other appropriate symbol; this is usually used for a short list with only a single level of nesting where the order either does not matter or is already contained in the items of the list. For example, `\itemize` was used to produce the list of sections of this manual which begins on page 1. The first few items in this list were produced by saying:

```
\itemize
\itm Sect.~\use{sect.start}:
Getting Started.  How to run and initialize \TeXsis,
and the "master file" concept.
\itm Sect.~\use{sect.fonts}:
Selecting sizes and fonts of type, choosing single,
double, or triple spacing, and automatically handling
quotation marks.
\itm Sect.~\use{sect.eqns}:
Automatic numbering of equations, automatic sizing of
parentheses and brackets, and special symbols for
```

```
physics equations....
```

A paragraph begun by `\itm` is not indented beyond the new left margin, but paragraphs that follow *are* indented unless `\noindent` is specified (or if they also begin with `\itm`).

You can change the symbol used to mark each item in the list to be anything you want by creating your own definition of `\itemmark`. For example, if you want the symbol to be a `\square` you would define `\itemmark` to be

```
\def\itemmark{$\square$}
```

before invoking `\itemize`, or at least before the first `\itm`. An example is the following important checklist:[13]

- □ Event timer started (display: SINED)
- □ Antenna Aligned (display: SEELED)
- □ Oscillation overthruster armed (display: DELIVERED)

This was produced by saying:

```
\itemize
\def\itemmark{$\square$}
\itm    Event timer started (display: SINED)
\itm    Antenna Aligned (display: SEELED)
\itm    Oscillation overthruster armed (display: DELIVERED)
\enditemize
```

`\itemize` starts a new group, which is ended by the `\enditemize`, so in this example `\itemmark` reverts back to it's previous definition when the list is finished. You can get a "check-mark" symbol instead of a square by replacing `\square` with `\surd`, or you can use any other symbol that suits you. Note though that in this example the `\square` is enclosed in math quotes — you will have to remember this for any symbol which is defined in math mode.

8.2.2. Enumerated lists

`\enumerate` produces numbered paragraphs, either with a single level or nested several levels deep, like so:

```
\enumerate
\itm Paragraph 1.
\enumerate
\itm sub-paragraph 1a.
\enumerate
\itm sub-sub-paragraph 1a(i).
\itm sub-sub-paragraph 1a(ii).
\endenumerate
\itm sub-paragraph 1b.
\itm and so on...
```

```
\endenumerate
\itm Paragraph 2.
\itm and so on...
\endenumerate
```

This produces the enumerated list:

1. Paragraph 1.
    1.1. sub-paragraph 1a.
        1.1.1. sub-sub-paragraph 1a(i).
        1.1.2. sub-sub-paragraph 1a(ii).
    1.2. sub-paragraph 1b.
    1.3. and so on...
2. Paragraph 2.
3. and so on...

The default form for the paragraph numbering is called `\enumPoints`, which produces numbers of the form $ii.jj.kk$. You can also have the list printed in a standard outline format with a mixture of Roman numerals, letters, and Arabic numerals by saying `\enumOutline`. Doing so to the example above produces:

I. Paragraph 1.
    A. sub-paragraph 1a.
        i. sub-sub-paragraph 1a(i).
        ii. sub-sub-paragraph 1a(ii).
    B. sub-paragraph 1b.
    C. and so on...
II. Paragraph 2.
III. and so on...

Unless enclosed within a group, the `\enumPoints` or `\enumOutline` style selected remains in effect until it is changed. The punctuation following the item number is `\enummark`, which by default is a period (.) but which can be redefined by the user [for example, you can change it to `\rparen` to obtain numbering of the form "2)"]. `\enummark` can also be used to change the spacing between the numbers and the text that follows.

A third enumeration style, called `\enumNumOutline`, is also available. Top level items are labeled by arabic numbers, the level below by lower case roman letters, and the level below that by lower case roman numerals. With `\enumNumOutline` the example above becomes:

1. Paragraph 1.
    a. sub-paragraph 1a.
        i. sub-sub-paragraph 1a(i).
        ii. sub-sub-paragraph 1a(ii).
    b. sub-paragraph 1b.
    c. and so on...
2. Paragraph 2.
3. and so on...

If you prefer a different style of enumeration you can create your own `\enumWhatever` to use in place of `\enumPoint` or `\enumNumOutline`. To do so, copy the definition of `\enumPoints` or one of its brothers, along with the comments that go with them, from the file `TXSenvmt.tex` into a file called `TXSmods.tex` (see Appendix A) and then make the appropriate changes as described in those comments.

8.2.3. Lists Labeled by Text

`\description` is used to create a list of words or other items labeled by some text. The syntax is

```
\description{⟨sample text⟩}
\itm{⟨label 1⟩} Paragraph 1...
\itm{⟨label 2⟩} Paragraph 2...
\itm{...}
\enddescription
```

The argument ⟨sample text⟩ is not printed, but its length is used to determine the indentation for the labels on the items, so it should be the longest item, followed by some extra space. For example, the descriptions of the flush and centered environments at the beginning of this section of the manual were typeset with

```
\description{~flushright~\qquad}
\itm{\CS{center}}
Center each line.
\itm{\CS{flushleft}}
Set each line flush left.
\itm{\CS{flushright}}
Set each line flush right.
\enddescription
```

since `\flushright` is the longest of the three items. (`\CS` is a special macro defined for this manual for printing control sequence names.) Each item in the list is separated by a `\parskip`, just as in itemized or enumerated lists.

If the label text for an item comes out wider than the space allowed, the text will usually be broken into several lines so that it will fit in the indented margin (if this is possible!). Saying `\singlelinetrue` will instead put the label text on a line by itself, above the item. `\singlelinefalse` is the default, but this manual has been printed with `\singlelinetrue`.

8.2.4. Spacing of List Environments

Each of the list environments described above is surrounded by the following skips:

```
\EnvTopskip    = \medskipamount     % space above the list
\EnvBottomskip = \medskipamount     % space below the list
\EnvLeftskip   = 2\parindent        % left indent for the list
```

```
\EnvRightskip  = \parindent          % right margin moves in too
```

You can change these as you please to any legal TEX dimension. The space between items which are begun by `\itm` is taken from the `\parskip`. Since the list environment begins a group you can change the `\parskip` inside this group to control the spacing between `\itm`'s. For example, if you wanted no extra space between the items in the list you could say set the `\parskip` to zero, as in,

```
\itemize
\parskip=0pt    \def\itemmark{$\triangleright$}
\itm    Event timer started (display: SINED)
\itm    Antenna Aligned (display: SEELED)
\itm    Oscillation overthruster armed (display: DELIVERED)
\enditemize
```

which would produce:

- ▷ Event timer started (display: SINED)
- ▷ Antenna Aligned (display: SEELED)
- ▷ Oscillation overthruster armed (display: DELIVERED)

8.3. Simple Examples and Quoted Text

The `\example` environment can be used to display an example of something in a way that sets it off from the main text. The indentation of an example is the same as in the list environments (see Section 8.2.4). The text in an example will be single spaced, even if the rest of the document is double spaced. End this environment with `\endexample`. An example of the use of `\example` is

```
\example
Physics is not a finished logical system.  Rather, at any
moment it spans a great confusion of ideas, some that
survive like folk epics from the heroic periods of the past,
and other that arise like utopian novels from our dim
premonitions of a future grand synthesis.
\endexample
```

which produces[14]

> Physics is not a finished logical system. Rather, at any moment it spans a great confusion of ideas, some that survive like folk epics from the heroic periods of the past, and other that arise like utopian novels from our dim premonitions of a future grand synthesis.

Another example of `\example` is given in the next subsectione as an example of the `\theorem` environment.

Since one of the uses of `\example` is to display block quotes, as we've just done above, a synonym for `\example` is `\blockquote`, which should be ended with `\endblockquote`.

8.4. Theorems and Proofs

Theorems, lemmas, and corollaries are typeset in a quasi-SIAM format, with the number in boldface and the text in slanted type. They are automatically numbered, including the chapter and section numbers if `\showchaptIDtrue` and `\showsectIDtrue` are being used. The syntax for a theorem is

```
\theorem{⟨label⟩}
⟨text⟩
\endtheorem
\proof
⟨text⟩
\endproof
```

and similarly for `\lemma`, `\corollary` and `\definition`. Definitions are printed in Roman type, not slanted. The `\endproof` macro inserts a black box to mark the end of the proof. Subsequent references to theorems, lemmas, corollaries and definitions can be made in the text automatically using `\Theorem{⟨label⟩}`, `\Lemma{⟨label⟩}`, `\Corollary{⟨label⟩}`, and `\Definition{⟨label⟩}`, which give the appropriate name and number. An example of a Theorem is:

```
\example
\theorem{Cauchy}
If $f(z)$ is a function of $z$ which is analytic on and inside of
a contour $C$, then
$$
\int_C f(z)\,dz = 0
$$
\endtheorem
\proof
Left as an exercise to the reader.
\endproof
\endexample
```

which produces

Theorem 8.1. *If $f(z)$ is a function of z which is analytic on and inside of a contour C, then*

$$\int_C f\,(z)\;dz = 0$$

Proof. Left as an exercise to the reader. ∎

Note the use of `\example` to indent the theorem and proof and print them single spaced. Had the `\example` been omitted the theorem and proof would have had the same margins and spacing as the rest of the text. This theorem would be referred to in the text with `\Theorem{Cauchy}`, which gives "Theorem 8.1." This should be sufficient in most cases, but if necessary one can refer to theorem or lemma numbers with the primitive `\use{⟨type⟩.⟨label⟩}`, where ⟨type⟩ is either `Thm`, `Lem`, respectively. See Section 6 for more on the use of `\use`.

8.5. Listing Computer Code

The `\Listing` environment can be used to display computer code or fragments of computer code. The text is printed in `\tt` type with line endings respected and every space counted. You can still use TEX control sequences in this environment, but many of the special characters like `%`, `&`, and `#` are simply printable characters, since they are used in languages like C. (`$`, `_` and `^` also lose any special meaning in this environment.) Since the curly brackets used by TEX are also used in C (and other languages) they are also simply characters in the `\Listing` environment. To perform grouping you should use `\bgroup` and `\egroup`. The backslash is also used in C as an escape character, so there has to be some way to print it — you simply use `\\`; thus `\n` should be typed as `\\n`. You end the listing environment with `\endListing`. As an example here is a very famous little program,[15]

```
#include <stdio.h>
main()
{
   printf("hello, world\n");
}
```

which was created by simply typing,

```
\Listing
#include <stdio.h>
main()
{
   printf("hello, world\\n");
}
\endListing
```

The indentation and spacing of the of the `\Listing` environment are controlled by the same parameters that control the indentation of the list environments, as described in Sect. 8.2.4. In addition, at the begining of the `\Listing` environment the macro `\everyListing` is executed, if it exists. You can define `\everyListing` to change the typestyle, type size, or spacing in the `\Listing` environment.

An alternative way to list computer code is `\ListCodeFile{`⟨filename⟩`}`, which will list the contents of a source code file exactly as it appears in the file (with no need to make `\n` into `\\n` and such). The call to this macro should *not* be enclosed in a `\Listing` environment. If the example program above were contained in the source file `hello.c` it could be listed simply by typing

```
\ListCodeFile{hello.c}
```

With `\ListCodeFile` the right margin of the page is extended outward to make room for long lines. If you find that you have problems getting the listing to fit on the page you may want to make the type size smaller before you invoke `\ListCodeFile` (you can do this using `\everyListing`). Note also that tab characters in the code file may not be treated exactly correctly. In the present implementation of TEXsis they will simply be replaced with eight spaces. We hope to be able to improve how this macro deals with tabs in in

a future release of TEXsis, but it's difficult to do it exactly right in all cases, so for now it may be best to change tabs to spaces in the source file if proper tab stops are needed. Most good editors (like emacs) already do this for you, so this should not pose much of a problem.

8.6. TEX example macros

Typing TEX examples can't be done in either the `\example` environment nor the `\Listing` environment, because the backslash must be turned into a simple printing character. There are several ways TEXsis allows you to show examples of TEX code. First, to display TEX instructions separately, use

`\TeXexample` Prints like an `\example`, but verbatim in `\tt` type, with "`|`" made the escape character. Note that you must therefore say `|endTeXexample` to end the example, since "`\`" is no longer the escape character!

To display TEX instructions in running text use

`\TeXquoteon` Make "`|`" a "TEX quote" so that everything between `|`'s is printed verbatim in `\tt` type.

`\TeXquoteoff` Turns off "TEX quotes."

All of the displayed examples in this document have been typed using `\TeXexample`, and many of the inline examples have been typed by putting `\TeXquoteon` at the beginning of the document and then typing `|\whatever ... |`. For both `\TeXexample` and `\TeXquoteon` the only special character is the escape character "`|`". In particular one must type

```
\TeXexample
⟨TeX stuff⟩
|endTeXexample
```

with `|endTeXexample` instead of `\endTeXexample`. All other characters are set in `\tt` type without any interpretation. The input line breaks and spacing are respected, including any leading spaces on the lines. TEX commands can be used within a `\TeXexample` environment by typing `|macro` in place of `\macro`, `|bgroup` and `|egroup` in place of `{` and `}`, etc. In particular, a `|` can be obtained by typing `|char124`. If you want to display the results of a complicated macro inside the `\TeXexample` environment you should probably define a different macro to display these results and then invoke that macro inside the environment. For example, in this manual we have sometimes inserted "meta" constructions showing where to insert a particular piece of text by using a `\meta` macro which encloses it's argument in angle brackets (see the example immediately above). But in a `\TeXexample` you cannot say "`|meta{TeX stuff}`," because the "`{`" and "`}`" are no longer grouping characters. Instead you could define the temporary macro `\tmp` as

```
\def\tmp{\meta{TeX stuff}}
```

and then inside the `\TeXexample` environment simply type `|tmp` to insert the phrase "⟨TeX stuff⟩".

The indentation and spacing of the `\TeXexample` environment are controlled by the same parameters that control the indentation of the list environments, as described in Sect. 8.2.4. The general method used by all of these macros to turn off TEX's usual mechanism for interpreting special characters is discussed in *The TEXbook*, p. 421ff. The `\begintt` and `\beginlines` macros described there are included in TEXsis for completeness.

9. Figures and Tables

Technical papers often include tables and/or figures. Regardless of how the contents of the figures or tables are created, these have a lot in common in the way they are generally treated. TEXsis makes use of TEX's floating insertion capabilities (see Chapter 15 of *The TEXbook*) to insert figures and tables on the current page or on a following page. It is also possible to have figures or tables saved and printed at the end of the document. Captions are easily added either above or below the figure or table. Both kinds of objects are automatically assigned numbers, which are associated with internal labels so that they may be referred to in the text without having to know their order. It is possible to create a list of the figures and tables which appear in a document, much like creating a table of contents. Figures themselves are generally created outside of TEX, either by hand or by computer graphics programs, while tables can be created using TEX commands. The most complicated tables have vertical and horizontal lines (called "rules") to guide the reader's eye and to make the table more readable. TEXsis has some very easy to use commands for making ruled tables.

Before discussing TEXsis's figure and table making macros in detail we should mention that TEXsis also adds a new insertion class to those of Plain TEX. In Plain TEX saying `\topinsert`, `\midinsert` or `\pageinsert` begins a floating insertion (ended by `\endinsert`) which will appear somewhere on the current page or a following page. Material entered with a `\topinsert` appears at the top of the current page, if there is room for it, or at the top of the next page where there is room. A `\midinsert` attempts to insert the text into the current page right where it is invoked, but if the insertion won't fit it becomes a `\topinsert`. Saying `\pageinsert` creates an insertion which takes up a whole page by itself. In TEXsis you can also say `\bottominsert` to make a floating insertion that will appear at the bottom of the current page, if it will fit, or at the bottom of the next page where it will fit. There is also `\heavyinsert`, which like `\midinsert` trys first to insert the material into the current page right where it is called, but in this case becomes a `\bottominsert` if the material wont' fit.

9.1. Figures

TEX is designed for technical typesetting, but does not include a graphics standard, so to include a figure or drawing in a document it is necessary to leave blank space and to either glue the drawing in later or to arrange for the figure material to be inserted by some mechanism outside of TEX when the page is actually printed (e.g., using the `\special` command or the `\epsfbox` command described later).

Imagine that you wish to insert into the text a drawing of a "penguin diagram," which is about 6 cm tall. You could create the space for this figure at the top of a page, along with a caption, with the following commands:

```
\figure{Penguin}
\hrule
\vskip 3cm
\line{\hss (glue your penguin here) \hfil}
\vskip 3cm
```

(glue your penguin here)

Figure 9.1: Sample figure showing how easy it is to insert numbered figures with captions in TEXsis.

```
\hrule
\Caption
Sample figure showing how easy it is to insert
numbered figures with captions in \TeXsis.
\endCaption
\endfigure
```

This produces Fig. 9.1 at the top of the next page (we put the `\hrule`'s in just to show off the empty space). "Penguin" is the label for the figure, so to refer to the figure in the text we typed `\Fig{Penguin}`, which produced "Fig. 9.1." The caption can be placed either before or after the figure: if it is to be at the top then `\Caption` must be the first token after `\figure`. As shown, the caption text must be terminated with `\endCaption`. When the caption is printed it will automatically start with the figure number in bold type, followed by the caption text. The caption is indented on both sides if it is more than one line long and centered otherwise.

`\caption{`⟨text⟩`}` is equivalent to `\Caption`⟨text⟩ `\endCaption` and may be more convenient for short captions. It should be avoided for very long captions to keep from overflowing TEX's internal memory.

There are commands to create figures for each of the insertion classes of Plain TEX(that is, `\midinsert`, `\topinsert` and `\pageinsert`) along with the new insertion class added by TEXsis (`\bottominsert`) . In addition, there are special commands for creating wide or full page figures while in double column mode (which is described later in Section 12):

`\figure{`⟨label⟩`}` Inserts a figure at the top of the current page, if there is enough room for it, or at the top of the next available page.

`\midfigure{`⟨label⟩`}` Inserts a figure at the current position, if there is enough room for it, or at the top of the next page if there is not.

`\fullfigure{`⟨label⟩`}`
: Inserts a figure on a separate page. The body of the figure should normally contain a `\vfill` to allow the figure to stretch to fill the whole page.

`\bottomfigure{`⟨label⟩`}`
: Inserts the figure at the bottom of the current page, if there is room for it, or on the bottom of the next page where there is room.

`\heavyfigure{`⟨label⟩`}`
: Inserts a figure at the current position, if there is room for it on the page, or at the bottom of the page following if there is not. This is therefore like `\midfigure`, except that if the figure can't be fit on the page it ends up on the bottom of a following page, not the top.

`\widetopfigure{`⟨label⟩`}`
: Inserts a figure at the top of a page across both columns in double column format. Equivalent to `\figure` in single column format. `\widefigure` is a synonym.

`\widefullfigure{`⟨label⟩`}`
: Inserts a full page figure across both columns in double column format. Equivalent to `\fullfigure` in single column format.

`\endfigure`
: Ends any of the figure insertions described above.

In all cases ⟨label⟩ is an internal label which is assigned to the figure number. The figure number will include the chapter and section numbers in the format "*cc.ss.nn*" if `\showchaptIDtrue` and `\showsectIDtrue` have been selected.

The body of the figure can include any vertical mode commands, as in the example above. When you are simply leaving space for a figure to be glued in later a convenient way to specify the amount of space to leave is to `\vskip` by some amount proportional to the width of the current column or page. The dimension `\colwidth` is the width of the page (the `\hsize`) in single column mode and the width of the current column in double column mode. So for example, if you have a figure which was half again as tall as it is wide you could say

```
\vskip 1.5\colwidth
```

Another way to standardize the space reserved for figures is to define your own dimension, such as `\myfigsize`, to use in the `\vskip` above. For example,

```
\newdim\myfigsize
\myfigsize= 1.5\colwidth
```

and then `\vskip\myfigsize`. This has the same effect as the previous example, but `\myfigsize` can be changed to any valid dimension.

9.2. Encapsulated PostScript Graphics

Instead of simply gluing figures into a paper by hand it is possible to have computer generated graphs and drawings automatically inserted into a document. Unfortunately, the method can vary radically from computer to computer and printer to printer. Recall that TEX produces as output a DVI file, which contains device independent instructions for typesetting the document. To actually print the document the DVI file must be translated into instructions for the specific printer being used by a suitable device driver. It is at this stage, when the device driver is converting the DVI file to printer instructions, that graphs and drawings may be inserted. However, since different printers may require different kinds of instructions, there are many different device drivers, each having its own way of including drawings in the document.

In recent years some standardization has become possible, because many printer manufacturers have chosen PostScript as the "language" for giving instructions to their printers. PostScript contains commands both for drawing and for typesetting text. Still, there are some problems, because there are many different device drivers for converting DVI files to PostScript, and they all treat the inclusion of figures slightly differently. Also, while all of these filters require that the drawings to be included are themselves already in PostScript, there are several variations of PostScript, leading to further problems of incompatibility.

Until there is some standardization for inserting drawings into TEX documents (and perhaps to encourage it) we recommend using the scheme which (a) seems to be the easiest to use and the most well thought out, and (b) is potentially available to the widest group of users. Our choice is, first, to use a specific variation of PostScript known as "Encapsulated" PostScript (EPS). Most newer computer programs which produce graphics output, including Mathematica, `xfig`, and `paw`, can produce EPS output files. To include these in the text we have chosen to rely on the program `dvips` by Tomas Rokicki,[16] along with his TEX macros for including EPS graphics, called `epsf.tex`. Be warned that there are other programs called "dvips," but as far as we know only Rokicki's version will work with `epsf.tex`. We expect that when including drawings in TEX is standardized, it will be based on Rokicki's scheme or something close to it.

To include EPS graphics in a document using the `epsf.tex` macros you must first input the macros. Somewhere early in your manuscript file you should put:

```
\input epsf.tex
```

You can have the macros report the size of a figure when it is included by saying:

```
\epsfverbosetrue
```

At the point where you want a graph or drawing included you simply say:

```
\epsfbox{foo.ps}
```

This will include the drawing from the file called '`foo.ps`,' which must already exist in the current directory. TEX will leave enough room for the figure (by creating a box of the correct size) and will use a `\special` command to insert instructions into the DVI file telling the DVI to PS translator to read in the EPS file `foo.ps`.

Once the document has been processed by TEXsis and the DVI file has been produced you will convert it to a PS file with the "dvips" program. If your DVI file is called `myfile.dvi` then the command is simply:

```
dvips myfile
```

This will convert `myfile.dvi` to PostScript, and depending on how dvips is set up at your installation it will either put the output in the file `myfile.ps` or it will send it directly to the default printer (see the man pages for dvips on how to change this behavior, or try a short example to see how it is set up on your computer). Note that the EPS files to be included must be in the same directory as `myfile.dvi` when you run dvips, so that they can be merged into the final output file (or you can specify the full path filename in the `\epsfbox` command).

Note that on Unix systems you may not be able to use the `lpr -d` command if you have included EPS files in the document. This command runs the DVI file through dvips (or some other DVI to PS filter) in a different directory, and the EPS files will therefore not be included unless you have used a full path name for the file.

If you use the latest version of the `xdvi` program to preview DVI files on your workstation you will find that it automatically includes EPS graphics in the display, but some other previewing programs (such as `texsun`, or older versions of `xdvi`) will not. Even if the previewer does not know how to deal with the `\special` commands in the DVI file, you will see that TEX has left room for the figure. Once you have converted your DVI and EPS files into a PS file you can preview that file if you have a PostScript previewer, such as GhostScript. Then your graphs and drawings will appear just as they would if you had printed the PS file.

Sometimes you will find that more room has been left for the figure than is really needed. The way `\epsfbox` decides how much room to leave is by reading the first part of the EPS file and looking for a line which looks like:

```
%%BoundingBox: 67 176 544 616
```

The two pairs of numbers are the coordinates of the lower left and upper right corners of the figure, respectively. These are measured in printer's points (1/72nd of an inch). If you don't like the positioning of the figure given by the original bounding box in the PostScript file you can override it by specifying a new bounding box like so:

```
\epsfbox[0 180 540 700]{file.ps}
```

You may have to experiment with positioning the BoundingBox for the best effect, and it may help to look at the PostScript file to see what the default BoundingBox is and adjust things using that coordinate system.

You can also control the size of the included graphic using `\epsfxsize` or `\epsfysize` before you call `\epsfbox`. The figure will be enlarged or reduced to fit the dimension you specify. Saying

```
\epsfxsize=⟨dimen⟩ \epsfbox{figure.ps}
```

will scale the drawing in the file `figure.ps` so that its *width* is ⟨dimen⟩, with the height scaled accordingly, while saying

```
\epsfysize=⟨dimen⟩ \epsfbox{graph.ps}
```

will scale the drawing in the file `graph.ps` so that its *height* is ⟨dimen⟩, with the width scaled accordingly. You can only specify the width or height, but not both at the same time. Whatever change of size you specify will only be applied to the `\epsfbox` command which immediately follows. After that, subsequent included graphics will revert to their natural sizes unless you also specify a particular size for them.

So now consider the earlier example of including a 6 cm tall penguin diagram in a figure, but now assuming that the drawing exists as an EPS graphic in the file `penguin.ps`, and that we need to override the default bounding box in that file. The commands to do this are:

```
\figure{Penguin}
\epsfysize=6cm \epsfbox[0 180 540 700]{penguin.ps}
\Caption
Example of including EPS graphics in a figure.
\endCaption
\endfigure
```

9.3. Simple Tables

Simple tables (without rules) are constructed in TEX with alignment tabs or with `\halign`. The reader who is not familiar with these should look at Chapter 22 in *The TEXbook*. In TEXsis simple tables with table numbers and captions are treated much the same as figures. For example, saying

```
\midtable{table example}
\caption{Charged leptons.}
\centerline{\vbox{\halign{#\ \hfil & #\ \hfil & #\hfil \cr
Name     & Mass     & Lifetime \cr
$e^-$    & .511 MeV & $>2 \times 10^{22}$ yr. \cr
$\mu^-$  & 105 MeV  & $2.197 \times 10^{-6}$ sec. \cr
$\tau^-$ & 1784 MeV & $3.3 \times 10^{-13}$ sec. \cr}}}
\endtable
```

produces the following table:

Table 9.1: Charged leptons.

Name	Mass	Lifetime
e^-	.511 MeV	$> 2 \times 10^{22}$ yr.
μ^-	105 MeV	2.197×10^{-6} sec.
τ^-	1784 MeV	3.3×10^{-13} sec.

Note the use of `\centerline` and a `\vbox` to center the table. The reference to this table in the text was produced by typing "`\Tbl{table example}`." Note also that we used `\caption` followed by the text in brackets as an alternative method of defining the caption; `\caption` should only be used when the caption is short. Table captions are traditionally placed above the table, so `\Caption` or `\caption` should be the first token after `\table{`⟨label⟩`}` and its brothers.

There are a number of different kinds of table insertions, and they match closely the different kinds of insertions used for figures:

`\table{`⟨label⟩`}` Inserts a table at the top of next available page.

`\midtable{`⟨label⟩`}` Inserts a table at the current position if there is room or at the top of a page if necessary.

`\fulltable{`⟨label⟩`}` Inserts a table on a separate page. The body of the table should normally contain a `\vfill`.

`\bottomtable{`⟨label⟩`}`
Inserts the table at the bottom of the current page, if there is room for it, or on the bottom of the next page where there is room.

`\heavytable{`⟨label⟩`}`
: Inserts a table at the current position, if there is room for it on the page, or at the bottom of the page following if there is not. This is therefore like `\midtable`, except that if the table can't be fit on the page it ends up on the bottom of a following page, not the top.

`\widetoptable{`⟨label⟩`}`
: Inserts a table at the top of a page across both columns in double column format. Equivalent to `\table` in single column format.

`\widefulltable{`⟨label⟩`}`
: Inserts a full page table in double column format. Equivalent to `\fulltable` in single column format.

`\endtable`
: Ends any of the tables above.

Again, in these macros ⟨label⟩ is an internal label which is assigned to the table number. The table number will include the chapter and section numbers in the format "*cc.ss.nn*" if `\showchaptIDtrue` and `\showsectIDtrue` have been selected. The table can then be referred to in the text by saying `\Tbl{`⟨label⟩`}`.

9.4. Ruled Tables

While `\halign` in Plain TEX is convenient for making simple tables, it is still difficult to make tables with horizontal and vertical rules. Indeed, typesetting ruled tables in TEX has often been likened to programming a computer in assembler language. Even *The TEXbook* acknowledges that "people who know how to make ruled tables are generally known as TEX Masters."[17] The macros described below take a great deal of the difficulty out of making ruled tables, so that anybody can become a TEX Master, or at least a master of ruled tables.

The basic idea behind making a ruled table is similar to `\halign`, and the syntax is the same as the Princeton table macros known as `TABLES.TEX` by Ray Cowan,[18] although the TEXsis macros are quite different internally and run about 5 times faster. You begin the table material with `\ruledtable` and end it with `\endruledtable`. The table material is a list of items to appear in the rows and columns of the tables. Just as with `\halign` you can separate items in different columns with `&` and end a row with `\cr`. The `&` alignment character separates columns without putting a vertical rule between the columns, while `\cr` ends a line and puts a thin horizontal rule under it. To get a vertical rule between columns replace the `&` with a '`|`' (the vertical line), and to end a line without a horizontal rule under it replace the `\cr` with `\crnorule`. One important thing to remember is to end the last line in the table with `\endruledtable` instead of `\cr`!

As a first example, here is a simple ruled table,

Year	World Population
8000 B.C.	5000000
50 B.C.	200000000
1650 A.D.	500000000
1850 A.D.	1000000000
1945 A.D.	2300000000
1980 A.D.	4400000000

and here are the instructions that created it,

```
\ruledtable
Year      |    World Population \cr
8000 B.C. |   ~~~5000000        \crnorule
~~50 B.C. |   ~200000000        \crnorule
1650 A.D. |   ~500000000        \crnorule
1850 A.D. |   1000000000        \crnorule
1945 A.D. |   2300000000        \crnorule
1980 A.D. |   4400000000
\endruledtable
```

This example also appears in *The TEXbook*, on page 246, where it is typeset using `\halign`. Compare how the two are constructed and see which way you think is easier.

By default each item in the table is centered in its column. Inside a table the tie character "~" takes up the space of exactly one digit (it is the same as saying `\phantom{0}`), and this has been used to line up the digits of the entries in both columns. It is also possible to change the definition of the `\TableItem` macro so that each entry in each column is left or right justified. This is described a little bit later.

Something to note about this example table is that the outer rules are thicker than the others. You can also get a thick vertical rule between columns by using "`\|`" in place of "`|`", and you can get a thick horizontal rule below a row by replacing the `\cr` with `\crthick`. The widths of thick and thin rules are controlled by the dimensions `\thicksize` and `\thinsize`. If you don't want thick rules at all simply say `\thicksize=\thinsize`, which makes the thick rules as thin as the thin rules.

Some synonyms: `\CR` is the same as `\crthick`, `\nr` is the same as `\crnorule`, and `\crrule` is the same as `\cr` (actually it's the other way around). `\tab` is the same as `&` — it separates columns without a vertical rule. `\vb` is the same as `|`, it separates columns with a thin vertical bar, while `\Vb` is the same as `\|` and separates the columns with a thick vertical bar. You can also use `\dbl` to separate columns with a double vertical rule.

As in Plain TEX you can use `\span` in place of `&` (or the other column separators) to join two columns together into one, and you can use `\omit` as the very first token in a column to tell it to omit the fancy column layout (the centering or other spacing). You can also use `\multispan` to `\span` and `\omit` several columns, just as in an `\halign`. The only difference is that the number of columns you tell it to `\multispan` is the number

Table 9.2: Another example of a ruled table.

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	$2.60
2	41-54	$2.70
3	46-55	$2.87
4	40-53	$3.24
5	45-52	$3.40
6	51-59	.95

of columns of the table it spans — in a ruled table constructed with `\halign` you would count the columns *and* any vertical rules.

Table 9.2 is another sample table, again taken from *The TEXbook*. Something important to notice is that the last ruled table example didn't have a table number or a caption, while this one does. That's because the last example was created using just `\ruledtable`, while this example is inside of `\table`. The point is that `\table` and `\ruledtable` do separate things and can be used either separately or together; `\table` creates the space for the table and gives it a caption and a number, while `\ruledtable` actually constructs the body of the ruled table. The instructions that created Table 9.2 are:

```
\table{example-II}
\caption{Another example of a ruled table.}
\singlespaced
\ruledtable
\multispan3\hfil AT\&T Common Stock \hfil\CR
Year \dbl Price | Dividend \cr
1971 \dbl 41-54 | \$2.60   \cr
~~~2 \dbl 41-54 | \$2.70   \cr
~~~3 \dbl 46-55 | \$2.87   \cr
~~~4 \dbl 40-53 | \$3.24   \cr
~~~5 \dbl 45-52 | \$3.40   \cr
~~~6 \dbl 51-59 |  ~~.95
\endruledtable
\endtable
```

Sometimes it is desirable to have rules across only some of the columns of a table. In this case put `\crule` in the columns that are to have rules, `\cskip` in the columns where no rules are desired, and end the line with `\crpart`, which signals that the line contains a partial rule. The column headings in Table 9.3 are an example of partial rules in a table. That table was created with the commands,[19]

```
\table{SiHCl}
\caption{...}
\singlespaced\tenpoint
\ruledtable
```

Table 9.3: Parameters used in the MS-Xα calculations for the chlorosilane molecules, H_xSiCl_{4-x}; x=0–4.

Molecule	Region	Radius (au)	α_{HF}	l_{max}	
				initial state	final state
$SiCl_4$	outersphere	6.21	0.7238	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
$SiHCl_3$	outersphere	6.21	0.7260	4	7
	Si	2.16	0.7275	2	3
	Cl	2.40	0.7233	2	3
	H	1.20	0.7772	1	2
SiH_2Cl_2	outersphere	6.25	0.7295	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
	H	1.20	0.7772	1	2
SiH_3Cl	outersphere	6.30	0.7360	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
SiH_4	outersphere	4.10	0.7524	4	7
	Si	2.04	0.7275	2	3
	H	1.20	0.7772	1	2

```
\vctr{Molecule} | \vctr{Region} | Radius |
  \vctr{$\alpha _{\rm HF}$}|\multispan{2} $l_{\rm max}$
\crnorule
\cskip | \cskip | \cskip |\cskip|    \crule     |  \crule \crpart
       |        | (au)   |      | initial state | final state \CR
SiCl$_4$      | outersphere | 6.21 | 0.7238 | 4 | 7 \crnorule
              | Si          | 2.20 | 0.7275 | 2 | 3 \crnorule
              | Cl          | 2.40 | 0.7233 | 2 | 3 ...
\endruledtable
\endtable
```

Ruled tables are normally centered on the page, but you can turn off the centering with `\noncenteredtables`. Then the table becomes just another piece to add to a horizontal list. One thing this lets you do is have two tables side by side on the page, like so:

Item ABC	Item DEF
Item GHI	Item JKL

Data 111	Data 222
Data 333	Data 444
Data 555	Data 666

This pair of tables was produced by saying:

```
\noncenteredtables
\medskip
\line{\ruledtable
Item ABC | Item DEF \cr
Item GHI | Item JKL \endruledtable
\hfil
\ruledtable
Data 111 | Data 222 \cr
Data 333 | Data 444 \cr
Data 555 | Data 666 \endruledtable}
\smallskip
```

Saying `\centeredtables` turns table centering back on. Tables are then centered across the page, and a table is then added to a *vertical* list.

Usually `\ruledtable` makes tables as wide as their natural width, but it is possible to make them wider by setting the dimension `\tablewidth` to the width desired before saying `\ruledtable`. We can make the same table in Table 9.2 but with the width set to 10cm:

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	$2.60
2	41-54	$2.70
3	46-55	$2.87
4	40-53	$3.24
5	45-52	$3.40
6	51-59	.95*

* (first quarter only)

The careful reader will notice that the version of Table 9.2 in *The TEXbook* had a footnote to the last entry, and we have added that here. To do so we had to put the footnote in an `\hbox` and stack it under the table, so we had to turn off table centering. The table is therefore a bit more complicated to make, but not much:

```
\noncenteredtables
\tablewidth=10cm
\line{\hfil\vbox{\singlespaced
  \ruledtable
  \multispan3\hfill AT\&T Common Stock \hfill\CR
  Year \dbl Price | Dividend \cr
  1971 \dbl 41-54 | \$2.60   \cr
  ~~~2 \dbl 41-54 | \$2.70   \cr
  ~~~3 \dbl 46-55 | \$2.87   \cr
  ~~~4 \dbl 40-53 | \$3.24   \cr
  ~~~5 \dbl 45-52 | \$3.40   \cr
  ~~~6 \dbl 51-59 |  ~~.95\rlap*
  \endruledtable
```

```
  \hbox{* (first quarter only)}
\vss}\hfil}
```

If we had not wanted to add the footnote, then the table would have been typed the same way as Table 9.2 except for setting the `\tablewidth`.

Important note: `\tablewidth` is turned "off" after the table is made, so the next table will have its natural width unless you explicitly set `\tablewidth` again!

`\ruledtable` normally inserts a `\quad` of space around each item. If you have a very wide table, especially one with many columns, you may wish to reduce this. `\TightTables` make this spacing a single space, the minimum acceptable. `\LooseTables` restores the default spacing. You can define this space to be something else by re-defining `\tablespace`, for example as

```
\def\tablespace{\ \ }
```

If the table is still too wide with `\TightTables`, you will probably have to reformat it or print it in landscape mode (tilted 90°on the page).

The careful reader may also have noticed that in the examples in Table 9.2 and Table 9.3 we said `\singlespaced`. That's because the spacing of the rows of a table can be controlled just like the spacing of the text with `\singlespaced`, `\doublespaced`, etc. The space between the rows of a table is set by a strut called `\tstrut`, which holds the rows apart. Its default definition is

```
\def\tstrut{\vrule height 3.1ex depth 1.2ex width 0pt}
```

You can change the spacing between rows by changing the definition of this strut. One of the things `\singlespaced` does is define the `\tstrut` to be 0.5ex higher and deeper than a parenthesis. `\doublespaced`, `\triplespaced`, and `\widenspacing` all increase the height and depth of the strut in proportion to the increase in the baseline spacing.

If you put something in a table which is larger than the strut, then it will touch the rules above or below. You can fix this by inserting your own strut into that row. More simply, you can say `\bigitem{`⟨stuff⟩`}`, and the "stuff" will be positioned correctly with a nice amount of space above and below.

You can also decide whether or not you want diagnostic messages printed when a table is created. Saying `\tableinfotrue` causes a message to be printed on your terminal every time a ruled table is created. The message tells you how many rows and columns were in the table. This is useful information, because it can tell you immediately if you made a mistake in the table or left out a row or column. `\tableinfotrue` is the default, but you can turn off these messages by saying `\tableinfofalse`. In any case, you will be warned if a row has a different number of columns from the previous one.

The tables we have demonstrated so far all contain numbers, but sometimes it is desirable to have an entry in a table which is a paragraph of text. For that purpose, you can use `\para{`⟨text⟩`}`. The width of the paragraph is determined by the dimension `\parasize`, which defaults to 4 inches.

Table 9.4a: Basic table macros for use with `\ruledtable`.

Macro name	*Description*
`\ruledtable`	Starts a ruled table.
`|` (vertical bar) or `\vb`	Separates one column from the next, with a a thin vertical rule between them.
`&` or `\tab`	Separates one column from the next, but with no rule between them.
`\cr`	Ends the current row, and starts the next one. The completed row will be separated from the next with a thin horizontal rule.
`\crthick` or `\CR`	Similar to `\cr`, but the rows will be separated with a thick horizontal rule.
`\crnorule` or `\nr`	Similar to `\cr`, but the rows will not be separated by a rule.
`\\|`	Same as `|`, but with a thick vertical rule.
`\dbl`	Same as `|`, but with two thin vertical rules.
`\endruledtable`	Ends the ruled table.

Continued...

Tables 9.4a and 9.4b on the next two pages summarize the most important commands for making ruled tables.

Finally, we can discuss how you can change how each column of a table is constructed. The macros which build the table take each entry and pass it to a macro called `\TableItem`. The default definition of `\TableItem` is:

```
\def\TableItem#1{%              % centers item in ruled table
   \hfil\tablespace              % left glue
   #1\killspace                  % item, no space after
   \tablespace\hfil              % right glue
    }%
```

The `\hfil` glue on both sides of the argument causes the item to be centered in the column, while `\tablespace` (which defaults to `\quad`) insures that there will be at least some amount of space between the item and the vertical rules on each side. Because of the way that TEX handles macro arguments, if you type a table with spaces between the text and the next column separator, an extra `\space` will be inserted into the table. If you first say `\NoTrailingSpaces`, then `\killspace` in the above macro is defined to remove any trailing glue items. `\hfill` is redefined by the table macros so that if it is the last token in an entry, then that entry is properly left justified, but to insert any other trailing glue, you must say

⟨text⟩⟨glue⟩`\null`

Table 9.4b: Additional macros for use with `\ruledtable`.

Continued...

Macro name	*Description*
`\multispan{`*n*`}`	Makes the next entry span the next *n* columns, where *n* is an integer, $n > 0$. See `\omit` below.
`\omit`	This TEX primitive causes the normal template for its entry to be omitted, allowing the user to do something else with this entry.
`\para{`⟨text⟩`}`	Formats *text* into a neat little paragraph like this one. The width of the paragraph produced is determined by the dimension `\parasize`. The default is 4 inches.
`\parasize=`⟨dimen⟩	Sets the width of paragraphs for the `\para` macro.
`\TableItem{`⟨stuff⟩`}`	Used in the standard template, this macro centers its argument in the column. The user can redefine it for special effects.
`\tablespace`	The spacing around each item for `\TableItem`.
`\LooseTables` `\TightTables`	Set `\tablespace` to `\quad` and `\ ` respectively.
`\TrailingSpaces` `\NoTrailingSpaces`	Use trailing spaces and glue items in table entries (default) or ignore them. See text.
`\bigitem{`⟨stuff⟩`}`	Use this for something that would normally not fit in one line of a table to make the spacing between rows large enough for this item.
`\vctr{`⟨stuff⟩`}`	Centers the item *vertically* so that it can appear between two rows.
`\centeredtables`	Turns table centering on. Each table will be centered left-to-right on the page. This is the default.
`\noncenteredtables`	Turns table centering off. Each table is returned as an `\hbox`, and it is up to the user to place it as desired.
`\tableinfotrue` `\tableinfofalse`	Turns on or off the diagnostic message telling you how many rows and columns were found in the table.
`\thicksize=`⟨dimen⟩	This dimension specifies the thickness of the thick rules in the table. The default size is 1.5 points.
`\thinsize=`⟨dimen⟩	This dimension specifies the thickness of thin rules in the table. The default size is 0.8 points.
`\tablewidth=`⟨dimen⟩	Specifies how wide to make the next table. If not specified, the table is made to its natural width. This value is reset following the construction of each table.

The default is `\TrailingSpaces`, which does incorporate any trailing spaces into the table but which also treats any trailing glue items normally. You may wish to change this default in your `TXSpatch.tex` file.

You can change the definition of `\TableItem` to suit your own purpose. The simplest modification is to omit the `\hfil` on the left or right so that each entry is right or left justified. This is so commonly used that saying `\RightJustifyTables` or `\LeftJustifyTables` makes the appropriate change for you. Saying `\NoJustifyTables` restores `\TableItem` to its original centering function.

Since `\TableItem` is used to create every column of the table, you cannot use it to modify a single column. To justify just one column you can simply put `\hfill` to the left or right of each item in the column, since `\hfill` is infinitely more stretchy than `\hfil`. (The behavior of glue items other than `\hfill` depends on whether `\NoTrailingSpaces` has been set.) With a little more work it is possible to design special or unusual tables by redefining the "preamble" of the table. This is possible because `\ruledtable` uses `\halign` with a preamble called `\TablePreamble`. The default definition of `\TablePreamble` is:

```
\def\TablePreamble{%                    % \ruledtable preamble
   \TableItem{####}%                    % the first item
   \plaintab\plaintab                   % && means repeat this
   \TableItem{####}%                    % the subsequent items
   \plaincr}%                           % end of preamble
```

The syntax for `\TablePreamble` is like an `\halign` preamble in Plain TEX but with `&` replaced by `\plaintab`, `\cr` replaced by `\plaincr`, and `#` replaced by `####`. If you have read this far and want to try changing `\TablePreable` then you should read the source code and the comments that go with it in the file `TXSruled.tex`.

As a final note we point out that it is also possible for you to create your own column separators with the macro `\nextcolumn`. This macro takes a single argument which is used to separate one column of the table from another. Thus for example, the `\dbl` macro for creating a double vertical rule between columns is defined as:

```
\def\dbl{%
   \nextcolumn{\vrule width\thinsize
   \hskip\thinsize\vrule width\thinsize}}
```

9.5. Lists of Figures and Tables

In large technical documents it is often desirable to have a list of the figures and/or tables which appear in the document, along with a brief caption and the number of the page on which the figure or table appears. This is similar to a table of contents, but is separate from it. In TEXsis you can add a figure or table to a list of figures or tables by using `\infiglist` after `\figure` or `\intablelist` after `\table`, similar to the way you would use `\caption`. The syntax is `\infiglist{`⟨text⟩`}`. The text is the text of the (possibly abbreviated) caption which is to appear in the list of figures. Remember though, if a table

has a caption it should appear above the table body, so the `\caption` must still be the first thing after `\table`; the `\infiglist` can come later.

As an example, consider the "penguin diagram" already discussed. To add an entry for this figure in the list of figures it would be entered as follows:

```
\figure{Penguin}
\epsfbox[0 180 540 700]{/home/fermi/qcd/penguin.ps}
\caption{Example of including EPS graphics in a figure.}
\infiglist{Example of including a figure in the list of figures.}
\endfigure
```

The `\caption` text is what is put under the figure in the body of the paper, while the `\infiglist` text is what shows up in the list of figures, and these do not have to be the same.

To produce the list of figures at the end of the document you say `\ListFigures`, and similarly to produce the list of tables say `\ListTables`. Even if the list is to appear at the front of the document, along with the table of contents, you should print it at the end and move the list to the front afterwards. This is because until the document is actually processed TEX does not know on which page the figure or table appears.

One word of caution: figures and tables are produced by putting them in "floating inserts", which TEX saves up and prints when there is room. The entry into the list of figures or tables is not made until the item actually makes its way into the output DVI file. If you ask for the list of figures or tables before one of these items is output it will not be in the list, and in fact that entry for the figure or table list will later end up spewing out on your terminal. To be sure that all floating insertions have been output you should say `\supereject` before you print the list of figures or tables.

9.6. Listing Figure and Table Captions

When a manuscript is submitted to some journals, like the Physical Review, one is supposed to include a separate list of the captions of all figures and tables. This is similar to the lists created by `\ListFigures` and `\ListTables`, but the page numbers are not supposed to be included. It is very easy to use the caption information stored by `\infiglist` and `\intablelist` to make just a list of captions, without including the page numbers. The command to make a list of all the figure captions is `\ListFigureCaptions`, and similarly all the table captions can be listed with `\ListTableCaptions`. Just as with `\ListFigures` and `\ListTables` these should appear at the end of the job, so that all of the caption information will have already been stored.

It should be said that this method of making a list of captions assumes that you have put the same captions under the figures, with `\Caption` or `\caption`, as the caption in the list of figures entered with `\infiglist`.

It may help to know a little bit more about how the macros `\ListFigures` and `\ListFigureCaptions` work. The `\infiglist` command writes the figure caption information you provide to a file which has the same name as your manuscript file but with

the extension `.fgl`, meaning "figure list" (for tables the file name extension is `.tbl`). The page number on which the figure appears is also recorded in this file. When the whole document has been processed all of the figure captions will have been saved in this file, and `\ListFigures` and `\ListFigureCaptions` simply read this file back in. One of them uses the page numbers to construct a table of contents for the figures, while the other just prints out the figure captions by themselves. In either case, it is important that all the caption information be in the file when the list is to be made, which is why a `\supereject` may be necessary to flush all floating insertions to the output.

9.7. Printing Figures and Tables at the End of a Document

When working on a paper to be sent for publication it is sometimes necessary to print all of the figures at the end of the document. To cause this to be done you simply have to say

```
\FiguresLast
```

at some point before you call `\figure` and its relatives, and then put

```
\PrintFigures
```

at the end of the document. `\FiguresLast` causes `\figure` to save the definition of your figure in an auxiliary file instead of immediately creating the figure, and `\PrintFigures` reads the material back in from that file. If you say `\PrintFigures` without first saying `\FiguresLast` nothing will happen. You can return to having the figures printed where they are defined by saying `\FiguresNow`. In that case `\PrintFigures` does nothing; you have to have said `\FiguresLast` for `\PrintFigures` to read any stored figures back in.

Tables can also be saved until the end of the document by saying `\TablesLast` and then `\PrintTables`. `\TablesNow` is the opposite of `\TablesLast`.

One important note: Since the figure or table material is copied to the file in which it is held one line at a time it is important to use `\Caption` and `\endCaption` for long captions instead of `\caption`. If you use `\caption` TEX will try to read the whole caption at once, which may result in the error message "`! TeX capacity exceeded, sorry.`"

9.8. Nonstandard Figure and Table Placement

On rare occasions you may need to have a figure or table from one section of your document appear in another section of the document. For example, if the introductory section is very short but has a figure, that figure might end up at the top of the first page when you really want it to appear at the top of the second page. You can cause the figure to come out later by moving it farther back in the manuscript file, but if it is moved into the next section it will then have the wrong figure number.

The solution to this problem is to have a way to override or modify the way figure and table numbers are assigned. Since the first part of the figure or table number is the chapter or section number you need to change this by saying

```
\setSectionID{⟨something-else⟩}
```

This changes the section number part of the figure and table numbers to whatever you put as ⟨something-else⟩. It will also effect equation numbers, so it is best to have this inside of a group, so that the change goes away at the end of the group. Next you have to change the figure or table number. TEXsis keeps count of tables and figures in global variables, and increments these counters globally when a new figure or table is created. Thus just changing the counter inside a group won't work, because the counter outside of the group would still be modified. To temporarily change the value of such a counter you use

`\setcnt{`⟨counter⟩`}{`⟨value⟩`}`

where ⟨value⟩ is the temporary value to assign to the counter ⟨counter⟩. This should be done inside of a group, and when the group ends the counter will go back to its previous global value. You should only use `\setcnt` once within a given group!

An example will make this procedure clear. Figure 1.17 is a part of Section 1, but we wanted to show it much later in this Manual. It was created in this section by saying:

```
\begingroup       % changes are temporary
\setSectionID{1}  % changes the section ID
\setcnt\fignum{16}% figure counter is 1 less than we want
\bottomfigure{intro.17}
\vskip 2in
\caption{A bizarrely placed figure for Section~\use{sect.intro}}
\infiglist{A bizarrely placed figure for Section~\use{sect.intro}}
\endfigure
\endgroup         % everything goes back the way it was
```

In this case the figure number in the counter `\fignum` is set to one less than the value we want, so that it comes out correct when `\bottomfigure` increments the count. You can use `\setcnt` to modify any of TEXsis's global counters. The complete list is given in Section 6 (where the use of `\use` is also described).

By the way, you should never move a figure out of a *Chapter*, unless you move it to the end of the document, and the appropriate way to do that is with `\FiguresLast`.

Figure 1.17: A bizarrely placed figure for Section 1

10. Style Files and Specialized Formats

Section 7 described the basic layout of a TEXsis document and the `\paper`, `\preprint`, `\Manuscript` and `\book` macros for typesetting simple documents. It is often necessary to modify the details of the layouts for such documents, particularly when submitting camera ready manuscripts for conference proceedings. Since TEXsis is built on plain TEX, you can of course use the full power of TEX to format a document as you please. However, TEXsis contains macros to produce documents in several of the common formats used in the physics community. These and other specialized formats are described in this section.

It would be very inefficient for TEXsis to load the definitions of all of the possible document styles listed below when at most only one of them will be used. TEXsis therefore puts these specialized definitions in files known as "style" files (which generally have filename extensions of "`.txs`") and only loads the particular style file when it is needed. Besides saving space this also allows for quite a bit of flexibility, because it is easy to modify style files or to create new ones. For example, if the proceedings of a conference must be typeset in a particular way (as is often the case) then the commands to set-up a document in exactly the right way can be put in a style file and e-mailed to all conference participants. Another use of specialized style files is to define a `\thesis` format which conforms to the rules and regulations of a particular university.

10.1. Style Files

The following styles for various conference proceeding formats, journals, and thesis formats are distributed with TEXsis 2.16. Use these in place of `\paper` or `\preprint`. The name of the style file from which the definitions are loaded is shown in brackets.

`\AIPproceedings`	Similar to `\paper`, but specialized to the American Institute of Physics (AIP) format. [AIP.txs]
`\IEEEproceedings`	An IEEE format using oversized double column paper. See Section 12. [IEEE.txs]
`\IEEEreduced`	An IEEE format reduced to 75% of full size and printed with 9 pt. type for $8\frac{1}{2}\,\text{in} \times 11\,\text{in}$ paper. [IEEE.txs]
`\NorthHolland`	Format for producing camera-ready copy for proceedings published by North-Holland/Elsevier. This is a single column format with titles and authors left justified and in eleven point type. [Elsevier.txs]
`\NorthHollandTwo`	The North-Holland double column format for oversized paper. [Elsevier.txs]
`\nuclproc`	The "NUCLPROC" style for Nuclear Physics Proceedings. [nuclproc.txs]

`\WorldScientific` Similar to `\paper`, but specialized to the World Scientific proceedings format. [WorldSci.txs]

`\PhysRev` Emulates the page layout and appearance of the *Physical Review*. [PhysRev.txs]

`\PhysRevManuscript` Similar to `\Manuscript` but customized for papers submitted to the *Physical Review*. [PhysRev.txs]

`\thesis` Similar to `\book`, but for printing a thesis. The default `\thesis` format conforms to the requirements of the Yale Graduate School, but it can (and probably should) be modified for your own installation. [thesis.txs]

Other even more specialized styles are available, and some of them can be found on the TEXsis anonymous ftp server, in the directory `/pub/texsis/styles`.

The IEEE and North Holland double column formats in the list above use the double column macros described in Sect. 12. For example, `\NorthHollandTwo` sets up a document with an overall width of 21.3 cm, a page length of 27.1 cm and a column width of 10.0 cm. (Note that this is just about as wide or a little wider than most small laser printers can print, so provisions have been made for shifting the printed page horizontally to print one column at a time if necessary.) A document printed with this format might have the following basic structure:

```
\texsis
\NorthHollandTwo   % set up double columns
\titlepage         % but still single column mode

    ⟨title page material⟩

\endtitlepage      % ends the title material
\doublecolumns     % now start double column mode

    ⟨text of paper⟩

\enddoublecolumns  % balance the final columns
\bye
```

Saying `\NorthHollandTwo` only defines the double column layout; it does not turn on double column mode. The title material is printed in single column format across the full page width, then the main body is printed in double column format once `\doublecolumns` appears. It is important to say `\enddoublecolumns` at the end of the document to balance the columns on the last page. Formatting text in balanced double columns can be quite tricky, so you should read Section 12 carefully before trying to use any of the double column macros.

10.1.1. Physical Review Formats

The `\PhysRevManuscript` style is similar to `\Manuscript` but is customized to satisfy the requirements for manuscripts submitted to the *Physical Review*. It uses true double spacing, and it places the tables and figure captions at the end of the document. By simply switching from this to `\preprint` you should be able to produce a version suitable for distribution as a preprint from the same manuscript file.

The `\PhysRev` and `\PhysRevLett` styles are not intended for submission but rather emulate the page layouts in the *Physical Review* and *Physical Review Letters* respectively. We have attempted to reproduce accurately the double column layout, headlines, footlines, reference style, etc., used in the actual journals. This is still work in progress, but it may be useful both to see how a paper would look in print and as an example for constructing other styles. Setting up the headlines and page numbering requires calling a few extra macros:

```
\texsis
\PhysRev
\TitleBanner{⟨Running Title Headline⟩}
\AuthorBanner{⟨Running Author Headline⟩}
\VolumeBanner{⟨series⟩}{⟨vol⟩}{⟨issue⟩}{⟨page⟩}
\titlepage
...
```

In `\VolumeBanner` the ⟨series⟩ should be one of A, B, C, D, E, for the *Physical Review* or L for *Physical Review Letters*; the volume, issue, and starting page arguments are self-explanatory. Both `\PhysRev` and `\PhysRevLett` typeset the document in double columns, so to make full use of these layouts you will need to read about the double column macros described in Section 12. In particular, you should be aware of the following: in both the `\PhysRev` and `\PhysRevLett` styles, double column mode is begun automatically at the end of the title material (`\endtitlepage`). Since any double column material is inside a group, any definitions made at that point will only be defined inside that group. If you want to define macros for use in the rest of your paper, you should do so before the title material.

10.1.2. Thesis or Dissertation Format

The style file `thesis.txs` contains two alternative definitions of `\thesis`, one for Yale University (called `\YaleThesis`) and the other for the University of Texas at Austin (called `\UTthesis`). The default is that saying `\thesis` gives you `\YaleThesis`, but this is easily changed by saying `\let\thesis=\UTthesis`.

If you are not at Yale or UT you will need to change the macros in `thesis.txs` to create your own version of the `\thesis` macro. Every university has different requirements for how a thesis is typed or typeset, and these requirements often conflict. (Perhaps the first thing you should do if you want to write a thesis using TEXsis is find out if someone else at your university has already done so, and then copy their definition of `\thesis`.) If you end up creating a new style file for your university then we hope that you will make

it available for other students to use. We would also be willing to help, by putting such style files on the TEXsis anonymous ftp server. If you have such a file (or other macros) you would like to contribute, send us e-mail at `texsis@lifshitz.ph.utexas.edu`.

10.2. Creating New Styles

It is very easy to create a new document style and style file. By convention, TEXsis style files end with the filename extension "`.txs`," so all you need to do is put the definitions and macros defining this new document style into such a file, and then read that file in when begining the document. For example, suppose you create a style file called `mine.txs` which contains a document style called `\mystyle`, which is a slightly altered version of `\paper`:

```
\def\mystyle{\paper
   \hsize=⟨dimen⟩
   \vsize=⟨dimen⟩
   \showsectIDfalse
   ...
   }
```

You could then simply read in this file at the begining of the document, with `\input`, and then invoke `\mystyle`, like so,

```
\input mine.txs
\mystyle
```

However, you may have noticed that none of the special document styles mentioned in Section 10.1 were actually loaded with `\input`, although they are in fact read from a style file. This was done using the `\autoload` command. Saying

```
\autoload\AIPproceeings{AIP.txs}
```

defines `\AIPproceeings` to be a simpler instruction which, if invoked, automatically loads the style file `AIP.txs`, which contains the real definition of `\AIPproceedings`. After the style file has been loaded this real `\AIPproceedings` is automatically called. It then appears as if the definition of the macro was in TEXsis's memory all the time, though it really was only loaded from the file when called for.

As a practical example of using `\autoload`, suppose that you are a student at Excited State University and you want to create a customized `\thesis` macro which will load automatically from a style file called `ESUthesis.txs`. You could copy the macros in the file `thesis.txs` to this new file, and then edit them as needed. Then, to make the new definition load automatically you would say

```
\autoload\thesis{ESUthesis.txs}
```

To make this new definition of `\thesis` available to anyone using TEXsis on your machine you would want to put this `\autoload` command in the file `TXSsite.tex` before building the TEXsis format. This is described in more detail in Appendix A. To make the new definition automatcially available just to you in your thesis directory you could put the command in a file called `TXSmods.tex` in that directory.

Important note: many TEXsis style files and source files use the "`@`" character in the names of macros which are to be "hidden" from the casual user. Since "`@`" is not a letter these macros can't normally be used. If you are writing a style file which needs to use these hidden macros you should say `\ATunlock` first, to make "`@`" a letter, and then `\ATlock` when you are done to turn "`@`" back into a non-letter.

10.3. Slides

TEXsis contains a simple `\slides` format for making overhead transparencies or similar material. While TEX is not a convenient language for graphical page layout, it is sometimes useful to be able to convert a paper to transparencies or visa versa.* If you want to use `\slides`, simply start your `.tex` file with

```
\texsis
\slides
...
```

The default is to produce $8\frac{1}{2}$ in $\times$ $10\frac{1}{2}$ in transparencies with 1 in margins in the "portrait" orientation with 24 pt. type and ragged right and bottom margins. `\slides` also defines `\bl` to insert a blank line and `\np` to create a new page, since these are often useful for slides. Since the `\slides` format is very simple, it is not loaded from a style file.

10.4. Twin Page Output

It is sometimes desirable to produce preprints or other documents in a compact format with two pages on each physical page in landscape orientation. While this can be done with `\SetDoubleColumns`, a much simpler approach is sufficient. TEXsis contains a macro, `\twinout`, which redefines the output routine to produce two pages per page without attempting to balance columns or to allow spanned equations, tables, etc., thus avoiding many of the possible problems with the more powerful double column macros. `\twinout` can be used with any of the standard formats described in Section 7 or Section 10 or even with Plain TEX. Since `\twinout` does not select fonts, the user must do so:

```
\texsis
\tenpoint
\def\Tbf{\twelvebf}
\def\tbf{\tenbf}
\twinout
 ⟨text⟩
```

Alternatively, the `\twinfmt` command can be used to produce any TEXsis format in `\twinout` layout with 10 pt fonts. Thus the above commands can be replaced by

* However, at least one of the authors of TEXsis prefers to make neatly lettered transparencies by hand using colored pens rather than using a computer and laser printer.

```
\texsis
\twinfmt
 ⟨text⟩
```

Here `\twinfmt` automatically sets up 10 pt fonts and appropriate values for the title bold face fonts, just as in the previous example. `\twinout` use the following dimensions:

`\twinhsize`	Total width for twin column output.
`\twinvsize`	Column height for twin column output.
`\twincsize`	Column width for twin column output.
`\normhsize`	Normal value of `\hsize` in portrait mode.
`\normvsize`	Normal value of `\vsize` in portrait mode.

All of these have sensible default values for 8.5 in × 11 in paper but can be reset before `\twinout` is called. The "normal" sizes should print properly with the default device driver and printer. Appropriate default values for all these dimensions can be set in `TXSsite.tex`.

`\twinout` puts `\LandscapeSpecial` on each page. This is null by default but can be redefined to be an appropriate `\special` command to produce landscape output with your device driver and printer. For example, with `dvips`[16] and a PostScript printer, use

```
\def\LandscapeSpecial{\special{papersize=11in,8.5in}}
```

A default value can be set in `TXSsite.tex`.

`\twinprint` is a substitute for `\preprint` which sets the title, author, and other titlepage material full size in landscape mode and then switches to `\twinout` twin column output with 10 pt type for the text of the paper. A typical preprint might read:

```
\twinprint
\titlepage
\title
 ⟨paper title⟩
\endtitle
\author
 ⟨author's name⟩
 ⟨address⟩
\endauthor
\abstract
 ⟨text of abstract⟩
\endabstract
\endtitlepage
 ⟨text of paper⟩
\bye
```

This produces a nice, compact format for preprints.

11. Letters, Memos and Referee Reports

The commands `\letter`, `\formletters`, `\memo` and `\referee` invoke special styles for producing letters or memos. The letter formats follow closely the letter format given in Appendix E of *The TEXbook*, although there are some differences and some additions.

11.1. Letters

`\letter` is designed for typing basic business letters on letterhead stationary. It uses the following commands:

`\letter` Initialize the letter format.

`\dated{`⟨date⟩`}`
Sets the current date to be ⟨date⟩ rather than today's date. This redefines `\today` to be the date you give. If omitted, `\today` just gives the current date.

`\letterhead{`⟨phone-ext⟩`}`
Puts the date and phone number ⟨phone-ext⟩ at the top of the page, so that it will appear near the letterhead when the letter is printed on your institutional stationary.

`\address` The name and address of the recipient should follow, in `\obeylines` format. The first line is used as the name of the addressee in a running headline for pages after the first.

`\body` Begins the body of the letter. The line immediately below this should be the opening salutation, (e.g. "Dear Ann Landers,").

`\text` A synonym for `\body`.

`\closing` Begins the closing material, in `\obeylines` format. The first line following is the closing salutaion (e.g. "Sincerely Yours,") and the second is the sender's name. When the letter is printed, room is left for a signature. Optional additional lines can be added after the name for the sender's title, or similar information.

`\annotations`
Optional annotations follow, in `\obeylines` format.

`\cc` Carbon copy. A "cc:" and the names following are printed, properly aligned.

`\Encl` Enclosure. This is like `\cc` but "Encl:" is printed for a list of enclosures.

`\ps` Optional postscript follows. This is not in `\obeylines` format.

`\endletter`
Ends the letter. You can say `\bye` instead.

The `\letter` document format automatically sets up a running headline for all pages after the first, containing the addressee's name (taken from the first line following `\address`), the date, and the page number. If you wish, you can turn off these running headlines by saying `\nopagenumbers`. If you want to change the name of the adressee you can simply redefine the macro called `\addressee` anytime after `\address` has been used (using `\gdef`, so that the re-definition exists outside of the group used to enclose the address). Or, if you don't want the person's name used at all, simply redefine `\addressee` to be empty, like so: `\gdef\addressee{}`.

The use of some of these macros is illustrated by the following example:

```
\texsis
\letter
\dated{1 April 1995}
\letterhead{7734}
\address
Prof.~D. Knuth
Stanford University
Stanford, CA~94305
\gdef\addressee{Knuth, the \TeX\ God}
\body
Dear Prof.~Knuth:

     We enclose the documentation and source code for a \TeX\
macro package which we have developed for physics papers.

\closing
Sincerely yours,
Eric Myers
Frank E.~Paige
\ps This stuff is also available by anonymous ftp.
\endletter
```

By default the closing material is indented about 3.5 inches. The actual amount is set by `\longindent`, which you can change. You should put the change on the same line as the `\closing`, like so:

```
\closing\longindent=0pt
Sincerely yours,
Joe User
```

This example puts the closing material flush with the left margin. You should not put `\longindent=0pt` on the line following, because `\closing` uses that line as the closing salutation (like "Sincerely,").

The default "letterhead" created by `\letterhead` is pretty plain – it just pust the datea and phone number at the top of the page. The idea is that you can put your own office letterhead stationary in the printer and when you print the letter on this paper the date and phone number will be in an appropriate position.* Alternatively, you can create a different letterhead macro to actually print a letterhead at the top of the page. The physics department at Excited State University could have a custom `\ESUletterhead` macro which prints their logo and letterhead (it should still take a phone number as an argument, to make it easier to switch between `\ESUletterhead` and `\letterhead`.) These kind of custom macro definitions should be put in a local site file, as described in Appendix A.

If a letter is somewhat too short or too long to fit on a page, its length can be adjusted indirectly by changing the width of the text. To do this insert `\letterwidth{`⟨width⟩`}` immediately after `\letter` to set the width of the text. Not only does this change the `\hsize` to the size you specify, but it also adjusts the margins and centers the text appropriately.

11.2. Form Letters

The `\formletters` layout makes it simple to send identical or similar letters to a list of people. The body of the form letter is stored in one file using the standard TEXsis letter format, while the control commands and a list of addresses are put in a second file. The body of the letter should begin with `\body` and end with `\endletter`, and it can contain an arbitrary number of control sequences defined for each recipient. It can make use of all the features of the letter format. For example it might use

```
\body
Dear \Whoever:

\if P\Job
     I wish to apply for a postdoctoral position
\else\if F\Job
     I wish to apply for a faculty position
\else
     I wish to apply for a postdoctoral or faculty position
\fi\fi
at your institution.  Enclosed you will find my curriculum vitae,
a list of publications, and ...

\closing
Sincerely yours,
Joe Gradstudent
\Encl Publication list
Curriculum vita
\endletter              % don't say \bye here!
```

* If you can't put stationary in your printer you can copy it onto stationary with a copying machine.

This example addresses the letter to `\Whoever`, and it uses the first text if `\Job` is `P`, the second if `\Job` is `F`, and the third otherwise. Thus form letters are very flexible.

The mailing list file initializes the form letter macros and contains the list of addresses. TEXsis should be run on this file, not on the text of the letter! For the example above it would have the form:

```
\formletters{\letterhead{7734}}{letterbody.tex}

\BL     Prof.\ John Doe
        Department of Physics
        University of Nowhere
        Nowhere, OK 73019 USA
        \V\Whoever{Professor Doe}
        \V\Job{F}\EL

\BL     Dr.\ Buckaroo Banzai
        Physics Department
        The Banzai Institute
        Grovers Mill, NJ 08544
        \V\Whoever{Dr.\ Banzai}
        \V\Job{P}\EL

\BL     ...
        ... \EL
\endform
\bye
```

The first argument of `\formletters` is something like the usual `\letterhead` macro, with your phone number. Depending on your letterhead this might have multiple parameters or might even be several control sequences (all enclosed in a set of curly braces, of course). The second argument to `\formletters` is the name of the file containing the body of the letter. The idea is that the stuff in the first argument gets put into the letter *before* the address, while the body of the letter (read from the file) is what goes after the address. Each letter begins with `\BL` ("Begin Letter") and has the name of the addressee on the same line and the address on the following lines. Any "variables" used in the letter can be defined using `\V`, which is just an abbreviation for `\gdef`. Because `\V` is simply `\gdef`, if you omit `\V` from an address entry, the value defined in a previous entry will be used. If necessary, variables can also be defined before the `\BL`. Each address in the mailing list is ended by `\EL` ("End Letter"). When the `\EL` is executed it reads in the file containing the text for the letter and actually produces the letter.

Any address in the list can be ignored simply by replacing the `\BL` with `\IL` ("Ignore Letter"). In particular, this can be done by a simple global edit after each batch of letters is made, and the same file can then be used to send a second mailing by changing `\IL` back to `\BL` for a select subset of addresses.

Mailing Labels

The same address file can be used to produce mailing labels instead of letters simply by replacing `\formletters{...}{...}` with `\formlabels`. It is assumed that there are three columns of names and addresses to fit on standard photocopy machine mailing labels. With the default parameters nine labels each 1 in. high are printed per column with the top and bottom rows left blank. This can be changed, and the alignment of the labels can be adjusted, by setting appropriate values for the following dimensions:

```
\fullHsize=8.500in    % hsize for label page
\fullVsize=9.999in    % vsize for label page
\lblHsize=2.833in     % width of standard label
\lblVsize=1.000in     % height of standard label
\lblVoffset=0.100in   % vertical starting position
\lblHoffset=-0.500in  % horizontal starting position
```

Choosing appropriate values may require some experimentation with your printer and labels. When the correct values are found, they can be set in the file `TXSsite.tex` before compiling the format (see Appendix A).

11.3. Memos

`\memo` sets up a special format for memoranda, including a header at the top of the page with the name of your organization and the word "MEMORANDUM". The organization name defaults to `\ORGANIZATION`, which you can change if you like simply by saying

`\def\ORGANIZATION{`⟨name⟩`}`

(This can be done once in the file `TXSsite.tex` before compiling TEXsis, as described in Appendix A, or you can explicitly put the name of your organization in your file.) The commands for the memo format are:

`\memo` Initialize TEXsis for memo format.

`\Date` Prints the word "Date:" and the date, which must be on the same line. If no date is given, then the current date is used, as given by `\today`. If you would rather have the date in the upper right hand corner, as in a letter, use `\letterdate` instead of `\Date`.

`\To` Prints the word "To:" and the name(s) which follow.

`\From` Prints the word "From:" and the name(s) which follow

`\Subj` Prints the word "Subject:" and the subject which follows. The subject may be several lines long. The first line of the subject field is used in a running headline at the top of all pages following the first.

`\n` New line for `\To`, `\From`, or `\Subj`.

`\nopagenumbers`
: Turns off the running subject headlines and the page numbers. This must be typed after `\memo` since it is redefined by the memo format.

`\Text`
: Begins the text of the memo. You can also say `\body`.

`\annotations`
: Optional annotations follow, in `\obeylines` format.

`\cc`
: Carbon copy. A "cc:" and the names following it are printed, properly aligned in `\obeylines` format.

`\Distribution`
: This is like `\cc` but prints "Distribution:" instead.

`\Encl`
: This is like `\cc` but prints "Encl:" instead, as for letters.

`\ps`
: Optional postscript follows. This is not in `\obeylines` format.

`\endmemo`
: Ends the memo. You can say `\bye` instead.

All of these commands with mixed case names have synonyms which are the same name but in all lower case letters. The use of the `\memo` format is illustrated by the following example:

```
\memo
\To   \TeXsis\ Users
\From E. Myers and F.E. Paige
\Date
\Subj \TeXsis\ 2.16
\text
        \TeXsis\ 2.16 is now available. The new version adds
several features and corrects various bugs.
\Distribution
A. Blake
I. Harrity
B. Nichols
\endmemo
```

If the memo is longer than one page a headline containing the subject and page number is put across the top of all pages after the first. If the subject is longer than one line only the first line of the subject is used in this headline. You can easily change what is used as this subject line by changing the definition of the macro `\subjectline` anywhere in the body of the memo. For example, although the subject of your memo might be simply "Annual Report," you might make the headline more specific by defining:

```
\def\subjectline{Annual Activity Report for Buckaroo Banzai}
```

Then this longer title will appear with the page number on all pages after the first.

11.4. Referee Reports

The `\referee` format creates a document very much like a `\memo`, although there are some important differences. Referee reports are supposed to be confidential, so instead of printing the name of your organization at the top of the first page it puts the words "Referee Report." In the `\referee` format you can use the same commands that are used in the `\memo` format, such as `\To` and `\Date`, but if you use `\From` a warning message will be printed on your screen. (This is only a warning, it still does what it has been asked to do!)

There are also some additional commands for the referee report format:

`\title`	The title of the paper being reviewed should follow this on the same line. This is similar to `\Subj` in the memo format.
`\author`	Prints "AUTHOR:" and the name that follows on the same line.
`\authors`	Same as `\author` for a paper with more than one author.
`\MSref`	Prints "MANUSCRIPT:" and the manuscript reference number which follows.

The text of the report should begin with `\text` or `\body`, just as for a memo, and the whole document should end with `\endmemo` or `\bye`. An example of the referee report format is given by the following:

```
\referee
\date    1 April 1995
\title   \TeXsis\ -- \TeX\ Macros for Physicists
\authors E.~Myers and F.E.~Paige
\MSref   LP-3105
\text
This is pretty good, but it's too long
for Physical Review Letters.
\bye
```

This produces a report which looks something like:

Referee Report

DATE: 1 April 1995

TITLE: TEXsis – TEX Macros for Physicists

AUTHORS: E. Myers and F.E. Paige

MANUSCRIPT: LP-3105

This is pretty good, but it's too long for Physical Review Letters.

12. Double Column Output

Typesetting text in more than one column on a single page is more involved than typesetting text in a single page-wide column, but this extra effort can be rewarded with a document which looks much more professional and which can be easier to read (provided that the resolution of your printer is up to it). Many publishers of "camera-ready" conference proceedings require double column typesetting, and the *Physical Review* and other APS and AIP journals are typeset in double columns. In order to make it easy for authors to prepare manuscripts either for such camera-ready conference proceedings or for submission to *Physical Review*, we have included some powerful double column typesetting macros in TEXsis. These macros can automatically balance columns which do not fill an entire page, and there are also provisions for equations which must be extended across both columns because they are too wide for a single column. Figures and tables (or other such "floating" insertions) can be put either in a single column or across both columns. The same is true for footnotes. By changing a single instruction in the manuscript file it is also possible to take a document which has been prepared for double columns and to print it in single column mode. The converse is not true — typesetting text in double columns with automatic column balancing is a nontrivial task. While the macros generally behave as intended, more attention to the layout of the document is required, and even then the macros will sometimes fail without some extra guidance. It is not possible to divide three lines into two balanced columns! In such cases you must rearrange text or the placement of figures and tables, or explicitly select column or page breaks elsewhere in the document, in order to give TEXsis some extra guidance on how best to format the page.

In order to use the double column formatting macros you must first specify the size of the columns with the command `\SetDoubleColumns{`⟨c-width⟩`}`. The width and height of the full page are taken from the current settings of `\hsize` and `\vsize`, while the width of each column is ⟨c-width⟩. This re-defines TEX's `\output` routine, so it should be called shortly after you say `\texsis`, before any output is actually produced, and it should only be called if double column output is to be produced.

`\SetDoubleColumns` only sets the column dimensions and prepares the `\output` routine for double column output at some point in the future — it does not actually put you in double column mode. Text will still be set in a single page-wide column until you say `\doublecolumns`. After you say `\doublecolumns` text will be set in double columns until you say `\enddoublecolumns`. This will cause the text collected so far to be put into two "balanced" columns of the same height. You can, if you desire, return to double column mode at any later time by saying `\doublecolumns` again, and you can switch back and forth as much as you like. However, keep in mind that frequently switching between single and double columns will make it very difficult for TEXsis to choose good points to break columns and pages, and you may get an error message warning you that a column cannot be balanced.

A simple example of the use of these double column commands is the following:

```
\SetDoubleColumns{0.47\hsize}
\doublecolumns\tenpoint
{\bf BISMUTH} (Ger.\ {\it Weisse Masse}, white mass; later
```

```
{\it Wismuth} and {\it Bisemutum}, Bi; at.~wt. 308.981;
at.~no. 83; m.p.  271.3$^\circ$C; b.p. 1560 $\pm$ 5$^\circ$C;
sp.~gr. 9.747 (20$^\circ$); valence 3 or~5.  In early times
bismuth was confused with tin and lead.  Claude Geoffroy the
Younger showed it to be distinct from lead . . .
     . . . . . .
. . .   High purity bismuth metal costs about \$4/lb.
\enddoublecolumns
```

Here the column size is chosen to be slightly less than half of the `\hsize` (0.47 times the `\hsize`) to allow some space (called the "gutter") between the two columns. The result is the following:[20]

BISMUTH (Ger. *Weisse Masse*, white mass; later *Wismuth* and *Bisemutum*, Bi; at. wt. 308.981; at. no. 83; m.p. 271.3°C; b.p. 1560 ± 5°C; sp. gr. 9.747 (20°); valence 3 or 5. In early times bismuth was confused with tin and lead. Claude Geoffroy the Younger showed it to be distinct from lead in 1753. It is a white, crystalline, brittle metal with a pinkish tinge. It occurs native. The most important ores are *bismuthinite* or *bismuth glance* (Bi_2S_3) and *bismite* (Bi_2O_3). Peru, Japan, Mexico, Bolivia, and Canada are major bismuth producers. Much of the bismuth produced in the U.S. is obtained as a by-product in refining lead, copper, tin, silver, and gold ores. Bismuth is the most diamagnetic of all metals, and the thermal conductivity is lower than any metal, except mercury. It has a high electrical resistance, and has the highest Hall effect of any metal (i.e., greatest increase in electrical resistance when placed in a magnetic field). "Bismanol" is a permanent magnet of high coercive force, made of MnBi, by the U.S. Naval Ordnance Laboratory. Bismuth expands 3.32% on solidification. This property makes bismuth alloys particularly suited to the making of sharp castings of objects subject to damage by high temperatures. With other metals, such as tin, cadmium, etc., bismuth forms low-melting alloys which are extensively used for safety devices used in fire detection and extinguishing systems. Bismuth is used in producing malleable irons and is finding use as a catalyst for making acrylic fibers. When bismuth is heated in air it burns with a blue flame forming yellow fumes of the oxide. The metal is also used as a thermocouple material (has highest negativity known), and has found application as a carrier for U^{235} or U^{238} fuel in atomic reactors. Its soluble salts are characterized by forming insoluble basic salts on the addition of water, a property sometimes used in detection work. Bismuth oxychloride is used extensively in cosmetics. Bismuth subnitrate and subcarbonate are used in medicine. High purity bismuth metal costs about $4/lb.

The `\IEEE`, `\PhysRev`, and `\NorthHollandTwo` document formats described in Section 10 also use these double column macros.

Any document prepared for double column output using the macros just described can also be printed in the simpler single column mode very easily, simply by leaving out the `\SetDoubleColumns`. Without the initialization provided by `\SetDoubleColumns` the `\doublecolumns` and `\enddoublecolumns` commands do nothing (they are, in fact, the same as saying `\relax`). However, if you restore the `\SetDoubleColumns` command, or invoke one of the specialized document styles which use double column outut (like `\PhysRev`) they will come back to life and perform as required.

In double column mode displayed equations that fit within a single column are typed within the usual `$$ ... $$`. If an equation is too long then it may have to be printed across both columns. This is easily done by saying `\longequation` before the `$$` which begins the equation and `\endlongequation` after the `$$` which ends the equation, like so:

```
\longequation
$$
\psi_{nlm}(r,\theta,\phi)
        = [({2Z \over n a_0})^3 {(n-l-1)! \over 2n(n+l)!} ]^{1 \over 2}
           \, (\rho)^l \, e^{-\rho / 2} \, \,
        L_{n-l-1}^{2l+1} (\rho)  \,\,  Y_l^m(\theta,\phi)
$$
\endlongequation
```

The command `\longequation` invokes `\enddoublecolumns` to end double column mode and balance the partial double columns, and it will also draw a ruled line across the bottom of the left column. After the equation has been completed `\endlongequation` invokes `\doublecolumns` to go back to double column mode, and it also draws a rule across the top of the right column.

The purpose of these ruled lines is to guide the reader's eye from the incomplete left column to the right column above the equation, and to the top of the incomplete right column from the left column below the long equation. The default style of these rules is the same as is used by *Physical Review*. The rules are drawn by macros called `\leftcolrule` and `\rightcolrule`, and these can be changed either by you or by a special document format read from a style file. For example, if you don't want these rules you can turn them off simply by redefining these macros to be `\relax`:

```
\def\leftcolrule{\relax}
\def\rightcolrule{\relax}
```

As with the other double column macros, `\longequation` and `\endlongequation` do nothing if they are invoked when double column mode has not been initialized with `\SetDoubleColumns`, to make it easy to print the document in single column mode.

Keep in mind that when dealing with long equations which span both columns, switching frequently between single and double columns will almost certainly cause trouble in balancing the double columns. Also note that each separate piece of double-column material forms its own group, so any definitions made inside that group will be forgotten outside it. To avoid this problem put all definitions at the beginning of the paper, before double column mode is begun. Alternatively, make your definitions global by using `\gdef` rather than `\def`.

Figures, tables, or other floating insertions can be inserted within a single column using the usual insertion commands of TEX and TEXsis: `\topinsert`, `\midinsert`, `\pageinsert`, `\bottominsert`, or `\heavyinsert`, all of which are terminated as usual by `\endinsert`. A `\midinsert` or `\heavyinsert` will appear in the column where it is called for, unless it won't fit, in which case it migrates to the next *column* (rather than all the way to the next page). You can force an insertion into a particular column by placing *within* the insertion either `\forceleft` or `\forceright`. Since insertions which have been forced into the left or right columns are handled separately, an insertion which fails to fit on a page is moved to the same column of the next page and not to the next column. This means that if there are several figures per page, then the figures can easily appear in the wrong order. Also, a `\topinsert` which follows a wide equation on a page will be placed after the equation,

not at the top of the page. Such problems can easily be corrected by moving the insertion to a better place in the manuscript file.

Besides putting insertions in one column or the other, it is also sometimes desirable to be able to put an insertion across the full width of the page. These "wide" insertions can be made with "wide" versions of some of the the insertions already mentioned: `\widetopinsert`, `\widepageinsert` and `\widebottominsert` (there is no such thing as `\widemidinsert` or `\wideheavyinsert`). These wide insertions behave just like their more conventional counterparts when in single column mode.

The standard TEXsis macros for table and figure insertions described in Section 9.3 use `\topinsert` and cousins, and so make insertions within a single column in double column format. There are also macros to make two-column wide tables and figures:

`\widetable{`⟨label⟩`}`
: Insert a two-column wide table, with the tag ⟨label⟩, at the top of the current page.

`\widefigure{`⟨label⟩`}`
: Insert a two-column wide figure, with the tag ⟨label⟩, at the top of the current page.

`\widetopfigure{`⟨label⟩`}`
: A synonym for `\widefigure`.

`\widefulltable{`⟨label⟩`}`
: Insert a two-column wide, full-page table, with the tag ⟨label⟩.

`\widefullfigure{`⟨label⟩`}`
: Insert a two-column wide, full-page figure, with the tag ⟨label⟩.

These also behave correctly in single column document formats.

Here is a summary of the commands which can be used to set-up and control double column mode:

`\SetDoubleColumns{`⟨c-width⟩`}`
: Set up the double column format, using the current `\hsize` and `\vsize` for the total size of the page and ⟨c-width⟩ for the column width.

`\doublecolumns` Begin double column mode.

`\newcolumn` Force a column break, much like `\newpage` forces a page break.

`\enddoublecolumns`
: End double column mode, making balanced double columns,

`\longequation` Begins a "long" equation which cannot fit within one column and therefore must stretch across both columns. This balances the columns, putting a `\leftcolrule` below the left column, and then enters single column mode.

`\endlongequation`
Ends a long equation, returning to double column mode and putting a `\rightcolrule` above the right column of the text following.

13. Miscellaneous Macros

There are a number of useful macros in TEXsis which don't fit into any of the topics covered in previous sections but which still deserve some discussion. We will simply lump them all together in this final section.

13.1. `\draft` mode

Saying `\draft` turns on a number of features which are useful for making a draft copy of a document. The disclaimer "`Preliminary Draft`" is added to the bottom of each page of the document, along with the page number and a time and date stamp. The page number will let you keep track of the page order even when using a document format which does not print page numbers (such as the `\nuclproc` or `\NorthHollandTwo` camera-ready formats). The date/time stamp can help you tell later drafts from earlier versions of the same document. `\draft` sets `\eqnotracetrue` so that equations are marked with their internal labels in the right margin next to the equation number (as described in Section 4). Other kinds of errors, such as trying to use undefined labels for citations equations, are also marked in the right-hand margin. `\draft` also sets `\overfullrule=1em` so that over full `\hbox`es are marked with a black box. Without `\draft` initializing TEXsis with `\texsis` sets `\overfullrule=0pt` so that over-full hboxes don't get these kinds of marks.

13.2. Ignoring blocks of text

Sometimes you'd really like to take out a whole paragraph or an even larger section of a document, but only temporarily, and you don't want to actually delete it from the manuscript file. Recalling that in TEX everything on a line following the comment character "`%`" is discarded, you could put a `%` on each line to be ignored, but that could be a lot of work. Instead, you can use `\comment`. The syntax is:

```
\comment/*  anything in here is ignored */
```

Everything between the `/*` and the `*/` is completely ignored, while by removing the `\comment/*` and the `*/` the text can easily be restored to it's previous state. You can include several paragraphs of text in the material to be ignored, or even several pages, but with one important restriction: All the curly brackets `{` and `}` must match in the text to be ignored. If there is a `{` with no closing `}` in the commented text then TEX will skip the `*/` while searching for the closing `}`!

Along the same lines is `\Ignore`, which will ignore everything following it up to `\endIgnore` (with the same restrictions above about matching grouping brackets). The useful thing about these is that if an `\endIgnore` is encountered before an `\Ignore` then the `\endIgnore` is simply ignored. Is that too confusing? What this means is that if you have a document with a large block of conditional text that you may or may not want included you can choose to include the text by simply removing (or commenting out) the `\Ignore`. If the block of conditional text looks like

```
⟨unconditional text⟩
\Ignore
Having obtained these eigenvalues we would like to
know their degeneracies.  For a given $l$ the ...
...
It is relatively easy to show that the number of
independent components of such a tensor is ${N+l \choose l}$.
\endIgnore
⟨more unconditional text⟩
```

then even if the block of text to be ignored spans many pages it can be restored simply by changing the "`\Ignore`" to "`%\Ignore`." Then the `\Ignore` is itself ignored, and when TEX reaches the `\endIgnore` this also does nothing. The effect is the same as if these two commands were completely removed from the manuscript file, but it is easily reversed.

Why are there two ways to ignore blocks of text? `\comment` was created first, and some programmer-types may prefer it because of the similarity it bears to comments in the C programming language. `\Ignore` is newer and probably a little more useful, but `\comment` will remain for those who want to use it.

13.3. Checkpoint/Restart

The checkpoint/restart feature allows long documents to be printed in pieces. You can, for example, process and print the first chapter of a thesis (or a book) and then later do the same for following chapters. Saying `\checkpoint{NAME}` saves the settings of important counters like the current page number and the last equation number in a file called `NAME.chk`. Saying `\restart{NAME}` reads in this file and resets the counters to their previous values.

As an example consider a document with two major chapters. Using a Master File you would first process the first chapters with the following:

```
\thesis
\input chap1
\checkpoint{chap1}
\bye
```

With this printed you could resume at the next page number with the correct subsequent numbers for equations, figures and tables by saying:

```
\thesis
\restart{chap1}
\input chap2
\bye
```

Normally `\checkpoint` and `\restart` are silent, but you can say `\endstat` to see the values of the counters saved by `\checkpoint`. The output looks like this:

```
% Last PAGE      number IS 63.
% Last CHAPTER   number IS 0.
% Last EQUATION  number IS 47.
% Last FIGURE    number IS 1.
% Last REFERENCE number IS 9.
% Last SECTION   number IS 13.
% Last TABLE     number IS 0.
```

Modern laser printer drivers or filter programs like `dvips`, `dvi2ps`, or QTeX have options that let you select which pages of a document are printed, or there is a useful program called `dviselect`, which can select certain pages out of a dvi file into a smaller dvi file. These offer other ways to selectively print only part of a document and still get the page and equation numbers correct.

13.4. Other Goodies...

`\topspace` Use this like `\vskip` to create a blank space to the top of a page (because any space inserted with `\vskip` is thrown out if it appears at the top of a page).

`\unobeylines`
The opposite of `\obeylines`, so line endings are no longer respected.

`\unobeyspaces`
The opposite of `\obeyspaces`, so that spaces in the manuscript file are not counted exactly.

`\pagecheck{`⟨dimen⟩`}`
Checks the amount of space remaining on the current page, and if this is less then ⟨dimen⟩ then skips to a new page.

`\leftpar{`⟨text⟩`}`
Turns the ⟨text⟩ into a paragraph with a ragged *left* margin.

`\jtem{`⟨label⟩`}{`⟨dimen⟩`}`
Similar to plain TEX's `\item`, but sets the label `{`⟨label⟩`}` with a variable hanging indentation `{`⟨dimen⟩`}`.

`\undertext{`⟨text⟩`}`
Underlines the ⟨text⟩, as in this example. This is generally okay for words but not phrases or sentences.

`\theBlank{`⟨length⟩`}`

Creates an underline of the specified length where one can "fill in the blank." For example, saying "`Name: \theBlank{6cm}` produces: "Name: _____".

`\tightbox{`⟨box⟩`}`
: Puts a ruled box around the argument (which must be a box), with no intervening space. Example: `\tightbox{\hbox{Boo}}` produces "Boo",

`\loosebox{`⟨box⟩`}`
: Puts a "loose" box around the ⟨box⟩, with about one `\jot` of space between the edge of the ⟨box⟩ and the enclosing box. What you do with this resulting box is up to you. Rules are *not* put around this outer box, but you can enclose it in a `\tightbox` to do so, as in `\tightbox{\loosebox{Boo}}`, which gives "Boo."

`\eqnbox{`⟨math-expr⟩`}`
: Puts a box around the mathematical expression ⟨math-expr⟩. This can be useful for highlighting an important result (though be careful not to over-use this!). You can use this in any kind of math mode, but usually you will want to use it in a displayed equation, like so:

```
$$
\eqnbox{E=mc^2}
\eqno(69)$$
```

which produces:

$$\boxed{E = mc^2} \tag{69}$$

In an in-line math expression it appears like so: $\boxed{E = mc^2}$.

`\eqnmark` Putting a box around an equation can be to distracting, but it is nice to have a way to mark "important" results. This is an alternative to `\eqnbox`. Simply put `\eqnmark` at the begining of the displayed equation, immediately after the `$$`, and it will put a marker defined by `\eqnmarker` at the left edge of the equation (the default is "▷"). For example, typing

```
$$\eqnmark
E=mc^2
\eqno(69)$$
```

produces

▷ $$E = mc^2 \tag{69}$$

Note that `\eqnmark` does not work with `\eqalignno` or `\EQNalign`.

A. Customization and Installation of TEXsis

(TEXsis version 2.16, as of 12 November 1994)

This appendix contains information about how to install TEXsis on several types of computer systems, and how to customize TEXsis for a particular installation. The discussion below on "Local Modifications" may be of interest to any user, but the rest of this Appendix is probably only of interest to someone who is responsible for installing or maintaining TEXsis.

A.1. Customization

There are three different mechanisms for making modifications to the basic TEXsis macro package. It is important to use one of these mechanisms rather than modifying the original code.

A.1.1. Local Modifications

TEXsis will read the file `TXSmods.tex`, if it exists, from the current directory or search path every time you run TEXsis. This lets you make your own custom modifications or additions to TEXsis. For example, say you have a directory specifically for a particular project you are working on and you are writing several different papers or other documents related to this project. You could add a file called `TXSmods.tex` to this directory to contain extra macro definitions that are used in all of these documents, and these will be loaded automatically whenever you run TEXsis in this directory. If you run TEXsis from another directory these macros will not be loaded.

A.1.2. Site Modifications

If a file called `TXSsite.tex` exits in the current directory at the time the TEXsis format file is created (see below) then it is automatically read in at this time. The purpose of this file is to allow you to add definitions and macros that are local to your computer installation. For example, in this file you would define the control word `\ORGANIZATION` to give the name of your organization or institution. `\ORGANIZATION` is used at the top of memos and at the top of the title page of a `\preprint`. You might also put a customized `\letterhead` macro in this file, and you may well want to change the definition of `\LandscapeSpecial`, which controls how your laser printer is put into landscape mode. Note that the file `TXSsite.tex` is only read *once*, when the TEXsis format file is created, not every time TEXsis is run. A prototype site file, called `TXSsite.000`, is included in the standard distribution of TEXsis.

A.1.3. Patches

TEXsis will also automatically read in a file called `TXSpatch.tex` from the standard input path, if it exists, every time TEXsis is run. This file should contain additions or corrections ("patches") to TEXsis. This patch file lets you correct minor problems or make minor changes to TEXsis without having to rebuild the format file.

Whenever we find a problem with TEXsis we try to correct it as soon as possible. To make the correction available to users before the next version of TEXsis is released we issue patch files containing the corrections. The most current patch file should always be available via anonymous ftp on `lifshitz.ph.utexas.edu` in the directory `/pub/texsis`. (Patches may also be available on CTAN, but `lifshitz` is always where the most current version of TEXsis is kept.)

A.2. Installation of TEXsis

To use TEXsis or any large macro package effectively you should run the `initex` program to create a pre-compiled format. The general method of doing this is machine independent, but the specific commands depend on the operating system on your computer. This appendix gives the appropriate commands for running `initex` and installing TEXsis on a computer running UNIX, a VAX running VMS, and for an IBM PC running PCTEX under MS-DOS. We assume that you are familiar with the operating system and that TEX and the Plain format have already been installed. Notes on how to install TEXsis on other kinds of machines (such as Atari) are available on the TEXsis ftp server (`lifshitz.ph.utexas.edu`).

A.2.1. Quick Install for Unix and TEX 3.0

If you do not have a Unix system with TEX 3.0 (or greater) then you will have to use the detailed installation instructions given below. If you do have a Unix machine running TEX 3.0 or greater then you can install TEXsis simply by unpacking the files into a working directory (e.g. `/usr/local/lib/tex/texsis`) and:

- 1. Edit the file `Makefile` and change parameters as appropriate, following the directions given there;
- 2. (Optional) Copy the file `TXSsite.000` to `TXSsite.tex` and edit this file to contain appropriate site dependent information, such as definitions for the name of your `\ORGANIZATION`, or your `\letterhead`, or change `\LandscapeSpecial`.
- 3. Give the command '`make install`' (or simply '`make`' first to be sure it works, and then '`make install`').

A.2.2. Detailed Installation

To install TEXsis follow the steps below or their equivalents for your computer and operating system. If you are installing a system-wide version of TEXsis then you may want to review the appropriate documentation on how TEX itself is installed on your system.

☐ 1. Create a separate directory to hold the TEXsis files. This is not strictly necessary, but it is strongly recommended, since there are more than 70 files for the source complete code and documentation. On a Unix machine we recommend using `/usr/local/lib/tex/texsis` or `/usr/local/tex/texsis`, depending on where your other TEX files are kept. For a VAX/VMS system you should use a subdirectory of the main TEX directory. On an IBM PC it is best to use a top-level directory `C:\TEXSIS` on your hard disk, rather than a subdirectory of `C:\PCTEX`, to simplify backing up your files.

☐ 2. Copy all of the TEXsis files into this directory. The TEXsis source files all have names of the form `TXS`*xxx*`.tex`, and most of the documentation files have names `TXS`*xxx*`.doc`. TEXsis style files end with the extension `.txs`. There will also be several other files in your distribution of TEXsis, such as `texsis.tex` and `Manual.tex`.

☐ 3. Make any necessary changes in the source code. You will probably want to copy the file `TXSsite.000` to `TXSsite.tex` and edit this file, changing the definition of `\ORGANIZATION`, `\LandscapeSpecial`, and the letterhead macros and (on VMS) defining `\TeXsisLib` to be the TEXsis directory and `\TXSpatches` to be the patch file in that directory. In some cases it may also be necessary to change some of the font definitions in the file `TXSfonts.tex`. Most distributions of TEX now use the "CM" (Computer Modern) fonts, but some places still use "AM" (Almost Modern) fonts. If you have problems with missing fonts look at the documentation in the file `TXSfonts.tex` and make appropriate changes.

☐ 4. Create the format file `texsis.fmt` by running `initex`. On a Unix machine the command is:

```
% initex texsis \&plain '\dump'
```

You must remember the "\" to escape the "&." On a VMS system the command is essentially the same:

```
$ INITEX TEXSIS &PLAIN "\dump"
```

For an IBM PC with PCTEX give the command,

```
TEX TEXSIS /I
```

and reply to the * prompt with "`\dump`." (In some versions of PCTEX the `TEX` command is replaced with `TEX386` or `TEX386b`. Check your manual to be sure.) TEXsis will also work with `emtex` or any other version of "big" TEX for the PC, but you will have to check the documentation for that particular implementation of TEX for specific instructions for creating a preloaded format.

In any case `initex` will load the Plain TEX format and then the various pieces of TEXsis, displaying a message for each piece. Finally it will spew out a list of fonts used and other information. This should contain near the end the

statement "No pages of output produced." (If it does not, something may be wrong with your distribution, although it is probably a minor problem. Check your editing of the `TXSsite.tex` file.) The preloaded format will be written to the file `texsis.fmt`.

☐ 5. Copy or move various files to where they need to be:

5.1. Copy the format file to the place where your version of TEX looks for pre-loaded formats. On Unix systems this is usually

```
/usr/local/lib/tex/formats.
```

or some variation of this. For VMS machines make sure the file is in `TEX$FORMATS`. With PC-TEX the file goes into `\PCTEX\TEXFMTS`.

5.2. Copy the TEXsis style files (which end with the extension `.txs`) to the place your version of TEX usually looks for input. Under Unix this should be `/usr/local/lib/tex/inputs` (or sometimes `/usr/local/lib/tex/macros`). For VMS machines make sure these files are in the directory `TEX$INPUTS`. For PC-TEX the proper place is `\PCTEX\TEXINPUT`.

5.3. If you have any run-time patches to add to TEXsis copy the file `TXSpatch.tex` to the same TEX inputs directory. If there is a `TXSpatch.tex` from a previous release of TEXsis you should be sure to remove the old file if it is not being replaced.

☐ 6. Having created the format file, you need to make a command `texsis` to run TEX using it. The way to do this is very dependent on the operating system.

6.1. On a Unix system running TEX 3.0 or higher you need only to make a link to the executable `virtex` with the same name as the format you want to load. `virtex` probably lives in `/usr/local/bin` or some similar place. The command to make the link is something like:

```
ln /usr/local/bin/virtex /usr/local/bin/texsis
```

For TEX versions 2.9x you can build an "undumped" version of TEXsis. You will have to see the TEX distribution notes for a description of `undump`. However, this is now a very old version of TEX (as of this writing TEX is up to version 3.1415), so you should seriously consider upgrading to the new version of TEX.

If you are going to create a "private" version of TEXsis rather than installing it for the whole system then you need to add your TEXsis directory to the environment variables `TEXFORMATS` and `TEXINPUTS` and define the command `texsis` as follows. For the C-shell, put in your `.cshrc` file:

```
alias texsis virtex \\\&texsis
setenv TEXSIS_LIB $HOME/texsis/
setenv TEXFORMATS ".:$TEXSIS_LIB:$TEXFORMATS"
setenv TEXINPUTS  ".:$TEXSIS_LIB:$TEXINPUTS"
```

(This assumes that `TEXFORMATS` and `TEXINPUTS` are already properly defined. If not, insert the proper directories.)

6.2. Under VMS, place in your `login.com` file (or in the system login file) the commands

```
$ DEFINE TEXSIS_LIB ⟨disk⟩:[⟨directory⟩]
$ TEXSIS :== $TEX$:TEX.EXE &TEXSIS_LIB:TEXSIS
```

where `TEX$` is the logical name for the main TEX directory. You can also put these commands in a TEX initialization `.COM` file.

Some versions of TEX on VMS have a `VIRTEX.EXE` in the TEX directory. For these versions it is necessary to create a new `TEXSIS.EXE` file from the `TEXSIS.FMT` file with the commands

```
VIRTEX "&TEXSIS"
\end
save texsis
```

`TEXSIS.FMT` can then be deleted. Your local TEX documentation should provide further information.

6.3. For PCTEX you should find in the distribution a file `TEXSIS.BAT`. If you don't, you can create it. The file should contain:

```
ECHO OFF
\PCTEX\TEX  &TEXSIS %1 /F 25000
```

(Replace the `TEX` command with `TEX386` or `TEX386b` if appropriate for your machine. For PC-TEX versions 3.14 and greater the `/F 25000` is no longer needed.) Either put this file in a directory in your command path or add the directory it is in to your `PATH` statement. If you want to be able to load a patch file then you should also include a line such as

```
\def\TXSpatches{C:/TEXSIS/TXSpatch}
```

in the `TXSsite.tex` file. Because TEX and MS-DOS both use backslashes for their own purposes the path is specified with *forward* slashes.

After these steps have been carried out, TEXsis should be installed and available to you, and to other users if installed system-wide. The command

`texsis` *name*

should run TEXsis on the file *name*`.tex` and produce the file *name*`.dvi`, which can be printed like any `.dvi` file. A good test is to copy all the files `TXS`*xxx*`.doc` from the TEXsis directory into a directory to which you have write access and then to do

```
texsis Manual
```

On the first try this will produce several warning messages coming from undefined tags for forward references. These will be defined later in the document and saved in the file `Manual.aux`; Running TEXsis a second time should produce the TEXsis manual with no errors.

B. Revisions

This appendix lists the major changes in TEXsis as new versions have come out. It is organized by sections of the manual and is intended as a guide for users familiar with the previous version. There are many smaller changes in addition to those listed here.

B.1. Revisions for Version 2.16

Created a single file `mtexsis.tex` which can be used for processing TEXsis e-prints and other documents if you have not installed TEXsis.

Section 3: Added `\NFootnotes` to produce automatically numbered footnotes.

Section 4: Added a number of useful symbols. Removed the raised `\zeta` and `\chi`, which caused trouble in some contexts.

Section 5: Improved the treatment of spacing for reference numbers and for multiple citations.

Section 7: Reorganized the documentation so that document formats, the title page, chapters and sections, running headlines, the table of contents, and the index are all described together here.

Section 7.3: Fixed spacing and hyphenation for long section titles.

Section 7.3: Changed default value of `\sectionskip` and `\subsectionskip`, so they are not quite so big.

Section 7.4: The page numbers for full-page tables and figures were wrong in the table of contents; this has been corrected.

Section 7.4: `\Contentstrue` and `\Contentsfalse` turn on and off the table of contents. The old `\ContentsSwitchtrue` is obsolete but still works.

Section 9: Figures and tables can now be inserted at the bottom of a page with `\bottominsert` or `\heavyinsert`.

Section 9: `\ListFigureCaptions` allow you to print the figure captions at the end of a document.

Section 9: In the previous version the obsolete `\Tablebody` was moved to a style file called `Tablebody.txs`. The name of this file has been changed to `Tablebod.txs` so that it can be used on a DOS PC.

Section 10: Added `\PhysRevManuscript` style for manuscripts to be submitted to the *Physical Review*, and `\PhysRev` and `\PhysRevLett` styles emulating the appearance of those journals. These are in the style files `PhysRev.txs`. Revised the `\IEEE` and `\WorldScientific` styles to follow those formats more closely. Moved twin page output to a style file.

Section 10: Changed the method of loading site and style files so that they work even with OZTEX on a Macintosh.

Section 12: Extensively revised the double column macros. They now support insertion of figures at the bottom of the page or of a column, allow figures to flow from one column to the next, and allow footnotes. The new version is backwards compatible except for the names of the headline and footline offsets, and can also be used by itself with Plain TeX.

B.2. Revisions for Version 2.15

An index has been added to the manual, using the index making macros of `index.tex` and the MakeIndex program. We have included `index.tex` in the standard distribution of TEXsis, but the only documentation at present is that contained in the source file.

Section 1: Added a separate file called `Example.tex` which contains a short example physics paper which uses the TEXsis macros.

Section 3: Support added for `\scr` using Ralph Smith's Formal Script fonts, assuming that they are available.

Section 3: Added `\Footnote` to act like `\footnote` but with the text of the footnote singlespaced and in 10 pt type. (The font can be changed by changing `\FootFont`.)

Section 5: Modified `\reference`, `\cite`, etc., to insert extra space after a reference following the end of a sentence.

Section 7.3: Added `\pagecheck` to `\section` and `\subsection`, etc. These macros skip to to a new page if less than `\sectionminspace` space is left on the page. The initial setting for `\sectionminspace` is `0.25\vsize`. Also made the skips before the titles of sections and subsections variables `\sectionskip` and `\subsectionskip` in place of the previous fixed values.

Section 10: The definition of `\thesis` has been moved to the style file `thesis.txs` so that it is easier to make local changes. The `\thesis` command automatically loads from this file.

Section 10: Added the style file `PhysRev.txs` containing a definition of `\PhysRev` for a Physical Review format. A page proof layout should be available in the future.

Section 9: Added `epsf.tex` to the standard TEXsis distribution and included documentation for `\epsfbox` and related macros.

Section 9: The macros for making lists of figures and tables have been improved (along the lines of the new table of contents macros).

Section 11: Modified `\formletters` to have a complete `\letterhead` macro or even more complicated things as one of the arguments, rather than just the phone number. This allows you to create more complicated letterheads. Note also that the order of arguments is now reversed from what it was previously! The first argument is the `\letterhead` stuff (it gets done *before* the address) and the second argument is the name of the text file (it controls what is printed *after* the address.

Section 12: Introduced `\longequation` and `\endlongequation` for equations which span both columns. The definitions of these can vary from style to style, so they may or may not include a `\leftcolrule` and `\rightcolrule`.

Section 13: Added this section to the manual to explain miscellaneous useful macros which don't fit anywhere else.

Section 13: Added `\Ignore` and `\endIgnore` to ignore large blocks of text. This is slightly more clever than `\comment`, which remains available.

B.3. Revisions for Version 2.14

Section 3: `\TrueDoubleSpacing` generates a document with 2 times the normal spacing, in contrast to `\doublespaced`, which gives 1.5 times the normal spacing.

Section 4: Fixed a bug in `\bordermatrix`.

Section 5: An error which occurred if `\endreference` was followed by additional text on the same line has been fixed.

Section 5: Extra space is now inserted after a reference number which follows a period.

Section 5: `\refrange` can now be used with `\reference` or `\cite`; the correct numbers and ranges separated by commas are printed.

Section 8: The internal spacing mechanism for environments has been modified to work with arbitrary `\baselineskip` and `\parskip` values. Some minor changes have been made to the center and flush environments.

Section 8: The `\Listing` environment has been modified so that the characters `{} % & () $ #` and `|` are just ordinary printable characters and so that `\\` gives a printed `\`.

Section 8: `\@startenv` and `\@endenv` to begin and end environments have been replaced with `\begEnv` and `\endEnv`, which are accessible to the user because the do not have `@` in their names.

Section 9.4: The `\ruledtable` macros can now use `\tabskip` glue to produce a table of arbitrary width. Spacing of the tables has been improved and made more robust.

Section 7.3: `\addsectioncont` has been replaced with `\addTOC`, i.e., "add to Table Of Contents." The new macro is simpler and produces better-looking output.

Section 7.3: Spacing such as `\singlespaced` placed after the `\texsis` initialization now sets the spacing for the entire document; the explicit spacing commands in `\abstract`, etc., have been removed.

Section 7.3: A macro `\setSectionID` has been added to allow one to place a figure or table for one section in a different section if necessary and still get the correct number.

Section 10: The `\NorthHolland` format now gives the more common single-column North-Holland/Elsevier format. `\NorthHollandTwo` produces the previous double-column oversized format. `\Manuscript` produces a preprint with true double spacing as requested by The Physical Review.

Section 10: `\endtitle`, `\endauthor`, etc., can now be omitted from the titlepage material, since `\title`, `\author`, etc., now close the previous mode automatically.

Section 11: Letters and memos can now be ended with `\bye`.

B.4. Revisions for Version 2.13

Section 3: Italic fonts are now available in large sizes, e.g. for `\slides`. Math italic bold fonts have been added for titles.

Section 4: By default parentheses and brackets are now ordinary characters except in math mode and so can be used, e.g., in file names. As a result, automatic sizing of parentheses works only in *displayed* equations. You can restore the old behavior by saying `\onparens` at the beginning of the document.

Section 5: The text for references is now written with `\unexpandedwrite`. In practice this means that references can contain almost anything, including a `\Ref`.

Section 5: The `\journal` macro was added to make it easier to create standard citations to journal articles, and to be able to switch between the American and European styles.

Section 5: Superscript references are now in `\citestyle`, which is initially set to `\scriptstyle` but can be changed.

Section 9.3: A way was created to save all figure and table insertions until the end of the document with `\FiguresLast`, and then print them with `\PrintFigures`, and similarly for tables.

Section 10: The `\NorthHolland` document layout was added to produce camera-ready copy for Proceedings published by North-Holland/Elsevier. This is a double column format with titles left justified and eleven point type.

Saying `\singlespaced` immediately after `\texsis` or its equivalent will now make the whole document single spaced.

Section 12: The double column macros have been modified so that removing the `\setdoublecolumns` will produce a single column document.

B.5. Revisions for Version 2.12

Section 2: A new command `\texsis` performs the initialization, and calling it more than once no longer causes an error. `\inittexsis` remains as a synonym.

Section 3: `\ninepoint`, `\twentypoint`, and `\twentyfourpoint` commands have been added to select these sizes. The 9 pt. size is really too small for a standard laser printer, but it is useful if you must print IEEE format on 8.5 in $\times$ 11 in paper. The large sizes are useful for slides and other special purposes.

Section 9.3: The table and figure insertion macros has been extensively modified, and the syntax has been changed. `\topTable`, etc., have been replaced by `\table`, etc., which make the insertion and define the number for the tag. `\Tbl` and `\Fig` use the number. References to tables defined later in the paper are satisfied using the `.aux` file, so `\TOPTABLE`, etc., are no longer needed. The old names remain as synonyms but their use is discouraged.

Section 9.3: `\ruledtable` and many associated macros have been added to make complex ruled tables, including those with entries spanning more than one column. `\Tablebody` is still available, but the new macros are much more flexible and should be used instead.

Section 9.3: New macros have been added to list figures and tables.

Section 10: New formats have been added. `\WorldScientific` and `\IEEEreduced` make conference proceedings in World Scientific format and in IEEE format at 75% of full size. `\referee` is based on the memo format and can be used for referee reports. `\slides` uses large type to make transparencies for overhead projectors and similar material.

Section 10: An `\authors` macro for papers with multiple authors and facilities for putting footnotes on titles have been added.

Section 10: `\bookpagenumbers` causes odd page numbers to be placed in the upper right corner and even ones in the upper left corner. Additional information on headlines is given.

Section 11: The form letter macros have been rewritten. The syntax is the same except that `\v` has been replaced by `\V` to avoid conflict with the Plain TEX accent. Also, `\V` now also works with `\formlabels`.

Section 11: The new `\referee` format is described.

Section 12: The double column format has been extended to handle floating insertions. `\topinsert` and `\pageinsert` produce insertions within a column, while `\widetopinsert` and `\widepageinsert` produce insertions across both columns.

Section 6: The tagging mechanism has been changed to add an `@` before and after the name. The practical effect is that using the name of a defined control sequence like `book` for a tag no longer produces an error.

C. Font Tables

The Computer Modern fonts that are used either by Plain TEX or by TEXsis and the required sizes for each are listed in Table C.1. The `cmmib10` and `cmbsy10` fonts may not exist in all installations, but they will not be loaded unless `\mib` is used, so they are not required. The command `\printfont{`⟨font⟩`}` can be used to print a table of the characters in any ⟨font⟩, e.g.,

```
\printfont{cmr10}
```

A separate file called `Fonts.tex` is included with the standard distribution of TEXsis. Running `Fonts.tex` through TEXsis will produce tables of all of the fonts used by your installation of TEXsis.

Most versions of TEX now use "CM" (Computer Modern) fonts, but some still use the older "AM" (Almost Modern) fonts. If you have problems with the fonts, try editing `TXSfonts.tex`, replacing all occurrences of `=cm` with `=am` and then recompiling TEXsis. Mixing the two types of fonts will produce errors.

Table C.1: Fonts used in TEXsis, with the design size names and the required sizes in points.

Base Name	Description	Sizes (pt)									
cmr10	Roman	5	7	9	10	11	12	14	16	20	24
cmti10	Text italic	5	7	9	10	11	12	14	16	20	24
cmsl10	Slanted	5	7	9	10	11	12	14	16	20	24
cmbx10	Roman bold	5	7	9	10	11	12	14	16	20	24
cmmi10	Math italic	5	7	9	10	11	12	14	16	20	24
cmsy10	Math symbol	5	7	9	10	11	12	14	16	20	24
cmex10	Math extended				10	11	12	14	16	20	24
cmtt10	Typewriter				10	11	12				
cmss10	Sans serif				10	11	12	14			
cmmib10	Math italic bold				10	11	12	14	16	20	24
cmbsy10	Bold symbol				10	11	12	14	16	20	24

Acknowledgments

We wish to thank Isabel Harrity for her patience in trying this code while it was being developed. Various other people have also given us suggestions, criticisms, and comments regarding the code and documentation for TEXsis; among them we would like to thank:

Vladimir Alexiev	`<vladimir@cs.ualberta.ca>`
Betty Armstrong	`<bettya@csa.lbl.gov>`
Leo Eskin	`<lion@navier.stanford.edu>`
Don Groom	`<deg@Csa6.LBL.Gov>`
Scott Hannahs	`<hannahs@buphy.bu.edu>`
Diab Jerius	`<dj@pelf.harvard.edu>`
Arik Kapulkin	`<arik@lifshitz.ph.utexas.edu>`
Bob Love	`<rlove@raptor.rmnug.org>`
Michael Morrison	`<mam@jilau1.Colorado.edu>`
Jeroen Rommers	`<rommers@zeus.rijnh.nl>`
Anil Trivedi	`<trivedi@yukawa.uchicago.edu>`
Samir Varma	`<varmint@ccwf.cc.utexas.edu>`
T.C. Pierce	`<es051@eng.warwick.ac.uk>`

Additional suggestions are welcome and will be incorporated in future versions.

This work was supported in part by the United States Department of Energy under contracts DE-AC02-76CH00016 and DE-AC02-89ER40509, and in part by the National Science Foundation under grant number PHY8919177.

References

1. D.E. Knuth, *The TEXbook* (Addison Wesley, 1986).
2. W. Groppe, *TechRpt* (unpublished).
3. E. Myers, *TEXsis Version 1* (unpublished).
4. M. Weinstein, SLAC-TN-84-007 (1984).
5. L. Lamport, LATEX, *A Document Preparation System*, ISBN 0-201-15790-X.
6. M. Doob, *A Gentle Introduction to TEX (A Manual for Self-Study)*, (TEX User's Group, Providence). Also available as the Plain TEX file `gentle.tex` via anonymous ftp from `lifshitz.ph.utexas.edu` and from various other sources on the net.
7. M. Spivak, *PCTEX User's Guide.* (Documentation distributed with PCTEX).
8. R.P. Feynman, Phys. Rev. 76, 749 (1949).
9. W. Pauli and F. Villars, Rev. Mod. Phys. 21, 434 (1949);
   J. Schwinger, Phys. Rev. 74, 1439 (1948).
10. J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Phys. Rev. 108, 1175 (1957).

11. E.D. Bloom, et al., Phys. Rev. Letters 23, 930 (1969).

12. Pehong Chen and Michale A. Harrison, "Index Preparation and Processing", in *Software: Practice and Experience*, 19, 897-915 (1988).

13. *The Adventures of Buckaroo Banzai across the Eighth Dimension*, (Twentieth Century Fox, 1984).

14. S. Weinberg, *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity* (Wiley, New York, 1972) pg. 3.

15. Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, (Prentice Hall, Englewood Clifs, NJ, 1978).

16. Tomas Rokicki, `dvips` User's Manual (unpublished). The C source for the program and the TEX source for the manual are available from several of the TEX anonymous ftp sites.

17. D.E. Knuth, *The TEXbook* (Addison Wesley, 1986), page 245.

18. Ray F. Cowan, *Making Tables with Macros* (unpublished).

19. J.D. Bozek, Ph.D. thesis (unpublished), Department of Chemistry, University of Western Ontario, London, Ontario CANADA.

20. *Handbook of Chemistry and Physics*, 56th Edition, (CRC Press, Cleveland, Ohio, 1974), pg. B-9.

Index

`&`, 57, 58
`*`'d citations, 22
`~`, 58

`\abstract`, **31**, 99
`\address`, **75**, 76
`\addressee`, **76**
`\addTOC`, **36**, 99
`\AIPproceedings`, **69**
alignment character, 57
alignment tabs, 56
`\and`, **30**
`\annotations`, **75**, **80**
`\Appendix`, **33, 34**
`\appendix`, **33, 34**
`\ATlock`, 73
atomic reactors, 84
`\ATunlock`, 73
`\author`, **30**, 81, 99
`\authors`, **31, 32**, 81, 101
`\autoload`, **72**
`\autoparens`, **16, 17**

`\banner`, 28, 29, 31, 32
baseline skip, 9
`\baselineskip`, 99
`\begEnv`, 99
`\beginlines`, 49
`\begintt`, 49
`\bf`, 7
`\bgroup`, 47
`\bigitem`, 62
`\bigl`, 18
`\Bigr`, 18
bismuth, 84
`\BL`, **78**
`\bl`, **73**
`\blockquote`, **45**
`\body`, **75**, 77, 80, 81
`\book`, **30**, 36–38, 70
`\bookpagenumbers`, **37**, 101
`\booktitle`, **21**
`\bordermatrix`, 99
`\bottomfigure`, **52**, 68
`\bottominsert`, 50, 51, 85, 97
`\bottomtable`, **56**
`\bra`, **17**
`\buildchar`, **18**
bullets, 41
`\bye`, 27, 76, 80, 81, 99

cadmium, 84
`\cal`, 8
calligraphic script, 8
camera-ready, 83
`\Caption`, **51**, 56, 66, 67
`\caption`, **51**, 56, 65–67
`\cases`, 18
`\cc`, **75, 76**, **80**
`\center`, **40**
`\centeredtables`, 61
`\centerline`, 56
`\chapter`, 15, 24, **33–35**, 36–38
`\ChapterStyle`, **35**
checkpoint, 89
`\checkpoint`, **89**
citations, *see* references
`\cite`, **23**, 25, 98, 99
`\citerange`, **24**
`\citestyle`, 100
`\closing`, **75–77**
`\cmsec`, **19**
column break, 86
`\colwidth`, 52
`\comment`, **88**, 89
compatibility
    with plain TEX, 3
`\Contents`, **36, 37**
`\Contentsfalse`, **36**, 97
`\Contentstrue`, **36**, 97
corollaries, 46
`\Corollary`, **46**
`\corollary`, **46**
Cowan, Ray F., 57
`\CR`, 58
`\cr`, 15, 57, 58, 65
`\crnorule`, 57, 58
`\crpart`, 59

\crrule, 58
\crthick, 58
\crule, 59
\cskip, 59
\csname, 25
custom modifications, 92
customization
    chapters and sections, 35

\Date, **79**
\dated, **75**
\dbl, 58, 65
\def, 85
\Definition, **46**
\definition, **46**
\degree, **18**
\degrees, **18**
\del, **18**
\description, **41**, **44**
\dimensions, **19**
\disclaimer, **31**
\displaylines, 15
distance between lines, 9
\Distribution, **80**
Doob, Michael, 23
double columns, 83
    figures and tables, 86
    insertions, 85
    long equations, 84
\doublecolumns, 70, **83**, 84, 85, **86**
\doublespaced, **9**, 62, 99
\draft, 12, 27, **88**
\dump, 94
DVI file, 53
dvips, 53, 54, 74

\egroup, 47
\EL, **78**
electronic bulletin boards, 32
\elevenpoint, **7**
Encapsulated PostScript, *see* PostScript
\Encl, **76**, **80**
\end, 25
\endauthor, 99
\endauthors, 31, 33
\endblockquote, **45**
\endCaption, **51**, **67**
\endcsname, 25
\enddescription, 41
\enddoublecolumns, 70, **83**, 84, 85, **86**
\endenumerate, 41
\endEnv, 99
\endexample, 45
\endfigure, **52**
\endIgnore, 88
\endinsert, 50, 85
\enditemize, 41, 42
\endletter, **76**, 77
\endListing, 47
\endlongequation, **84, 85**, **87**, 98
\endmemo, **80**
\endproof, 46
\endreference, 21–23, 99
\endreferencelist, 23
\endruledtable, 57
\endstat, **89**
\endtable, **57**
\endtheorem, 46
\endtitle, 30, 99
\endtitlepage, **30**, 71
\enumerate, **41**, 42
\enummark, **43**
\enumNumOutline, **43, 44**
\enumOutline, **43**
\enumPoint, **44**
\enumPoints, **43, 44**
\Ep, **11**
EPS file, 54
\epsfbox, 50, **53–55**, 98
\epsfxsize, **54, 55**
\epsfysize, **54, 55**
\Eq, **11**, **13**, 25
\eqalign, 13, 14, 17
\eqalignno, 13, 91
\EQN, **11**, **13**, 15
\EQNalign, **13**, 14, 15, 17, 91
\eqnbox, **91**
\EQNdisplaylines, **15**
\EQNdoublealign, **15**
\eqnmark, **91**
\eqnmarker, **91**
\eqno, 11
\eqnotracefalse, 12

\eqnotracetrue, **12**, 88
\Eqs, **11**, **13**
equations
  alignment, 13, 15, 19
  labeling, 11, 12
  multiple, 12, 13
  numbering, 11, 16
\Eurostyletrue, **22**
\eV, **19**
\everychapter, **35**
\everyListing, **47**
\everysection, **35**
\everysubsection, **35**
\everysubsubsection, **35**
\example, **45**, 46, 48

Feynman, Richard P., 21
\Fig, 26, **51**
\fignum, 68
\figure, **51, 52**, 65, 67
figures, 50
  at end of document, 67
  nonstandard placement, 67
  placement, 51, 52
  size, 52, 53
\FiguresLast, 67, 68, 100
\FiguresNow, 67
fire detection, 84
floating insertion, 50
\flushleft, **40**
\flushright, **40**
\folio, 25, 36
fonts, 7, 94
\FootFont, **9, 10**, 98
\Footnote, 3, **9, 10**, 98
\footnote, 3, 9, 10, 32, 98
footnotes, 9
\footnum, **10**
\forceleft, **85**
\forceright, **85**
form letters, 77
formal script, 8
\formlabels, 79, 101
\formletters, 75, **77**, 78, 98
forward references, 27
\fourteenpoint, **7**
\frac, **18**
frisbee, 39
\From, **79**, 81
\fullfigure, **52**
\fulltable, **56, 57**
functional measure, 8

\gdef, 76, 78, 85
Geoffroy, Claude, 84
\GeV, **19**
\grad, **18**
\gtsim, **18**

\half, **18**
\halign, 15, 56–59, 65
Hall effect, 84
Harrity, Isabel, 103
\hbox, 61, 88
\HeadFont, **38**
\HeadText, 38
\heavyfigure, **52**
\heavyinsert, 50, 85, 97
\heavytable, **57**
"hello, world", 47
hep-lat, 32
\hfil, 63, 65
\hfill, 63, 65
\hrule, 51
\hsize, 52, 74, 77, 83, 84, 86

\idx, **38, 39**
\IEEE, 84
\IEEEproceedings, **69**
\IEEEreduced, **69**, 101
\Ignore, **88**, 89, 99
\IL, **78**
\imag, **19**
\index, **38**
\infiglist, **65, 66**
initex, 94
\input, 23, 72
insertion classes, 51
\institution, **31, 32**, 33
\intablelist, **65, 66**
\it, 7
\itemize, **41, 42**
itemizing, 41
\itemmark, **42**

\itm, 41, 44, 45
\jot, 91
\journal, **22**, 100
\jtem, **90**
\ket, **17**
\KeV, **19**
\killspace, 63
Knuth, Donald, 1
label, 21
\label, **26**, 35
Lagrangian, 8
\LandscapeSpecial, 74, **92**, 94
LaTeX, 1, 3
\lbrack, 17
\left, 16, 17
\Leftcases, **18**
\leftcolrule, **85**, **87**, 98
\LeftJustifyTables, 65
\leftpar, **90**
\Lemma, **46**
\lemma, **46**
lemmas, 46
\letter, **75, 76**, 77
\letterhead, **75**, **77**, 78, 92, 98
letters, 75, *see* form letters
  adjusting size, 76
\letterwidth, **77**
lines
  centering, 40
  flush left, 40
  flush right, 40
\ListCodeFile, **47**
\ListFigureCaptions, **66, 67**, 97
\ListFigures, **66, 67**
\Listing, **47**, 99
listing computer code, 47
\ListReferences, 21, 23, **24**
lists
  different kinds, 41
  enumerated, 41, 42
    styles, 43, 44
  indentation, 44
  itemized, 41
  of figures, 66
  of references, 24
  of tables, 66
  text labels, 41, 44
\ListTableCaptions, **66**
\ListTables, **66**
\longequation, **84, 85**, **87**, 98
\longindent, **76, 77**
\loosebox, **91**
\LooseTables, 62
\lparen, 17, 18
\ltsim, **18**
macros
  arguments, 3
  names, 3
  syntax, 3
mail merge, *see* form letters
mail servers, 32
makeindex, 38
\Manuscript, **29**, 71, 99
\markindextrue, **39**
master file, 5
\mb, **19**
\memo, 75, **79**, 80, 81
memos, 79
  customization, 79
MetaFont, 8
\MeV, **19**
\mib, **7, 8**, 102
\midfigure, **51, 52**
\midinsert, 50, 51, 85
\midtable, **56, 57**
MS-DOS, 93
\MSref, **81**
\mub, **19**
multiple columns, *see* double columns
\multispan, 58
\n, 35, 36, 47, **79**
\nabla, 18
\nb, **19**
\newcolumn, **86**
\newpage, 86
\nextcolumn, 65
\NFootnote, **10**, 35
\NFootnotes, 97
\ninepoint, **7**, 100
\noalign, 15

\noexpand, 37
\NoJustifyTables, 65
\noncenteredtables, 60
\nopagenumbers, **37**, 76, **80**
\normhsize, **74**
\normvsize, **74**
\NorthHolland, **69**, 99, 100
\NorthHollandTwo, **69**, 70, 84, 88, 99
\nosechead, 24, **33, 34**, 35, 36
\NoTrailingSpaces, 63, 65
\np, **73**
\nr, 58
\nuclproc, **69**, 88
\nunubar, **18**

\obeylines, 30, 40, 75, 76, 80, 90
\obeyspaces, 90
\offparens, 16–18
\omit, 58
\onparens, 17
\ORGANIZATION, 79, 92, 94
\output, 83
\overcirc, **18**
\overfullrule, 88

\pacs, **31**
\pagecheck, **90**, 98
\pageinsert, 50, 51, 85, 101
\pageno, 36
\pagenumbers, **37**
\paper, 28, **29**, 31, 72
\para, 62
\parasize, 62
\parenbar, **18**
parentheses
  sizes, 16
\parskip, 10, 44, 45, 99
\partial, 18
\pb, **19**
PCTEX, 96
penguin, 50, 51, 55
, 58
philosophy, 3
Physical Review, 83
\PhysRev, **70, 71**, 84, 97, 98
\PhysRevLett, **71**, 97
\PhysRevManuscript, **70, 71**, 97
PHYZZX, 1
plain, 1
plain TEX, 1
\plaincr, 65
\plaintab, 65
PostScript, 53, 54
\ppbar, **18**
\preprint, 28, **29**, **31**, 71, 74, 92
\PrintFigures, 67, 100
\printfont, **102**
\PrintTables, 67
\proof, **46**
\ps, **76**, **80**
\pubcode, **32**
\pubdate, **32**

\quad, 62, 63
quote balancing, 10
\quoteoff, **10**
\quoteon, **10**

\raggedcenter, **40**, 41
\rbrack, 17
\real, **19**
recursion, *see* recursion
\Ref, **24**, 100
\referee, 75, **81**, 101
referee reports, 81
\reference, **21, 22**, 23, 98, 99
\referencelist, **23**
references, 21
  citation marks, 23
  entering a list, 23
  printing the list, 24
\refFormat, **24**
\refindent, **24**
\refrange, 99
\refskip, **24**
\relax, 24, 84, 85
restart, 89
\restart, **89**
restore, *see* restart
\right, 16, 17
\Rightcases, **18**
\rightcolrule, **85**, **87**, 98
\RightJustifyTables, 65
\rm, 7, 9

(Boldface page numbers are the defining entries)

Rokicki, Tomas, 53, 74
\rparen, 17, 18
rsfs font, 8
ruled tables, 57
  centered, 60
  customization, 65
  footnotes, 61
  size, 61, 62
  spacing, 62
  text, 62
\ruledtable, 57, 59, 61, 62, 65, 99, 101
running TEXsis, 5
\RunningHeadstrue, **38**

sans-serif fonts, 8
\scr, **7, 8**, 98
\scriptstyle, 100
\section, 3, 15, 24, **33–35**, 36–38, 98
\sectionminspace, **34**, 98
\sectionskip, **34**, 97, 98
\SectionStyle, **35**
\setcnt, 68
\SetDoubleColumns, 73, **83–86**
\setdoublecolumns, 100
\setHeadline, **38**
\setSectionID, 67, 99
\showchaptIDfalse, **34**
\showchaptIDtrue, 16, **34**, 37, 46, 52
\showsectIDfalse, **34**, 37
\showsectIDtrue, 16, **34**, 37, 46, 52
\simge, **18**
\simle, **18**
\singlelinefalse, **44**
\singlelinetrue, **44**
\singlespaced, **9**, 62, 99, 100
\sixteenpoint, **7**
\sl, 7
\slashchar, **19**
\slides, **73**, 100, 101
small title bold face, 8
\smallfrac, **18**
\smallskip, 24
Smith, Ralph, 8
\space, 63
\span, 58
\special, 50, 54, 74
\square, **18**, 42
\sterling, **19**
\Subj, **79**
\subjectline, **80**
\submitted, **31**
\subrightarrow, **18**
\subsection, **33, 34**, 98
\subsectionskip, **34**, 97, 98
\SubsectionStyle, **35**
\subsubsection, **33, 34**
\supereject, 66, 67
\superrefsfalse, **23**
\surd, 42
symbols
  physics, 17

\tab, 58
\table, **56, 57**, 59, 65, 66
table of contents, 36
\Tablebody, 97
\tableinfofalse, 62
\tableinfotrue, 62
\TableItem, 58, 63, 65
\TablePreable, 65
\TablePreamble, 65
tables
  at end of document, 67
  nonstandard placement, 67
  ruled, *see* ruled tables
  simple, 56
\TablesLast, 67
\TablesNow, 67
\tablespace, 62, 63
\tablewidth, 61, 62
\tabskip, 99
\tag, **25**, 26
\Tbf, **8**, 30, 35
\tbf, **8**, 35
\Tbl, 57
TechRpt, 1
\tenpoint, **7**, 9
\tenss, 8
\TeV, **19**
TEX, typing examples, 48
TeX capacity exceeded, sorry, 67

The TEXbook, 1, 3, 10, 11, 49, 50, 56–59, 61, 75, 111
\TeXexample, **48**
\TeXquoteoff, **48**
\TeXquoteon, **48**
\TeXsis, 5
TEXsis, 1
\texsis, 5, 7, 16, 29, 83, 88, 99, 100
\Text, **80**
\text, 75, 81
\theBlank, **90**
\Theorem, **46**
\theorem, **46**
theorems, 46
thermocouple, 84
\thesis, 36–38, 69, **70, 71**, 72, 98
thesis.txs, 71
\thicksize, 58
\thinsize, 58
\tightbox, **91**
\TightTables, 62
\title, **30**, 37, 81, 99
Title bold face, 8
\titlepage, **30**, 32
\To, **79**
\toappear, **31**
\TOCwrite, **37**
\today, 75, 79
\topinsert, 50, 51, 85, 86, 101
\topspace, **90**
\Tr, **19**
\tr, **19**
\TrailingSpaces, 65
\triplespaced, **9**, 62
\TrueDoubleSpacing, **9**, 99
\tstrut, 62
\tt, 7, 48
\twelvepoint, **7**, 9
\twelvess, 8
\twentyfourpoint, **7**, 100
\twentypoint, **7**, 100
twin page output, 73
\twincsize, **74**
\twinfmt, **73, 74**
\twinhsize, **74**
\twinout, **73, 74**
\twinprint, **74**
\twinvsize, **74**
type sizes, 7
typestyles, 7

\undertext, **90**
undump, 95
\unexpandedwrite, 100
\units, **20**
\unobeylines, **90**
\unobeyspaces, **90**
\use, 24, **25, 26**, 37, 46, 68

\V, 78, 101
Vacuum Expectation Value, 17
\Vb, 58
\vb, 58
\vbig, 17
\vbigl, **18**
\vbigr, **18**
\vbox, 56
\vev, **17**
\vfill, 52, 56
\Vfootnote, **10**
\vfootnote, 10, 32
virtex, 95
\vol, **21**
\vsize, 74, 83, 86
\vskip, 15, 52, 90

\widebottominsert, **86**
\widefigure, 52, **86**
\widefullfigure, 52, **86**
\widefulltable, 57, **86**
\widenspacing, **9**, 62
\widepageinsert, **86**, 101
\widetable, **86**
\widetopfigure, 52, **86**
\widetopinsert, **86**, 101
\widetoptable, 57
\WorldScientific, **70**, 101

This page was intentionally left blank.

Contents

1. Overview 1
   1.1. Some Philosophy 3
2. Getting Started 5
   2.1. The Master File 5
3. Type Sizes, Type Styles, and Line Spacing 7
   3.1. Interline Spacing 9
   3.2. Footnotes 9
   3.3. Automatic Quote Balancing 10
4. Equations 11
   4.1. Equation Numbering 11
   4.2. Automatic Parentheses Sizing 16
   4.3. Special Symbols 17
5. References and Citations 21
6. Labels and Tags 25
   6.1. Forward References 27
7. Typesetting Complete Papers 28
   7.1. Document Types 29
   7.2. The Title Page 30
   7.3. Chapters and Sections 33
   7.4. Table of Contents 36
   7.5. Running Headlines and Footlines 37
   7.6. How to Make an Index 38
8. Environments 40
   8.1. Flush or Centered Text 40
   8.2. List Environments 41
      8.2.1. Itemized lists 41
      8.2.2. Enumerated lists 42
      8.2.3. Lists Labeled by Text 44
      8.2.4. Spacing of List Environments 44
   8.3. Simple Examples and Quoted Text 45
   8.4. Theorems and Proofs 46
   8.5. Listing Computer Code 47
   8.6. TEX example macros 48

9. Figures and Tables 50
9.1. Figures 50
9.2. Encapsulated PostScript Graphics 53
9.3. Simple Tables 56
9.4. Ruled Tables 57
9.5. Lists of Figures and Tables 65
9.6. Listing Figure and Table Captions 66
9.7. Printing Figures and Tables at the End of a Document 67
9.8. Nonstandard Figure and Table Placement 67
10. Style Files and Specialized Formats 69
10.1. Style Files 69
10.1.1. Physical Review Formats 71
10.1.2. Thesis or Dissertation Format 71
10.2. Creating New Styles 72
10.3. Slides 73
10.4. Twin Page Output 73
11. Letters, Memos and Referee Reports 75
11.1. Letters 75
11.2. Form Letters 77
11.3. Memos 79
11.4. Referee Reports 81
12. Double Column Output 83
13. Miscellaneous Macros 88
13.1. `\draft` mode 88
13.2. Ignoring blocks of text 88
13.3. Checkpoint/Restart 89
13.4. Other Goodies... 90

Appendices:

A. Customization and Installation of TEXsis 92
A.1. Customization 92
A.1.1. Local Modifications 92
A.1.2. Site Modifications 92
A.1.3. Patches 93
A.2. Installation of TEXsis 93

A.2.1. Quick Install for Unix and TEX 3.0 93
A.2.2. Detailed Installation 94
B. Revisions 97
B.1. Revisions for Version 2.16 97
B.2. Revisions for Version 2.15 98
B.3. Revisions for Version 2.14 99
B.4. Revisions for Version 2.13 100
B.5. Revisions for Version 2.12 100
C. Font Tables 102
Acknowledgments 103
References 103
Index 105

List of Tables

Table 9.1. Charged leptons. 56

Table 9.2. Example of a ruled table — AT&T Common Stock. 59

Table 9.3. Example of a ruled table — Parameters used in the MS-Xα calculations for the chlorosilane molecules, H_xSiCl_{4-x}; x=0–4. 59

Table 9.4a. Basic macros for `\ruledtable` 62

Table 9.4b. Additional macros for `\ruledtable` 62

List of Figures

Figure 9.1. Sample figure showing how easy it is to insert numbered figures with captions in TEXsis. 51

Figure 1.17. A bizarrely placed figure for Section 1 68