One also gets something else for the modest investment: access to the end products from a community of some of the most avid computer enthusiast/programmers in the world. There are literally hundreds of public domain software tools for the Amiga. The open operating system allows these utilities to work together so that a user can tailor a truly flexible personal application environment.

This flexibility can yield a productivity increase a few orders of magnitude greater than any benchmark of straight cpu performance could possibly indicate. Perhaps a better "benchmark" of both the developer community's sophistication and the Amiga's capabilities is the fact that already next to my Amiga are five software product manuals typeset with AmigaTEX.

These capabilities are not without price: users have to learn AmigaDOS on their own as documentation of the Amiga as a workstation is all but nonexistent. Most dealers are completely ignorant of its potential, and it can take many (enjoyable) hours of testing new utilities to optimize an Amiga for specific work habits. Without a hardware deinterlacer (available only for the model B2000) there is an intolerable screen flicker in the hi-res mode that requires a very stable balanced room light and careful choice of colors to reduce.

Like TEX, AmigaDOS can initially seem frustrating because it offers so many choices: modular extensible software exacts a different kind of toll in the freedom it affords. When the Rexx language became available for the Amiga it was as if the last piece of a puzzle fell into place. A common vocabulary can now be used both to write macros within programs as well as to mediate interaction between programs.

As every capability of the system is available at all times the overlapping window paradigm becomes a true asset rather than mere dressing. The user who understands why TEX is not Desktop Publishing, and understands that ease of use is not enviable when the tradeoff is quality, might check out the Amiga. Like TEX, it can reward effort.

◇ ◇ ◇

ENVY is a transgression in that it focuses attention to surface appearance, diverting one from the true substance, *the kernel*, of a thing. *The heretic might suggest that all one needs for an* optimum TEX environment is a fast 32-bit cpu and a large screen. But when the system beneath such glitz is basically an 8-bit style program loader the suggestion is ludicrous. Absolution, at least in this one case, came easily: before I got an Amiga, I *really* envied people who had those big Sun workstations to run TEX.

I don't any more.

## Using TEX and LATEX with WordPerfect 5.0

Michael F. Modest
Pennsylvania State University

### 0.1 Introduction

While TEX is an extremely powerful typesetting language that can make your publications look as close to professional as you could possibly hope for, this beauty comes at a price: TEX by itself is somewhat tedious to use, and it is not WYSIWYG (what you see is what you get). There appears to be no truly WYSIWYG word processor available for the IBM PC or compatibles. The good news is that — when combined with WordPerfect, which allows customization of printer drivers — the tediousness of TEX can be overcome. I have written a TEX-LATEX driver for WordPerfect which allows you to use WordPerfect normally while typing ordinary text, and uses simple macros when typing equations. I have written the preprocessor in such a way that it may be used for LATEX documents or for straight TEX documents as long as the keys/macros specifically geared towards LATEX are not employed. The user prepares his or her WordPerfect file and sees centering, indented paragraphs, Greek letters, mathematical symbols, subscripts, etc., actually displayed (and entered with single keystrokes). Then, rather than sending this file directly to a printer, "prints" it to a diskfile using the preprocessor which converts the WordPerfect format into a standard TEX or LATEX file. If only the macros activated by the WordPerfect keyboards are used, the preprocessor even assures matching of braces, dollar signs and `\begin` and `\end` of LATEX environments.

### 0.2 WordPerfect Fonts

I have programmed the driver so that — using the ordinary WordPerfect font change commands — a number of different types of fonts are available, as well as a predefined math displaymode, as indicated below. Depending on the graphics card and monitor present in your PC, different fonts and/or attributes may or may not be displayed:

- EGA card and monitor: choosing the 1 font/ 512 character option the full Greek alphabet plus most other important symbols will be displayed. Bold, underline, italics, etc. are color-coded.

- Hercules graphics card Plus, or Incolor card: choosing the 6 font/512 character option together with a *Ramfont* set prepared by myself the full Greek alphabet, all other important symbols, bold, underline, italic, courier (i.e., verbatim), sub- and superscripts, small caps, calligraphic letters, "Large", etc. will be displayed.
- Neither: sorry, you are out of luck: only the standard 256 ASCII characters, bold and underline will be displayed.

## 0.3 Compatibility Considerations

You can use most of the features of the WordPerfect wordprocessor that you may usually use, such as super$^a$ or sub$_b$-scripts, the speller, the thesaurus, the search and replace routines, block mark/copy/move/retrieve routines, the "merge" and "macro" features, as well as **bolding** and <u>underlining</u>. A number of special features of WordPerfect, such as *tabs*, *indent*, *center*, *flush right*, *footnote*, and *linespacing* are converted into blank spaces by WordPerfect before sending it to the printer and, thus, cause a problem. Most of these features have been made compatible as indicated in the section on *WordPerfect Keyboards*, a few (e.g., linespacing) must be treated by using TEX macros directly for these functions.

If a little care is taken a text file can be printed like a normal WordPerfect file or as a TEX file, as long as the file does not contain any fancy typesetting, such as equations using TEX-commands or any "backslash"-commands. Even then the file can be printed out under WordPerfect, but it will now not enforce these commands. Since a number of "special TEX characters" are frequently used in normal text I have replaced three of them in my version of WordPerfect: $, # and % are no longer reserved, i.e., they print out as seen here. No replacement has been programmed for # and %, since the user of this WordPerfect/LATEX customization has no longer any need for them, while the reserved character $ has been replaced by the △ symbol[1]. An exception is the *Verbatim* font: in this font there are **no** special characters, i.e., \ will print out as \, etc. (usually used in conjunction with `\tt`, i.e., in typewriter mode).

## 0.4 WordPerfect Keyboards

WordPerfect 5.0 allows you to define custom keyboards and to access zillions of special symbols which cannot normally be displayed on the screen such as $\alpha\beta\gamma$, or $\rightarrow$ or $\partial$ or $\infty$ etc. This is done by depressing the corresponding key in the appropriate keyboard, sometimes accompanied by simultaneously pressing either the *Alt*-key (abbreviated here as @) or the *Ctrl*-key (here ^). Thus most keys may be used in four ways, e.g., `a`, `shft-a` (A), `@-a` and `^-a`, although some keys are limited to three by not allowing either the *Alt* or *Ctrl* key combination. I have defined six new keyboards to be used with TEX: `NORMAL`, `FOREIGN`, `GREEK`, `SINGLE`, `SMCAPS` and `CALLIGRAPHIC`.

### `NORMAL` keyboard

As the name implies this keyboard is the one used most of the time. Its basic layout is shown in Fig. 1, where the central character in each box is the name of the key, the bottom left character shows the character displayed if the key itself is pressed, top left shows the *shft* character, bottom right the *Alt* character and the top right shows the *Ctrl* character (for the function keys the WordPerfect bottom-to-top sequence is used instead). The function of a few of the individual keys is explained below in some detail (note that in most of the examples below unnecessary blanks have been included to distinguish a key like `@-a` from the three-key sequence `@ - a`).

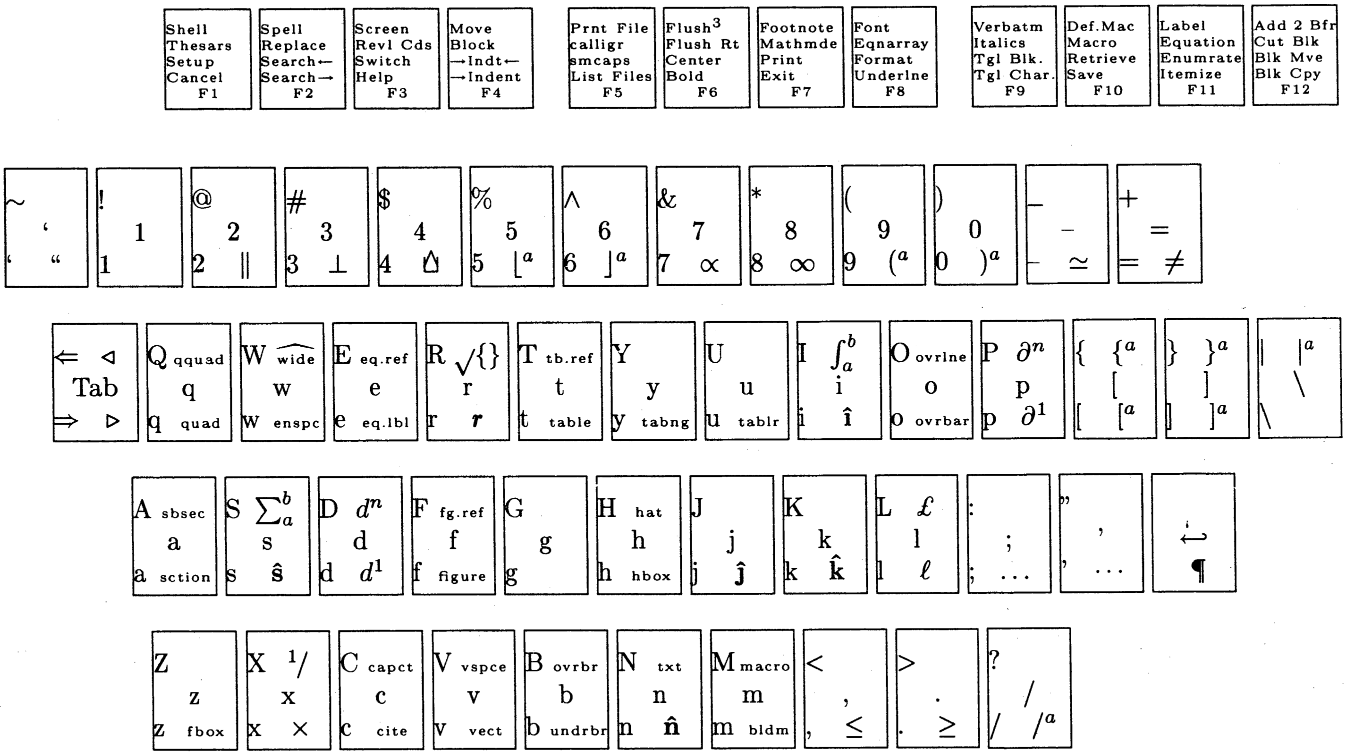
Many function keys in this keyboard have been made compatible with TEX (such as *indent*, *underline*, etc.) while others have been redefined. For example:

`^-F5` This key, newly labeled *Print File* translates a WordPerfect file (prepared with the present set-up) into a variety of documentstyles.

`F9` This key, labeled *Toggle Character*, switches into keyboard `SINGLE` for a single character, then switches back into `NORMAL`.

`F11` This key, labeled *Itemize*, produces a LATEX itemized list: the user is prompted for each item ending with a `<ret>`; items are by default labeled with a •, other labels can be used by enclosing them in square brackets (e.g., the present label was achieved by typing `[F11]` after the prompt); itemization is exited by hitting `<ret>` (to end each item), followed by `^-<ret>` (to exit list); lists may be nested by hitting `<ret>` twice (once to end item, second time to start nesting), followed by another

[1] The reason why the △ symbol is used at all is due to a WordPerfect weakness: WordPerfect is unable — even when using *Ramfonts* — to display two "attributes" at once, such as italics and subscripts. Therefore, I have chosen to display math-italic like ordinary text, with △ as delimiters.



[a] delimiters of self-optimizing size

**Figure 1**: The NORMAL keyboard

F11 (itemize) or shft-F11 (enumerate); nesting can be done to arbitrary levels, but only the first two levels of indentation are shown on the screen; after the nested list is exited (with ^-<ret>) another <ret> must be hit to get back into the original list. (LaTeX only.)

@-F11 This key, labeled *Equation*, switches into the Math Italic font to *display* an equation centered on a separate line (without equation number), ended by hitting the <ret> key; for visualization, the expression is surrounded by the ⌂⌂ symbol (on separate lines above and below the equation).

Since WordPerfect allows for the definition of other keys on the regular keyboard (by pressing *Alt* or *Ctrl* at the same time), many additional keys have been defined as indicated in Fig. 1. For example:

@-a Produces an invisible \section{'...} or \section*{...} command, first asking whether the section should be numbered (displayed by a preceding #-symbol) or unnumbered (preceding *), then prompting for a title, ended by <ret>; the title is displayed in "Very large Type". (LaTeX only.)

@-p Produces a visible \p{...}{...} (partial derivative) command, prompting for a numerator and a denominator, each ended by <ret>; for example, @-p T <ret> x <ret> results in $\frac{\partial T}{\partial x}$. (*Mathmode* only.)

^-i Displays and prints a visible integral sign, and prompts for a lower limit (displayed as subscript) and an upper limit (displayed as superscript), each ended by <ret>; for example, @-i x=0 <ret> @-8 <ret> results in $\int_{x=0}^{\infty}$. (*Mathmode* only.)

In addition, I have programmed a few keys of the cursor pad, and of the number key pad (not displayed in Fig. 1), e.g.:

@-↑ Displays and prints a single-character superscript.

Num-/ Produces a fraction {...\over ...}, prompting for numerator and denominator, each ended by <ret>, for example Num-/ x+2 <ret> x+1 <ret> produces $\frac{x+2}{x+1}$.

### GREEK and SINGLE keyboards

These keyboards contain the full Greek alphabet plus a number of math symbols as shown in Fig. 2. None of the function keys have been redefined, i.e., they contain the original definitions supplied by

WordPerfect. The SINGLE keyboard is identical to GREEK except that control is returned to NORMAL after each character, and is toggled by hitting the F9 key. Thus typing a F9 bc results in "a$\beta$c".

### FOREIGN, SMCAPS, CALLIGRAPHIC keyboards

For people who type in foreign languages I have defined the FOREIGN keyboard which allows you to type umlauts, etc. The keyboard is identical to NORMAL except that some characters have been replaced. At this point only some umlauts etc. have been redefined. For example, typing @-a gives ä, ^-a gives Ä, etc. The SMCAPS keyboard contains a displayable set[2] of *small caps* characters A, B, THROUGH Z. Control is returned to the normal keyboard by hitting <ret>. The CALLIGRAPHIC keyboard acts like SINGLE, i.e., after typing a single calligraphic character (upper case A through Z), control is returned to NORMAL automatically.

### Equation example

To demonstrate how the system works for typing equations, consider the equation

$$\mathbf{f}(x)=\frac{1}{4}\int_{\xi=0}^{\infty}\left[\frac{g(x,\xi)}{1-\xi^2}+1\right]\frac{\partial f}{\partial\xi}\hat{\mathbf{n}}d\xi$$

which is typed as:

$\boxed{\texttt{@-F11}}$ $\boxed{\texttt{F6}}$ f $\boxed{\texttt{F6}}$ (x)= $\boxed{\texttt{\^{}-x}}$ 4 $\boxed{\texttt{\^{}-i}}$ $\boxed{\texttt{F9}}$ x=0 $\boxed{\texttt{ret}}$
$\boxed{\texttt{@-8}}$ $\boxed{\texttt{ret}}$ $\boxed{\texttt{@-[}}$ $\boxed{\texttt{num-/}}$ g(x, $\boxed{\texttt{F9}}$ x) $\boxed{\texttt{ret}}$ 1- $\boxed{\texttt{F9}}$
x $\boxed{\texttt{num-}\uparrow}$ 2 $\boxed{\texttt{ret}}$ +1 $\boxed{\texttt{@-]}}$ $\boxed{\texttt{\^{}-p}}$ f $\boxed{\texttt{ret}}$ $\boxed{\texttt{F9}}$ x $\boxed{\texttt{ret}}$
$\boxed{\texttt{@-n}}$ d $\boxed{\texttt{F9}}$ x $\boxed{\texttt{ret}}$

and appears on the screen approximately as

⌂⌂
f(x)=$^1$/4$\int_{\xi=0}{}^{\infty}$$\Big[$\{g(x,$\xi$) \over 1-$\xi^2$\}+1$\Big]$\p\{f\}\{$\xi$\}$\hat{\mathbf{n}}$d$\xi$
⌂⌂

(The big brackets actually appear "broken" on the screen to indicate their variable size).

## 0.5 TeX Printer Drivers

As indicated in the description for the ^-F5 key above I have defined a number of "printers", which will take your WordPerfect files and translate them into TeX files (tex.prs), or into LaTeX documentstyle files (article, book, conference, letter, memo or sideways). Besides translating TeX and LaTeX commands (such as $\alpha$ to \alpha, etc.), these drivers also add the necessary commands at the beginning and end of the TeX file.

## 0.6 Custom Made TeX Macros

I have defined the following local macros which, therefore, are only available in my version of TeX/LaTeX:

\flush#1#2#3 Identical to the prompted ^-F6, i.e., this macro will place the text #1 flush left, text #2 centered, and text #3 flush right, e.g. \flush{left}c{right}:
left c right

\ls#1 This macro sets the linespacing, where #1 must be an integer number and sets the spacing in multiples of "jots" (1*jot* = 3*pt* = $\frac{1}{4}$ of regular line spacing); the default is \ls4, i.e., one line every 4 jots or 12 points, translating to six lines per inch, while this paragraph is surrounded by {\ls6...} (notice the "1$\frac{1}{2}$-spacing"?). The unit of "jot" was chosen to make simple input (\ls followed by a single digit) possible for fractional values of line spacing. (Enclosing the paragraph in {...} reestablishes former linespacing beyond the }). Note that this macro takes effect only as soon as TeX goes into "vertical mode" after it is called (i.e., between paragraphs); thus putting {\ls*n* ... } within a paragraph will have no effect at all.

\d#1#2 Total derivative of #1 with respect to #2 (*mathmode* only), e.g. \d Tx results in $\frac{dT}{dx}$ (accessed via @-d)

\p#1#2 Partial derivative of #1 with respect to #2 (*mathmode* only), e.g. \p Tx results in $\frac{\partial T}{\partial x}$ (accessed via @-p)

\dn#1#2#3 #1-th order total derivative of #2 with respect to #3 (*mathmode* only), e.g. \dn 4Tx results in $\frac{d^4T}{dx^4}$ (accessed via ^-d)

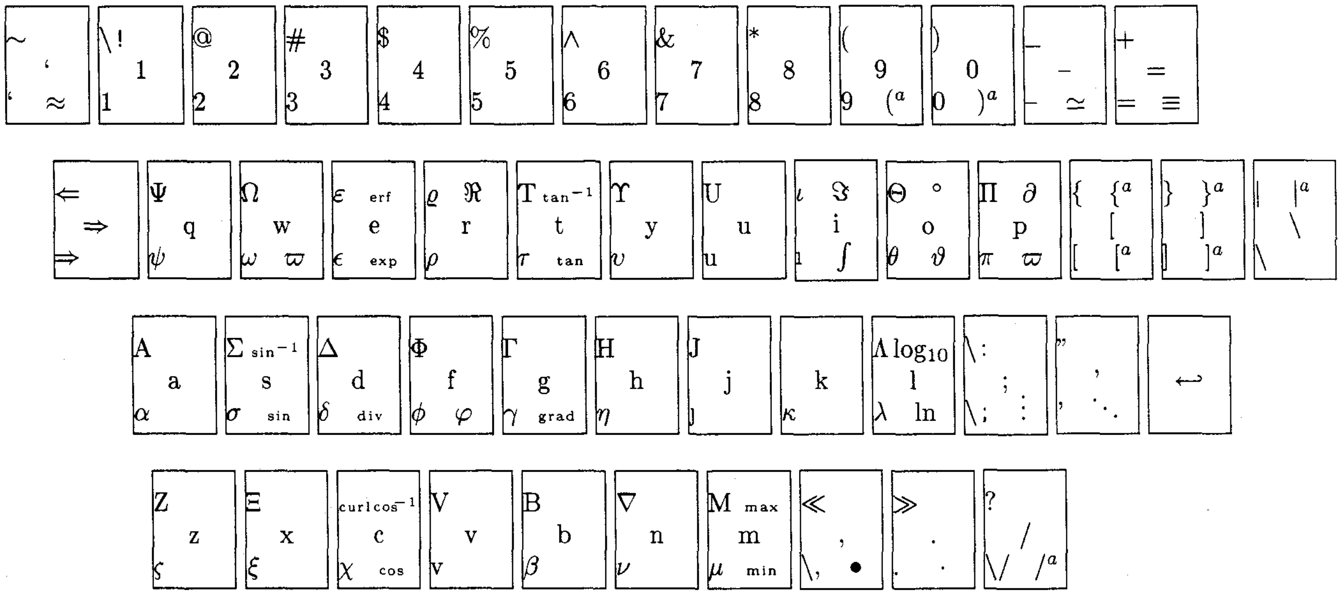
\pn#1#2#3 #1-th order partial derivative of #2 with respect to #3 (*mathmode* only), e.g. \pn 4Tx results in $\frac{\partial^4T}{\partial x^4}$ (accessed via ^-p)

\fr#1 #1-th fraction, (*mathmode and normal mode*), e.g. \fr4 →$\frac{1}{4}$ (accessed via ^-x)

\Cases{#1} is identical to the normal TeX \cases {#1} command, but <u>both</u> columns are typeset in mathmode.

\txt#1 Small text between equations like "and" can be inserted by typing \txt{text} (accessed via ^-n)

[2] While the normal WordPerfect command for *small caps* may be used, this would not *display* them (because of the 6 font limitation)



[a] delimiters of size "Big"

**Figure 2**: The GREEK and SINGLE keyboards

### 0.7 Typing Memos With LaTeX

The memo document style is designed to make a number of memos at once and is really just a slightly modified LaTeX letter style file. The first page of each memo will have a LaTeX-generated version of the PENNSTATE logo at the top (which, obviously, other users may want to modify to their needs).

### 0.8 Custom Journal Styles

Every journal has slightly different ways to arrange the list of references, to refer to references within text, to refer to equations, etc. I have defined customized .sty files for four types of journals in which I usually publish, viz. the Transactions of the American Society of Mechanical Engineers (asme.sty), journals of the American Institute of Aeronautics and Astronautics (aiaa.sty), the International J. of Heat and Mass Transfer (ijhmt.sty), and the J. of Quantitative Spectroscopy and Radiative Transfer (as well as many other *Pergamon Press* journals) (jqsrt.sty). Of these four styles the first one (asme) orders references alphabetically, while the other three order them sequentially.

For these bibliography styles I have redefined or newly defined a number of LaTeX macros, which act differently for different journals. These macros are discussed under *Custom Made LaTeX Macros* below.

### 0.9 Custom Made LaTeX Macros

I have written a number of TeX macros which act together with LaTeX macros, mostly for labeling and referencing. The following macros will behave differently depending on the journal style option chosen, i.e., asme, aiaa, ijhmt or jqsrt:

\cite{*key(s)*} will place the reference number (or year) into the text (i.e., the year within parentheses for asme, superscripted number for aiaa and jqsrt, and bracketed number for ijhmt). In addition, a BibTeX command will be placed for those that use a bibliography database (accessed via @-c).
**Note:** for the asme style the \bibitem command must have the year of publication as a label, i.e., \bibitem[year] {*key*}. (Of course, for BibTeX users this is taken care of automatically).

\Cite{*key(s)*} similar to \cite but the numbers are large and accompanied by "Ref." or "Refs.". Obviously, this macro is not defined for the asme style (since references are not numbered).

\capcite{*key(s)*} identical to \cite except that it only *reads* the .aux file (to display the correct reference number or whatever), but does not *write* an entry into the .aux file; therefore, \capcite can only be used if \cite is used elsewhere for the same reference; \capcite may be used in places where \cite cannot, primarily within figure and table captions.

\e{*key*} same as the \ref command but surrounds the number with parentheses.

`\eq`{*key*} will write something like "equation (*n*)" (depending on the journal), where *n* is the referenced equation (accessed via `^-e`).

`\eq+`{*key1*,*key2*} will write something like "equations (*n1*) and (*n2*)" (depending on the journal), where *n1* and *n2* are the referenced equations (accessed via `^-e`).

`\eq-`{*key1*,*key2*} will write something like "equations (*n1*) through (*n2*)" (depending on the journal), where *n1* and *n2* are the referenced equations (accessed via `^-e`).

`\eqs`{*key*} will write something like "equations (*n*)" (depending on the journal), where *n* is the first referenced equation (to be used with `\e` if some other inter-equation text than "and" or "through" is desired) (accessed via `^-e`).

`\Eq`{*key*}, ... All of the above `\eq`... commands exist also as `\Eq`... commands for equation references at the beginning of a sentence (accessed via `^-e`).

`\fg`{*key*}, `\fg+`{*key*}, `\fg-`{*key*}, `\fgs`{*key*}, ... All the `\eq` commands are repeated for figures. In this case all four journals have the same format, i.e., the `\fg` commands translate to "Fig. *n*" or "Figs. *n*", while the `\Fg` command translate to "Figure *n*" and "Figures *n*" (accessed via `^-f`).

`\tb`{*key*}, `\tb+`{*key*}, `\tb-`{*key*}, `\tbs`{*key*}. All the lower case `\fg` commands are repeated for tables. In this case all four journals have the same format, (and the upper case version does not exist since there is no difference between beginning-of-sentence and mid-sentence reference), i.e., the `\tb` commands translate to "Table *n*" and "Tables *n*" (accessed via `^-t`).

Another group of macros that I have written deals with cross-referencing between different documents. For example, if you write a long report or a book, you may want to break up the documents into parts or chapters, but without losing the ability to reference equations/figures/pages etc. from other parts, keeping up continuous page/chapter etc. numbering, preparing an overall table-of-contents, etc.:

`\crossref`{*file(s)*} will read the files *file(s)*`.crf` (if the entry is non-empty), and will open the new file *present-file-name*`.crf`;

`\closecrf` will close the file *present-file-name*`.crf` (if it is open, i.e., if `\crossref{...}` has been called within the document. This command should come **after** the `\end{document}` (and is automatically included in my WordPerfect `book` documentstyle);

`\TOC` will open and prepare a *present-file-name*`.toc` file similar to the `\tableofcontents` command, but will not actually print the Table of Contents (eventually one can splice all `.toc` files together to make a single Table of Contents);

`\LOF` is similar to `\TOC`, and is designed to be used instead of the `\listoffigures` command;

`\LOT` is similar to `\TOC`, and is designed to be used instead of the `\listoftables` command.

`\Section` is equivalent to the `\section*` command, but places an unnumbered entry into the Table of Contents;

`\Label` is equivalent to the `\label` command, but places a cross-referencing line not only into the `.aux` file, but also into the `.crf` file;

`\plcounter{#1}` places the value of a counter into the `.crf` file, (which can later be read by the `\crossref` command from a different document part). `#1` may be `page` (page number), `chapter` (chapter number), `section` (section number), `subsection` (subsection number), `figure` (figure number), `table` (table number), or `equation` (equation number). When placed at the very end of a file (last entry before the `\end{document}` command) this transmits to the `\crossref` command the latest page number etc., making continuous numbering possible.

`\widen{#1}` increases the document `\textwidth` by `#1`, and decreases both margins by $\frac{1}{2}\times$`#1`. This macro comes in handy when LaTeX breaks pages in such a fashion that only one or two lines end up on the last page (particularly in letters and memos).

### 0.10 BIBTeXing

Since the standard LaTeX formats are geared towards mathematical and computer journals (styles `plain` and `unsrt` are available), I have written a few more styles, viz. `asme`, `aiaa`, `jqsrt`, `ijhmt`, i.e., drivers for the aforementioned journals, which are primarily of interest to engineers and physicists. Others could be customized easily.

### 0.11 Availability

I am prepared to supply anybody who would like it with the software that I have developed (i.e., you supply your own WordPerfect and TeX). To receive the software you need to send $10 (to cover diskettes and postage), and specify whether you want 360kB or 1.2MB 5.25in floppies, or a 1.44MB 3.5in diskette. In addition I would appreciate any suggestions on how to further improve the present WordPerfect-to-TeX/LaTeX preprocessor.