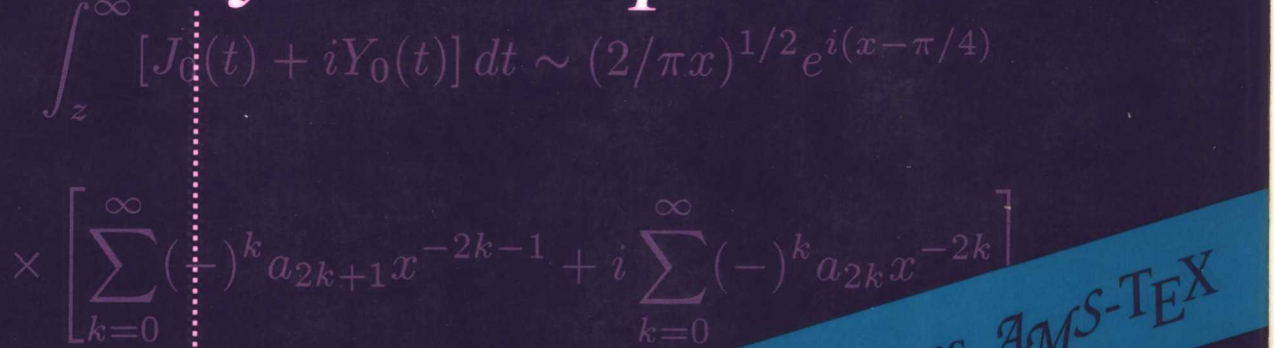
Mathematical TEX by Example



Also Covers AMS-TEX

```
$$ \int_z^\infty [J_0(t) + iY_0(t)]\,dt
\sim (2/\pi x)^{1/2} e^{i(x-\pi/4)} $$
$$ \times \left[ \sum_{k=0}^\infty
(-)^k a_{2k+1} x^{-2k-1} +
i\sum_{k=0}^\infty (-)^k
a_{2k} x^{-2k} \right] $$
```

Arvind Borde

Mathematical TEX by Example

Arvind Borde

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich, Publishers
Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

Copyright © 1993 by Academic Press, Inc.
All rights reserved.
No part of this publication may be reproduced or
transmitted in any form or by any means, electronic
or mechanical, including photocopy, recording, or
any information storage and retrieval system, without
permission in writing from the publisher.

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101–4311

Apart from its covers, the book was typeset in TEX by the author, with
assistance from the production department of Academic Press, Boston
A few examples were typeset with AMS-TEX, a package of commands
distributed by the American Mathematical Society The PostScript®
examples were produced with dvips, from Radical Eye Software
The formatting commands for the book are shown in Chapter 6

The final copy was produced by the American Mathematical Society
The PostScript examples were typeset on an Agfa/Compugraphic 9600
imagesetter, all other pages on an Autologic APS Micro-5 phototypesetter

'TEX' and 'AMS-TEX' are trademarks of the American Mathematical Society
'PostScript' is a trademark of Adobe Systems, Inc

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24–28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data

Borde, Arvind, date.
Mathematical TeX by Example / Arvind Borde
p. cm.
Includes bibliographical references and index
ISBN 0-12-117645-2
1. TeX (Computer file) 2. Computerized typesetting—Computer
programs. 3. Mathematics printing—Computer programs I Title
Z253.4.T47B67 1992b
686.2'2544—dc20 92-29048
CIP

Printed in the United States of America
92 93 94 95 96 97 MV 9 8 7 6 5 4 3 2 1

Preface

THIS BOOK is aimed at readers who are already broadly familiar with the basic uses of TEX: it seeks to introduce them to a variety of additional tools and techniques—mainly related to typesetting mathematics. Near the start of the book there is a series of reworkings in TEX of samples from standard mathematics books and journals. Some of these have been done with Plain TEX, others with $\mathcal{AMS}$-TEX, others with still other packages. The spirit here is precisely the one-example-is-worth-a-hundred-explanations attitude of my earlier book, *TEX by Example*. It is hoped that these real-world applications will prove a useful source of typesetting ideas.

The examples have been chosen with an eye towards variety. Because of this, they touch on a number of interesting additional issues: how TEX handles languages other than English, how non-standard typefaces may be used with TEX, how pictures may be included in TEX documents. Some readers may, in fact, find these discussions of additional issues to be the main attraction of the book. These issues are not covered in detail, except for a long discussion of typeface selection in TEX. Complete sets of alternate faces for typesetting mathematics with TEX have recently become available, and a systematic and thorough description seemed warranted. The book ends with a long Glossary/Index that covers all Plain TEX and $\mathcal{AMS}$-TEX commands that are likely to be of interest in typesetting mathematics.

Academic Press has been generously supportive throughout my association with them. I particularly thank David Pallai, my publisher, and Jenifer Swetland, my editor, for their original suggestion that I put together this book, and for their patience as it ballooned beyond the original plan for a slim mathematical companion to *TEX by Example*. I also thank Sue Purdy Pelosi, my production editor, for her suggestions and advice. Many other people have helped as well, some by reading parts of the manuscript, others by making suggestions or answering questions. They include Barbara Beeton, Matthew Choptuik, Rosanne Di Stefano, Victor Eijkhout, Michael Ferguson, Tomas Rokicki, Rainer Schöpf, Jim Simone and Michael Spivak. I thank them all. And, of course, I thank Donald Knuth—not only for making it all possible, but also for making it all so enjoyable.

Arvind Borde
Cambridge, Massachusetts
August 1992

Contents

此为试读,需要完整PDF请访问: www.ertongbook.com

1

Introduction

DONALD KNUTH composed TEX not just for "the creation of beautiful books," but "especially for books that contain a lot of mathematics." He has succeeded remarkably well: the briefest of reflections on the scissors-and-paste manipulations and the inking-in by hand of symbols that had to be undertaken in the old pre-TEX days, or on the clumsily composed formulas that emerged from other systems (and still emerge, even from some contemporary ones), will confirm the truth of this. Without any exaggeration at all, it can be asserted that TEX has wrought a quiet revolution in the way in which scientists and mathematicians are able to present their work, and even in how they communicate with each other.

But progress has its price. The very power of TEX leads its users to expect it to be all programs to all people, and when it isn't, there are grumblings that it is deficient in this way or that. Some of these grumblings are widely aired, and some touch—either directly or peripherally—on the typesetting of mathematics. Addressing such complaints is a secondary aim of this book. A discussion of what are often seen as weaknesses in TEX is particularly important at the present time. Many of the early contributors to TEX have moved on to other work, and even Donald Knuth has announced this as his intention (KNUTH [1990b]). There is much active debate, therefore, on how best to build on the successes of TEX and on how to maintain various TEX systems in the future. Some of these systems have institutional support—$\mathcal{AMS}$-TEX, for instance. Others are likely to continue to be supported by their creators, or are already officially in other hands—LATEX, for example, has been handed over by Leslie Lamport to an international committee (which has undertaken a major enhancement to that system).

Since developments are occurring at a rapid clip, it is impossible in a book of this kind to be fully up-to-date. Current information can usually be obtained from the TEX Users Group, or by posting questions on electronic bulletin boards. The inside back cover gives more information about these resources.

1.1 TEX Today

TEX is widely used. It not only serves as the basic typesetting mechanism for a number of mathematical and scientific journals, it has also been used to typeset a vast range of other material: everything from chess to Chinese. Designed from the start to be portable, it now also serves as the almost universal language of discourse when technical messages are exchanged by electronic mail. Still, criticisms are often voiced. Here is a list of prominent contemporary complaints, each followed by a few comments:

- *TEX is too powerful.* This is heard most commonly from editors and publishers. TEX has given their authors too much power. They *will* go ahead and do their own, often incompatible, things. They *will* define lots and lots of private new commands, perhaps even redefine some basic commands, then forget to tell anybody (and forget to supply files that contain the new definitions). They *will* fiddle with things like fine spacing, instead of confining themselves to the content of their documents and leaving the detailed adjustments to the professionals.

There is truth to all of these. Authors don't distinguish between privately produced reports like preprints—where they often have to adjust spacing, and so forth, themselves—and articles and books submitted for professional publication. Publishers invariably have their own style guidelines, and it is important (if the efficiencies that TEX promises are indeed to be realized) that authors not introduce commands of their own that might interfere with matters of house style. It is now becoming common—but is perhaps not yet done to a great enough extent—for publishing houses to supply instructions to their authors about what they can and cannot do, along with a style file (see § 1.7) that the authors are required to use. Following these guidelines may make for some tedium for authors, but it would be a lot more tedious for them in the long run if, in a backlash against the indiscriminate use of private commands, publishers stop accepting submissions in TEX altogether.

One of the problems with private commands is the possibility of clashes with others of the same name. There are mechanisms available to reduce the incidence of such clashes; for example, AMS-TEX provides a command called `\define` (to be used in place of `\def`) that first checks to see if a command with the chosen name already exists.

There is an extreme point of view that would ban any attempt by users to define their own formats, directing them always to some preexisting and professionally designed package. For documents designed for actual publication, this is a sound course of action. But TEX has a multiplicity of uses and, for at least one very important one, this course isn't possible: the use of TEX as a language of electronic communication. TEX has made it possible for users to

transfer documents back and forth, and to print hard copy at any installation at all, confident that it will be identical in appearance in all essential aspects to copies printed elsewhere. What is needed for such uses is the continued availability and use of small TEX packages, ones that travel light.

- *TEX is not powerful enough.* This complaint usually comes from authors, and it is more and more commonly heard. After the initial excitement of seeing TEX's beautifully formed formulas wears off (and the fact that TEX reliably produces such output every time becomes familiar and unremarkable), users sometimes start making comparisons with commercial word processors. Compared to the flashy effects that are often easy to achieve with such programs, the quiet pleasures of TEX—the perfectly placed superscript, the precise spacing around a binary relation—can seem unsatisfying, and the program can begin to appear limiting.

The perception about power is mostly false, though it can sometimes seem indisputably true. In principle, any of the showy effects of a commercial word processor or desktop publishing program can not only be imitated, but far surpassed by TEX and its companion program METAFONT. In practice, however, a lot of the necessary programming remains to be done, or when done, remains to be widely disseminated. Lists of companies and agencies that provide extensions to basic TEX can be obtained from several sources (see the inside back cover).

- *TEX offers only a few typefaces; it doesn't draw pictures.* These are special cases of the *not powerful enough* objection, but are so frequently voiced that they warrant explicit expression. The complaint about pictures is a long-standing one; the complaint about typefaces—and the sameness of appearance of all documents produced with TEX—is more recent.

Though both complaints are true of standard TEX distributions, there are solutions easily within everybody's grasp. For example, many different sets of typefaces may be used with TEX. Examples 7–9 in this book, for instance, have been done with non-standard typefaces and—for those interested in it—Chapter 5 tells the full story behind typeface selection in TEX. There are several sources of new typefaces, public domain and commercial. As for pictures, TEX has a simple mechanism whereby it can import graphics from other sources and, in fact, the program itself can be used to draw simple pictures through packages like PICTEX. Examples 10–13 display several pictures.

- *TEX is difficult to use.* People with mouse-driven word processors might encounter problems by clicking on the wrong thing and making unintended selections—problems usually easily remedied; mean-

此为试读,需要完整PDF请访问: www.ertongbook.com

TEX Today

while, TEX users might be staring blankly at

```
Overfull \hbox
```

or even more incomprehendingly at something like

```
! Missing { inserted.
<to be read again>
                    \fam
\rm ->\fam
                \z@ \tenrm
```

on the screen.

There is something to this complaint. For simple applications with simple layouts—preprints, technical reports, even simple books—using TEX is simplicity itself. For more complicated tasks, standard TEX packages are difficult to use, and they can quickly lead into a morass of incomprehensible error messages or to wildly unexpected results in the output. Again, the solution is to use a higher-level package of TEX commands, one that automatically provides the needed features—and several such packages now exist. Most features needed for typesetting mathematics are, for example, provided by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX, LA$\mathcal{M}\mathcal{S}$-TEX or $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX, and users are urged to investigate them. Higher-level commands may also be directly defined, but those who wish to do so must be prepared to invest a certain amount of initial time and effort. Chapter 6 carries the definitions of several such commands that were used in the making of this book; they can perhaps be used as models.

- *TEX doesn't distinguish between the logical structure of a document and its visual structure.*

This is partially true (even though the complaint itself will seem incomprehensible to those whose uses of TEX have so far kept them insulated from issues of design philosophy). The question is discussed in § 1.7.

- *TEX doesn't cost anything.* This isn't a complaint, exactly, but a source of suspicion among commercial publishers and other business organizations. Used for so long to having to pay good money for nothing, they find it hard to believe that it is possible to pay no money and get something. The makers of computer programs are, after all, allowed some of the quickest conversions of pure thought to pure cash known in human history, and it is hard for a lot of people to accept that a good program—let alone a great one—can be in the public domain.

There isn't a real answer to this, except to say, "Try it and see." But there is also a rational component to this otherwise irrational fear of the free: Who will provide support in case there are problems? Who will answer questions? The solution here is to shop around among vendors of TEX software—and many

exist—to find one with satisfactory support policies. TEX also comes bundled with some advanced small computer systems (usually those in the workstation class), and there is sometimes support from the group that supplied the TEX part of the bundle.

The wary relationship between TEX and its potential users outside the academic world is compounded by a further factor. TEX attracts a large number of highly enthusiastic adherents who have used its programmable aspects over the years to construct a fascinating variety of additions—some of them quite brilliant—often purely for the fun of it. But this very enthusiasm can seem like play from the outside—especially to those for whom somberness is synonymous with seriousness of purpose. Contributors to TEX are usually serious about what they do, but rarely somber.

1.2 The Flavors of TEX

At the core of TEX lies an immensely powerful, highly programmable computer system—a computer language, in some senses—specially directed to the needs of typesetting. Strictly speaking, it is just this core that constitutes TEX; the name, however, is almost always used more loosely to include an additional package (as discussed below). The commands that TEX (the core) comes with—called primitive commands—can be used to create practically any format at all. They are also practically impossible to use directly, except by people with a large amount of time, a large degree of skill and a large sense of humor (since working with the core of TEX can sometimes be like working with a very intelligent but very willful child).

It was never the intention, however, that all users of TEX get down on their hands and knees, logical wrench in hand, and create smoothly running typesetting machines. Rather, it was intended that professional format designers, building on the core that TEX provides, construct higher-level structures to meet the needs of users.

There are two widely used higher-level structures. The first is Plain TEX—though most of its users do not know it by that name. The structure is so widely used, and the functions that it performs are mostly so basic, that people usually refer to Plain TEX simply as TEX. In fact, the word "Plain" is conventionally not even capitalized. This book will break with that convention, since it is (or so it seems to the author) important that users realize the relationship between flavors of TEX, a relationship that is obscured when "plain TEX" is confused with a description of TEX rather than being seen as a name.

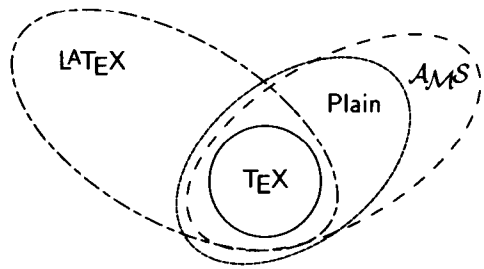
The other widely used structure, LATEX, is also built (by definition) out of the primitive commands of TEX. Thus, the LATEX-versus-TEX debates that sometimes rage are meaningless: the participants are really comparing LATEX

The Flavors of TEX

with Plain TEX. LATEX does offer considerably more to users, but it also imposes a certain rigidity of style. Among other offerings, it allows a choice—actually, it requires a choice—of document styles, and then provides a great deal of fully automated typesetting to go with that style; it automatically generates, on request, tables of contents; and it provides rudimentary graphical tools and a wider range of typefaces.

Both of these structures—geared as they originally were towards meeting the needs of the authors of technical reports and books, with their usually straightforward layouts—have deficiencies. These deficiencies have become more noticed and more commented on, as users have become more demanding. But some remedies are on hand. Version 3.0 of LATEX, for example, will provide more flexible styles. And, though Plain TEX superstructures are somewhat limited in their range of styles, they at least frequently meet the special needs of special groups of users. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX, distributed by the American Mathematical Society, is the most well known. Other packages include LA$\mathcal{M}\mathcal{S}$-TEX (discussed in some places in this book), TEXsis (a format developed by Eric Myers and Frank Paige) and `eplain` (an extension to Plain TEX developed by Karl Berry). They are all available at TEX archives. There is also an interesting recent 'meta-format' called `lollipop` (EIJKHOUT and LENSTRA [1991]).

Plain TEX, LATEX, and $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX are the most widely used of TEX superstructures. Here is a diagram that roughly illustrates their relationships to each other and to TEX:



The diagram isn't drawn to any particular scale, but it is meant to make clear that there are several commands (beyond the primitive ones) that are common to all three packages, and that LATEX otherwise goes off in a different direction from the other two. There are also a couple of commands that LATEX and Plain TEX have in common by themselves, and a few unique to Plain TEX. Other than that, Plain TEX is functionally pretty much a subset of $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX.

In this book, "TEX" will usually mean the area common to all these major packages; commands and attitudes particular to a particular package will be labelled as such. It will also be explicitly stated when the name is used in the strict sense, referring only to the primitive commands at the core.

Part of the reason why there is confusion in the minds of users about Plain TEX is that users often never see it directly. There is a file of commands called `plain.tex`, but it is rarely introduced explicitly with an `\input` statement, since that is an inefficient method for a large package. A more efficient method is to create a *format file* that contains pre-processed versions of the commands in `plain.tex`; this may be done with a variant of TEX called IniTEX. The new file has the extension `.fmt` and it is automatically included when the command to run TEX is given, saving users from having to include it explicitly themselves.

IniTEX can be used to create format files for other packages of commands as well. A particular format, say `pformat.fmt`, can be used by saying something like

```
tex &pformat
```

in place of the usual `tex` command (or by defining a new system command that contains `tex &pformat` within it).

1.3 What Is in this Book?

The main focus of the book is the typesetting of mathematics, although several peripheral issues are discussed as well along the way (as in this chapter).

Chapter 2 is the soul of the book: it contains a series of examples illustrating the uses of various flavors of TEX. Left-hand pages contain TEX input, with occasional notes given below, and right-hand pages contain the corresponding output. The examples consist of standard pieces of mathematics taken from a variety of books and journals, going back to the beginning of the century. It is hoped that these applications of TEX to "real life" will be found useful.

This part of the book is similar in style to the corresponding part of *TEX by Example* (BORDE [1992]). There is, however, an important difference. A deliberate attempt was made in that book to keep the examples as simple as possible. This meant that the input was also simple, allowing users to pretty much copy what they needed, just as it appeared there. Output pages here are more complex than those in that book, and the input is also more complex. Commands that appear in the input boxes are often not standard ones at all (see § 1.4). In order to use them, their definitions must also be copied. These definitions occur either in the input itself, close to where the command is used, or in one of the formatting files that were put together for the making of this book. The files are reproduced in Chapter 6.

Of course, if the example features a special TEX package like, say, PICTEX or LAMS-TEX, there is a good chance that a command in the input is not a standard command (it can be looked up in the Glossary to see if it is). In this case, users will have to get hold of the package in order to use the command.

What Is in this Book?

Most of the packages used in this book are in the public domain: information about them is on the inside back cover.

The examples intentionally make many different points. They all show pieces of mathematics (in a few cases, diagrams), but some demonstrate additional effects as well: document formatting, typeface switching, and the like. It is, in fact, under headings corresponding to some of these additional effects that the examples have been grouped, since it is a given that they all illustrate mathematics.

A piece of mathematics selected for display sometimes bears very little connection to the topic of its source: for instance, Example 2 is from a book called *Semi-Riemannian Geometry*, but the pieces of mathematics displayed there show partitioned matrices. This may make the classification chosen for the examples—by source, under broad categories that correspond to additional effects—seem perverse, but the apparently more logical grouping under typographical topic proved impossible. Real-world pieces of mathematics often simultaneously contain so many different typographical features—spacing, the positioning of equation numbers, an unusual alignment, and so on—that it was difficult to decide where to put some of them. The final decision on classification was a compromise, but one made more palatable by including entries in the Glossary for all the topics that might have been used, in a simpler world, to group the examples. The entries point to places in the book where illustrations of the topics are given, and they often briefly discuss the topics on the spot.

No effort was made to find "interesting mathematics" for the examples: the reasons for or against inclusion were purely typographical.

Chapter 3 gives a quick summary of the principal features of $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX. The chapter provides only an overview to go along with the uses of $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX in Chapter 2. Detailed information on $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX may be found in Michael Spivak's excellent manual, *The Joy of TEX* (SPIVAK [1982, 1990]).

Chapter 4 gives a very brief summary of $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX and $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX. The intention here is merely to let users know what is out there, not to do the packages justice.

Chapter 5 gives a detailed and systematic account of how TEX makes typeface decisions and how users can impose their own choices. TEX users have been timid in the past about exploring typeface lifts for their documents, but that now appears to be changing as more and more of them realize that it is indeed possible to enjoy the full power of TEX without also having to adopt its regulation appearance. An attempt has been made in Chapter 5 to tell the full story, starting at the beginning and omitting nothing. On the one hand, the discussion has been designed to be accessible even to somebody who has used TEX just once or twice and knows that `{\bf bold type}` will give **bold type**, `{\it italic type}` will give *italic type*, `{\sl slanted type}` will give *slanted type*, and `{\tt typewriter type}` will give `typewriter type`, but nothing

else. On the other hand, no details have been left out, so this discussion should also serve as a useful reference for experienced users.

The material in Chapter 5 should allow users to make practically arbitrary typeface changes—as long as the typefaces they want are indeed available on their computer systems.

Chapter 6 reproduces the definitions of most of the new commands that are introduced in this book.

These chapters are followed by a Bibliography. References in the book (e.g., ZAPF [1987]) are to sources listed here.

The book ends with a long Glossary/Index that explicitly lists all the commands of Plain TEX and $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX that are useful in typesetting mathematics. It also lists and discusses (from the typesetting point of view) a large number of mathematical topics, e.g., continued fractions or integrals.

1.4 Conventions

In the input for Examples 1 to 9, commands whose names are entirely in lower case are Plain TEX commands. They may be used freely. Examples 10 to 18 cover specialized packages, and the commands used there are often a mix of ones from the package and from Plain TEX.

Commands that contain capital letters in their names are usually special commands defined for this book. The definitions are all given in Chapter 6, and the definitions must first be copied if the commands are to be used. Occasionally, an $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX or a $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}$-TEX command will also contain capital letters. It will be clearly indicated when that is the case.

Terms that appear in sans serif type in the text are listed in the Glossary. In order to avoid excessive typographical distraction, such type has not been used promiscuously, only when it may indeed be worthwhile to look up more information.

The next few lines are aimed at experts; normal readers may safely skip to the next paragraph. Most of TEX's specialized expressions will be deliberately avoided throughout this book in the interest of making it accessible even to casual users who—like it or not—will want to browse. The expression "control sequence" will not be used at all—in most cases the word "command" will be used instead—and the word token will be used only when absolutely necessary.

A command that is reproduced in the middle of text will usually just be reproduced straight, as it would appear in the input: `\hfil`, `\vfil`, etc. If, however, there appears to be the possibility of misunderstanding what is part of the command and what is surrounding text, single quotes will be used around the command: for example, '`\hskip 1in`', or '`\.`'.

此为试读,需要完整PDF请访问: www.ertongbook.com