The P$_I$CT$_E$X Manual

Michael J. Wichura

The University of Chicago

First printing: November 1986
Second printing: September 1987

This manual was written using TeX supplemented by the PıCTeX macros. The features described herein exist in Version 1.1 of PıCTeX.

The author provides no guarantee as to the correctness of this manual and the associated software; the user accepts them as is.

PREFACE

In the preface to *The TEXbook*, Knuth describes TEX as a "typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics". PICTEX is a collection of TEX macros by means of which TEX users can easily instruct TEX to typeset beautiful pictures as a part of their books—and especially mathematical figures, such as the one below:
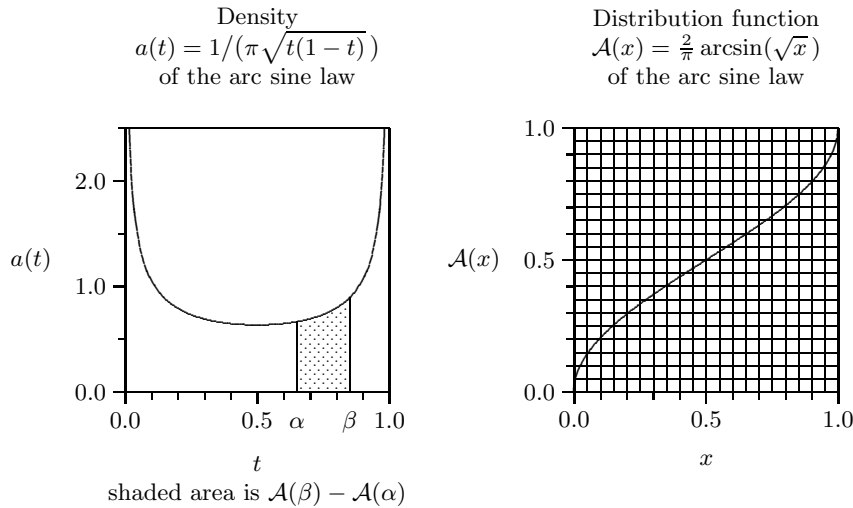
Density
$$a(t) = 1/(\pi\sqrt{t(1-t)}\,)$$
of the arc sine law

Distribution function
$$\mathcal{A}(x) = \tfrac{2}{\pi}\arcsin(\sqrt{x}\,)$$
of the arc sine law

shaded area is $\mathcal{A}(\beta) - \mathcal{A}(\alpha)$

FIGURE 1

That figure is destined to appear in a book on probability theory that I'm writing using TEX. Indeed, the PICTEX macros are an outgrowth of the attempt I made at coercing TEX into drawing the kinds of figures I wanted to include in that book. Happily, it turned out that what I wanted to be able to do fell just within the limits of what seems feasible by way of making TEX function as a graphics device.

PICTEX offers these advantages: (1) Figures become an integral part of the typesetting process. You can avoid having to leave the proper amount of space in your document for material that has to be created on some external device and later stripped into the finished product. (2) All of TEX's formatting capabilities are available for annotating your figures. In addition, that annotation will be done (if you so desire) in the same fonts as you're using in the rest of your document. (3) Just as TEX is machine independent, so too is PICTEX. It doesn't matter whether you're working on a PC or mainframe computer. (4) Since typeset figures are embedded in the `dvi` file along with the

1. INTRODUCTION

1.1. P$_I$CT$_E$X COMMANDS

To draw a P$_I$Cture you first have to load the P$_I$CT$_E$X macros into T$_E$X's memory—see your local system guru for details. To start drawing a P$_I$Cture you type the command \beginpicture, and to finish it off you type the command \endpicture. The overall structure is thus

```
\beginpicture
    additional PICTEX commands
\endpicture
```

All this goes in your T$_E$X input file.

The P$_I$CT$_E$X commands are described in detail in the following pages. A few words need to be said here about the syntax with which sample commands are presented. Consider, e.g., the \put command of Subsection 2.1:

\put {*text*} $\big[$[$[o_x][o_y]$]$\big]$ $\big[$<*xshift*, *yshift*>$\big]$ at *xcoord* *ycoord*

First of all, notice the matched pairs of thin brackets: $\big[$ $\big]$ and [ ]. In contrast to the thick brackets '[' and ']', these are <u>not</u> part of the command; rather they indicate that the phrases contained therein may be omitted. Secondly, note the blank spaces separating the various phrases of the command. These are <u>essential</u>. However to enhance the readability of your input, you can use as many spaces as you like (provided there is at least one) wherever a sample command shows a single space. Moreover, at least one blank <u>must</u> follow every P$_I$CT$_E$X command. Do try hard to get the syntax of the P$_I$CT$_E$X commands right, paying particular attention to the delimiters {, }, [,], <, >, (,), and /. If you mess up, T$_E$X will get confused and you will get a lot of error messages—and quite possibly no P$_I$Cture. Appendix C lists all the P$_I$CT$_E$X commands alphabetically, so you can easily check the syntax of any command you write.

Exercise 1. B. L. User didn't pay attention to the three words that were underlined in the preceding paragraph. What price did he pay in consequence?

It is important that you understand T$_E$X's concept of grouping (see Chapter 5 of *The T$_E$Xbook*). The point is that any change you make to one of P$_I$CT$_E$X's parameters is local to the group in which that change is made. In particular, since \beginpicture and \endpicture mark the start and end of a group, T$_E$X will undo the changes you've made while drawing a P$_I$Cture once it reaches the terminating \endpicture command. Consider, e.g., the effect of P$_I$CT$_E$X's \setcoordinatesystem command, which is discussed in the next subsection. There is a major difference between typing

APPENDICES

A. ANSWERS TO ALL THE EXERCISES

1. Untold hours of frustration.

2. The reference point of the first system is at its origin, while the reference point of the second system is 20 x-units to the left of its origin. Since 1 x-unit in the second system amounts to 10 pts, the origin of the second system is 200 pts to the right of that of the first system.

3. (1) He should have omitted the ['s and]'s. (He presumably skimmed Subsection 1.1 and so misinterpreted the []'s that appear in the statement of the syntax of the \setcoordinatesystem command on page 3.) (2) He should have left a space after 'units'. (3) He should have typed '1in,2in' instead of '1inch,2inches'. (Page 57 of *The TEXbook* lists the only ways dimensions can be written). (4) He should have typed '3 -2' instead of '(3,-2)'.

4.
```
\put {$\bullet$} at 1 2
\put {$(1,2)$} [l] <10pt,0pt> at 1 2
```

5.
```
\put {\ninepoint$\bullet$} at 1 2
\put {\ninepoint $(1,2)$} [l] <10pt,0pt> at 1 2
```
or (better)
```
\ninepoint
\put {$\bullet$} at 1 2
\put {$(1,2)$} [l] <10pt,0pt> at 1 2
```

6. The author used

```
\setcoordinatesystem units <1pt,1pt>
\put {\vrule height .4pt width 300pt} [l] at 0 0
\multiput {\vrule height 18pt} [t] at  0 0  *3 100 0  /
\multiput {\vrule height 14pt} [t] at  0 0  *6  50 0  /
\multiput {\vrule height 10pt} [t] at  0 0  *30 10 0  /
\multiput {\vrule height  6pt} [t] at  5 0  *29 10 0  /
\put {0 pt}   [t] at   0 -24   \put {100 pt} [t] at 100 -24
\put {200 pt} [t] at 200 -24   \put {300 pt} [t] at 300 -24
```

7. You get: $(3,2)$ $\rightarrow$ $\times$ stacking
                                              is
                                              as
                                           simple
                                          as ABC

8. You get: $(3,2)$ $\rightarrow$ $\times$ Rows of lines
                                            are easy to put
                                            into a PiCture

9. (a) No. In the first case the word 'lines' sits just above the point $(3,2)$, whereas in the second case the word 'Two' sits just above this point. (b) Yes.

CONTENTS