©TEXGuru, August 16, 1999

附录E　计算机现代字体

当 Donald E. Knuth 发明 TEX程序时，他也同时给它装备了丰富的字符集或字体。 Knuth 称这些字体为 *计算机现代字体*，这样就不必依赖于所用计算机上的可用字体。在当时，打印机字体质量并不是很高，当然也不会一致。利用他所提供的字体， TEX能在所有打印机上生成相同的高质量输出。

LATEX 当然继承了这些字体，因此这些字体已经成为由 TEX 或 LATEX 生成的文档的标志。而而并不是必需如此的，因为 LATEX 并不需要与任何特定的字体集合捆绑在一起，特别是在当今新字体选择框架出现 (8.5 节) 之后，新框架极大地简化了字体安装的步骤。

但是对绝大多数 TEX/LATEX 用户而言，计算机现代字体仍然起着重要作用，因此有必要对其做一详细介绍。即使我们在 NFSS 中只是指定了一种字体属性，最终这些属性组合还是要与已有的某种字体名称联系在一起，见 8.5.8 节。对于标准安装版本，这些名称就来自于本附录余下部分描述的字体集合。

§E.1　简介

排版学就是对字样进行研究与分类的科学，而字样指的是可以上溯到五个半世纪前 Gutenberg 发明的活字 (*movable type*)(不是指中国人发明的活字印刷) 。从那时开始，人们创建了相当多的字体族，并把其分类为 Baskerville, Caramond, Univers, 等等。如此字体族中每个成员拥有同样的整体设计，或者称基本式样，但是可以有 slanted，斜体，黑体或细体等变体；当然，它们可以有不同的尺寸。

对字体族的分类通常是根据某种规则进行，主要的考虑原则是它们的用途。

Serif 字体 是一种在边缘有小水平线 (也称为衬线， serifs) 的字体，这样眼睛看起来感觉很好。经验表明这种字体适于阅读，因此通常用作正文主体。按 NFSS 的术语，这些字体称为 *罗马*(*Roman*) 字体。

Sans serif 字体 一种缺少衬线的字体。具有赤裸外形的这种字体通常用在标题或者页眉中。比较 sans serif 和 regular text 这两种字体就可以知道它们的区别了。

固定字体 是一种具有相同宽度的字体，它是从打字机字体演化而来，进而进入到计算机字体的行列中。按照正统的观点来看，在书籍印刷中是没有这种字体的用武之地的，因为这时成比例字体 (字母 i 要比 m 窄) 占主导地位。

装饰性字体 是一种由于某些不寻常之处而显得突出的字体。通常用它来吸引眼睛的注意力。这一族在形状和宽度方面还不完整，通常用在广告中。

数学字体 是数学作品中所需要的特殊符号全体。对它的进一步分类是无法与对文本字体的分类相提并论的。

书籍设计人员必须决定要使用的字样族类型。他 / 她有可能在正文用罗马 (有衬线的) 字体，页眉用 sans serif 字体，而且如果书中还有计算机程序代码，就用一种固定字体表示这部分内容。最后，如果作品中有数学内容，还需要符号字体。

在计算机现代字体集合中包含所有如此种类的族。然而，由于它们是在 NFSS 属性分类系统建立之前出现的， CM 字体命名法并不与这种框架完全一致。 NFSS 系统把用户从必须记住 CM 字体名称的细节中解脱出来，而只需要指定属性。当然， NSFF 安装中必须有字体描述文件 `.fd` ，这个文件把属性组合转化为真实的字体名称，或者进行某种过得去的替代。

§E.2 TEX基本字体的分类

任何特定的字体都可以用在 LaTeX中，方法之一就是利用 `\newfont`(4.1.6 节) 命令直接给出字体的名称，另外一种方法则是用 `\usefont`(8.5.1 节) 或 `\DeclareFixedFont`(8.5.4 节) 给出其属性。

所有 TEX 字体文件的名称都是以 `cm`(表示 “Computer Modern”) 字母开头的，后接一至四个字母描述字体的样式，最后是一两个数字指定设计尺寸的点数。这也就是用在 `\newfont` 命令中的字体基本名。因此基本名的完整语法为：

`cm`*xxnn*　　*xx*= 样式， *nn*= 设计尺寸

基本字体文件的扩展名为 `.tfm` ，含义为 “TEX font metric” 。

在 `.tfm` 文件中并没有包含生成符号的信息，其只是由字符尺寸信息组成，这些尺寸包括宽度，高度和深度。对于 slanted 字体，每个字母还有 “倾斜校正”(3.5.1 节) 。而且，要为一些字母组合指定与通常不同的间距，例如 AV 与 AV 的区别，或者有可能进行连写。最后， `.tfm` 中还包含字符的斜率 (对于非 slanted 字体，这个值为零) ，标准单词间距及其伸展和收缩量 (9.5.4 节) ， `em` 和 `quad` 间距的大小，句子结尾处间距。数学和符号字体在其 `.tfm` 文件中还包含更多的信息。

`\newfont` 命令的一般语法是

```
\newfont{\ 字体 }{ 基本名 scaled 因子 }
```

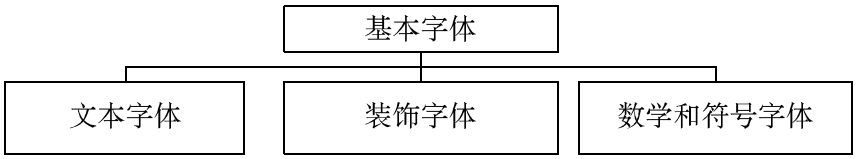
其中 *基本名* 指的是字体文件的外部基本名， *因子* 就是对字体进行放大时所采用的放缩因子的 1000 倍。这样所得的新命令 *\字体* 就是一个声明，它激活指定的字体，作用范围为当前环境结束，或者又调用了新的字体命令。如果所用的是 LaTeX ，那么 *因子* 可以取任何值，因为处理时就用 `.tfm` 文件中的数乘上这个因子。

也可以用下述命令得到同样的效果：

```
\DeclareFixedFont{\字体}{编码}{族}{序列}{尺寸}
```

这条命令自动地把 \字体 定义成适当的字体，必要时进行放缩。

下面我们给出 CM 字体的名称，并说明它们如何遵从 NFSS 属性模式。它们的组织如下：



§E.3 文本字体

用来设置普通文本的字体可以分为三族，其编码都为 OT1 。

§E.3.1 罗马字体族

罗马字体族就是有衬线的字体，间距成比例。这些字体通常充满了正文的主体。

cmr 族， OT1 编码

形状 =	n	sc	sl	it	u
序列					
m	cmr5 cmr6 cmr7 cmr8 cmr9 cmr10 cmr12 cmr17	cmcsc10	cms8 cms9 cms10 cms12	cmti7 cmti8 cmti9 cmti10 cmti12	cmu10
b	cmb10				
bx	cmbx5 cmbx6 cmbx7 cmbx8 cmbx9 cmbx10 cmbx12		cmbxsl10	cmbxti10	

直立罗马字体

直立罗马字体组包括在书籍印刷中最经常使用的标准字符集。字体样式标志为字母 `r`，因此在这一族中所有文件名都以 `cmr` 开头，后接以点为单位的设计尺寸。设计尺寸的变化范围从 8 到 17 pt 。`cmcsc10` 字体也属于该族，但形状为 `sc` 。这里的 `csc` 代表大写和小大写 (*captial and small caps*) 。因为在该族中的大写字母与 `cmr10` 中的一样，但小写字母是一种介于 `cmr7` 和 `cmr8` 之间尺寸的一种较小的大写字母。

`cmr10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´˘˜¯˚^¨˝˜ ¸˙·,;:¿?¡¡’“”- – —*+/=( )[ ]

`cmcsc10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
ABCDEFGHIJKLMNOPQRSTUVWXYZ ↑ ↓ ' ¡ ¿ ÆŒØSSIJ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´˘˜¯˚^¨˝˜ ¸˙·,;:>?<¡’“”- – —*+/=( )[ ]

仔细查看 `cmcsc10` 字体，并与 `cmr10` 进行对比，我们会注意到所有小写字母的高度是一样的 (介于 `cmr7` 和 `cmr8` 的相应字母大小之间)，但宽度要大一些，而且字母间距也要大些。另外连写被其它符号代替了。

`cmu10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#£%ℭ
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´˘˜¯˚^¨˝˜ ¸˙·,;:¿?¡¡’“”- – —*+/=( )[ ]

`cmu10` 字体实际上是一种没有倾向的斜体。这样就得到了一种直立的斜体字母。在 NFSS 框架中它占据了一种特殊的形状属性 `u` 。

Slanted 字体

本组的字体来源于直立罗马字体，只是对它进行斜率为 1/6 的倾斜；也就是说，这些字体向右倾斜了高度的六分之一。*Slanted* 字体有介于 8 到 12 pt 间的四种设计尺寸。字体样式标志为 `sl`，因此所有成员的文件名以 `cmsl` 开头，后接的 `8`, `9`, `10` 或 `12` 表示设计尺寸。这些字体属于形状定义 `sl` 。

`cmsl10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&

©TEXGuru, August 16, 1999

abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˜˘¯°ˆ¨˝˜¸˙,;:¿?¡¡’“”- – —+/=( )[ ]*

斜体

在斜体字体组中的字体在基本形式上不同于直立和 slanted 罗马字体。它的倾斜值为 1/4，这要比 slanted 字体斜得更大。形状属性的标志为 `it`，而字体样式标志为 `ti`，表示文本斜体 (*text italic*)。因此字体文件的名称开头为 `cmti`，后接设计尺寸值。有从 7 到 12 pt 之间的五种尺寸。(数学字符字体属于另外一种斜体族，样式标志为 `mi`，代表数学斜体 (*math italic*)。)

`cmti10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#£%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˜˘¯°ˆ¨˝˜¸˙,;:¿?¡¡’“”- – —+/=( )[ ]*

黑体序列

这组字体还是由罗马字符组成, 只不过现在字样要黑一些, 线要粗一些。字体样式标志为 `bx`，代表 *bold extended*，因为这些字体要比同样高度的相应罗马字体宽一些。这也是它们的 NFSS 序列标志。提供介于 5 到 12 pt 之间的七种设计尺寸，名称分别为 `cmbx5` ... `cmbx12`。对于 10 pt 的设计尺寸，同时还有一种 slanted 和斜体变体，名称为 `cmbxsl10` 和 `cmbxti10`。

在 10 pt 的设计尺寸中还包含另外一种字体 `cmb10`，这种黑体中的字母具有与普通罗马字母同样的宽度。其字体样式标志为 `b`，代表通常黑体。

`cmbx10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˜˘¯°ˆ¨˝˜¸˙,;:¿?¡¡’“”- – —*+/=( )[ ]

`cmbxsl10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˜˘¯°ˆ¨˝˜¸˙,;:¿?¡¡’“”- – —*+/=( )[ ]

`cmbxti10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#£%ℭ
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
*ΓΔΘΛΞΠΣΥΦΨΩ` ´ˇ˘¯ ° ^ ¨ ˝ ˜ ¸ ˙ , ; : ¿ ? ¡ ¡ ’ “ ” - – — * + / = ( ) [ ]*

cmb10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ` ´ˇ˘¯ ° ^ ¨ ˝ ˜ ¸ ˙ , ; : ¿ ? ¡ ¡ ’ “ ” - – — * + / = ( ) [ ]

§E.3.2 Sans serif 族

所有 sans serif 字体在其名称中都有样式标志 `ss`，因此其名称开头为 `cmss`。这也是其 NFSS 族的名称。

cmss 族， OT1 编码

形状 =	n	sl	—
序列			
m	cmss8 cmss9 cmss10 cmss12 cmss17	cmssi8 cmssi9 cmssi10 cmssi12 cmssi17	
bx	cmssbx10		
sbc	cmssdc10		
—			cmssq8 cmssqi8 cminch

直立 sans serif 字体

介于 8 到 17 pt 之间有五种设计尺寸的 sans serif 字体可以使用，名称分别为 `cmss5 ... cmss17`。还有一种字体 `cmssq8`，称为 *sans serif quotation*，其中的小写字母比通常情形按比例放大了一些。这种字体并不是按标准 NFSS 属性框架进行分类的。

cmss10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ` ´ˇ˘¯ ° ^ ¨ ˝ ˜ ¸ ˙ , ; : ¿ ? ¡ ¡ ’ “ ” - – — * + / = ( ) [ ]

cmssq8

©TEXGuru, August 16, 1999

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯˚^¨˝˜¸˙,;:¿?¡!'“”-–—*+/=( )[ ]

Slanted sans serif 字体

Slanted sans serif 字体是与同尺寸和编号的直立字体对应的。样式标志为 `ssi` 和 `ssqi`，其中的 `i` 代表倾斜 (inclined)，而不是斜体 (italic)。正是因为如此，它们被赋予属性 `sl`。倾斜大致为 1/5。另外，quotation 字体 `cmssqi8` 并不包含在 NFSS 框架中。

`cmssi10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯˚^¨˝˜¸˙,;:¿?¡!'“”-–—+/=( )[ ]*

`cmssqi8`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯˚^¨˝˜¸˙,;:¿?¡!'“”-–—+/=( )[ ]*

Sans serif 黑体

只有设计尺寸为 10 pt 的 sans serif 黑体。其中一个名为 `cmssbx10`，表示 sans serif bold extended，其与 `cmss10` 的关系就如同 `cmbx10` 与 `cmr10` 的关系。另外一种黑体是 `cmssdc10`，表示 sans serif demibold condensed。

`cmssbx10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯˚^¨˝˜¸˙,;:¿?¡!'“”-–—*+/=( )[ ]

`cmssdc10`

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯˚^¨˝˜¸˙,;:¿?¡!'“”-–—*+/=( )[ ]

---

cminch 字体

在 cminch 字体只有大写字母和 0,1,...,9 的数字，而再没有其它符号。其中的字符高度均为一英寸。字体的名称是一个例外，因为这时并没有用数字表示设计尺寸。对于这种情形，inch 既表示字体样式，也表示尺寸。在 NFSS 中也没有这种字体的分类。

cminch

§E.3.3 打字机字体

对于固定字体，每个字符都具有同样的宽度。甚至单词之间的距离也等于字体的宽度，而并不会进行伸展或收缩。然而，同所有 TEX 字体一样，单词间距总量并不与输入的空格数多少有关。标准的固定字体，都有一个字符表示打字机样式，它们是：

cmtt 族， OT1 编码

形状 =	n	sc	sl	it	—
序列					
m	cmtt8 cmtt9 cmtt10 cmtt12	cmtcsc10	cmsltt10	cmitt10	
—					cmtex8 cmtex9 cmtex10 cmvtt10

直立打字机字体

直立打字机字体有介于 8 到 12 pt 之间的四种设计尺寸。样式标志 tt 表示打字机类型 (*typewriter type*) 。因此文件基本名为 cmtt8 ... cmtt12 。

cmtt10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%& ΓΔΘΛΞΠΣΥΦΨΩ
abcdefghijklmnopqrstuvwxyz æœø ßıȷ¡¿ ()[]{}\|/↑↓
0123456789 -+*<=> ␣_␣.,:;!?‘’~`´¨ˇ˘¯°^ˈ"¨

大写与小大写

打字机大写与小大写字体 cmtcsc10 生成 10 pt 打字机字体的大写字母，而小写字母则为 8 pt 的打字机大写字母。

cmtcsc10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%& ΓΔΘΛΞΠΣΥΦΨΩ
ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ SSIJ¡¿ ()[]{}\|/↑↓
0123456789 -+*<=> ␣_␣.,:;!?‘’~`´¨ˇ˘¯°^ˈ"¨

有斜度的打字机字体

只在 10 pt 的设计尺寸时才有两种有斜度的打字机字体。 Slanted 的打字机字体类型 cmsltt10 与直立 tt 字体一样，只是具有斜率 1/6 。而另一方面，斜体打字机字体 cmitt10 用的则是斜体设计的字母，斜率为 1/4 。

cmsltt10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%& ΓΔΘΛΞΠΣΥΦΨΩ
abcdefghijklmnopqrstuvwxyz æœø ßıȷ¡¿ ()[]{}\|/↑↓
0123456789 -+<=> ␣_␣.,:;!?‘’~`´¨ˇ˘¯°^ˈ"¨*

---

cmitt10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#£%ℭ ΓΔΘΛΞΠΣΥΦΨΩ
abcdefghijklmnopqrstuvwxyz æœø ßıȷ¡¿ ()[]{}\|/↑↓
0123456789 -+<=> ␣_␣.,:;!?‘’~`´¨ˇ˘¯°^ˈ"¨*

数学公式中的打字机字体

在数学打字机字体中的字母和数字与相应的直立 tt 字体一样。而其它字符都被特殊的数学符号代替。

这些字符集的设计尺寸有 8, 9 和 10 pt ， 文件基本名为 cmtex8, cmtex9 和 cmtex10 ， 表示 *typewriter extension* 。

这些字体也不适合于 NFSS 框架。实际上, 它们是一组相当独特的群体，只是数学字体 cmex 的打字机类型版本。在任何 LaTeX版本中其都没有任何作

用。

cmtex10

·↓αβΛ¬∈πλγδ↑±⊕ϖ∂⊂⊃∩∪∀∃⊗↯↔→≠◇≤≥≡∨ !"#$%&’()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ [\]^_‘abcdefghijklmnopqrstuvwxyz{|}~∫

可变打字机字体

另外一种在 NFSS 属性系统中没有立足之地的独特字体是 cmvtt10，这种字体中的字符形状与打字机字体一样，但是间距是变化的。

cmvtt10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯°ˆ¨˝˜ ¸˙,;:¿?¡¡’“”- – —*+/=( )[ ]

§E.4 装饰性字体族

有三族装饰性字体或者称特殊字体，它们只有唯一的设计尺寸和有效的属性。

	杂类族， OT1 编码， m 序列		
	形状 =	n	it
族	名称		
cmfr	Funny	cmff10	cmfi0
cmfib	Fibonacci	cmfib8	
cmdh	Dunhill	cmdunh10	

cmfr *Funny Roman*，有两种字体组成，一种是 cmff10，它具有负的倾斜度 -0.1，也就是说向左而不是向右倾斜；另一种是 cmfi10，它具有正的斜率 0.1。小写字体也要比通常的高。

cmfib *Fibonacci*，来源于数的 Fibonacci 序列。严格地讲，它是一种没有增宽的黑体，但却具有中间序列属性。

cmdunh *Dunhill*，其中的小写字母与 10 pt 时字符高度相同，但是大写字母，数字和 b, d, f, h, k, l 等有支柱的字母高度有些儿夸张。

cmff10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ`´ˇ˘¯°ˆ¨˝˜ ¸˙,;:¿?¡¡’“”- – —*+/=( )[ ]

©TEXGuru, August 16, 1999

cmfi10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#£%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ ˋˊˇ˘˜¯˚˙¨˝˛¸·,;:¿?¡!’“”- – —+/=( )[ ]*

cmfib8

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ ˋˊˇ˘˜¯˚˙¨˝˛¸·,;:¿?¡!’“”- – —*+/=( )[ ]

cmdunh10

ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØ @#$%&
abcdefghijklmnopqrstuvwxyz ff fi fl ffi ffl æœøßıȷ 0123456789
ΓΔΘΛΞΠΣΥΦΨΩ ˋˊˇ˘˜¯˚˙¨˝˛¸·,;:¿?¡!’“”- – —*+/=( )[ ]

§E.5 数学与符号字体

数学与符号字体就是由数学公式中需要的特殊字符组成。另外，在 LaTeX 的 picture 环境中也需要用到特殊符号，例如斜线，箭头，都可以在 LaTeX符号字体中找到。最后提一下，一些特殊记号字体也包括其中。

§E.5.1 数学字体族

数学字体所具有的编码框架与普通文本中的字体没有任何关联。正是出于这个原因，每一种字体都具有自己的编码和族定义。

	数学字体族		
编码 =	OML	OMS	OMX
族 =	cmm	cmsy	cmex
形状 =	it	n	n
序列			
m	cmmi5 cmmi6 cmmi7 cmmi8 cmmi9 cmmi10 cmmi12	cmsy5 cmsy6 cmsy7 cmsy8 cmsy9 cmsy10	cmex10
b	cmmib10	cmsyb10	

数学文本字体

这些字体由大写和小写的拉丁字母与希腊字母组成，还包括其它一些符号。由于在公式中有各种名称的斜体，而这些是基本的斜体字体，因此其字体样式标志为 `mi`，表示数学斜体 (*mathematical italic*)，以与通常的斜体区分开。字体有介于 5 到 12 pt 之间的七种设计尺寸。对于 10 pt 的设计尺寸，还有一种数学斜黑字体 `mib10`。

`cmmi10`

ΓΔΘΛΞΠΣΥΦΨΩαβγδϵζηθικλμνξπρστυϕχψωεϑϖϱςφ↼↽⇀⇁ ̓ ̔▷◁
0123456789.,</>⋆∂*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
♭♮♯⌣⌢*ℓabcdefghijklmnopqrstuvwxyzıȷ℘*⃗⁀

`cmmib10`

ΓΔΘΛΞΠΣΥΦΨΩαβγδϵζηθικλμνξπρστυϕχψωεϑϖϱςφ↼↽⇀⇁ ̓ ̔▷◁
0123456789.,</>⋆∂*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
♭♮♯⌣⌢*ℓabcdefghijklmnopqrstuvwxyzıȷ℘*⃗⁀

数学符号

这些符号字体给出的是公式中其它数学符号，但尺寸可以变化的除外。字体样式代码为 `sy`，表示符号 (*symbol*)。有介于 5 到 10 pt 之间的六种设计尺寸。在 10 pt 时还有一种黑体符号字体 `cmbsy10`。这些符号字体的文件基本名为 `cmsy5 ... cmsy10`。

`cmsy10`

−·×∗÷⋄±∓⊕⊖⊗⊘⊙◯∘•≍≡⊆⊇≤≥⪯⪰∼≈⊂⊃≪≫≺≻
←→↑↓↔↗↘≃⇐⇒⇑⇓⇔↖↙∝′∞∈∋△▽/ ̸ ∀∃¬∅ℜℑ⊤⊥
ℵ𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵∪∩⊎∧∨
⊢⊣⌊⌋⌈⌉{}⟨⟩|‖↕⇕\≀√∐∇∫⊔⊓⊑⊒§†‡¶♣♢♡♠

`cmbsy10`

−·×∗÷⋄±∓⊕⊖⊗⊘⊙◯∘•≍≡⊆⊇≤≥⪯⪰∼≈⊂⊃≪≫≺≻
←→↑↓↔↗↘≃⇐⇒⇑⇓⇔↖↙∝′∞∈∋△▽/ ̸ ∀∃¬∅ℜℑ⊤⊥
ℵ𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵∪∩⊎∧∨
⊢⊣⌊⌋⌈⌉{}⟨⟩|‖↕⇕\≀√∐∇∫⊔⊓⊑⊒§†‡¶♣♢♡♠

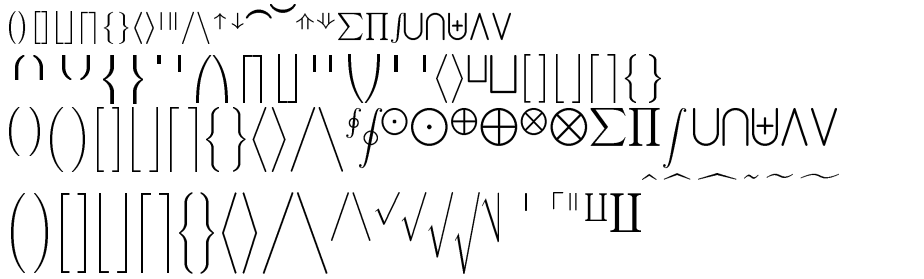
可变尺寸的符号

那些尺寸会随着公式其它部分自动放缩的符号包括在 `cmex10` 字体中，

©TEXGuru, August 16, 1999

这里 ex 表示 extension，其只有 10 pt 的设计尺寸。

cmex10



§E.5.2 其它字符字体

前面给出的 75 种字体是 TEX实现所提供的标准字体。也可以从一些商业机构或者公开来源得到其它的字体。要想知道那底有哪些字体可以使用，需要咨询一下你所在的计算机中心，或者查阅使用手册。

在 LATEX宏包中还有其它的一些字体，其中包括一些相当特殊的符号以及 picture 环境的构造元素，例如斜线和箭头。

还有一些记号字符字体，也包含在 TEX实现中，在 Donald E. Knuth 的书籍 *The TEXbook* 和 *The METAFONTbook* 中用到了它们。

LATEX的 lasy 字体

作为对数学符号字体 cmsy 的推广，LATEX提供了从 5 到 10 pt 六种设计尺寸的其它一些符号。这些字体文件的基本名为 lasy5 ... lasy10。其中每个包含 16 个符号： ⌔ ⌕ ◁ ◁ ▷ ▷ ↻ ⋈ □ ◇ ∼ ⇝ ⊏ ⊐ 。

在 LATEX 2ε中，只有用了 latexsym 宏包，这些字体才有定义。

生成图形的字体

在 LATEX的 picture 环境中所使用的特殊图形元素被保存在名称分别为 line10, lcircle10, linew10 和 lcirclew10 的字体文件中。前两个用来生成 \thinlines(6.5.1 节) 起作用时的直线、卵形线和圆。斜线和箭头 (6.4.3 和 6.4.4 节) 可以在 line10 中找到，而圆与卵形线片断 (6.4.5 和 6.4.6 节) 可以在 lcircle10 中找到。后面那对字体名称中增加了 w，表示粗线，这是用在 \thicklines 起作用的情形中。

记号字体

在记号字体 logo8, logo9, logo10, logosl10 和 logobf10 中只有七个字母 A, E, F, M, N, O, T，用来生成记号

METAFONT　*METAFONT*　**METAFONT**

§E.6 字体中的字符对应

除 cminch 外所有标准 TEX字体都是由 128 个字符组成，在 TEX内部给它们赋为 0 到 127 的数值。在字体中字符的布局见下面的表格。这里用的是八进制，而相应的十进制数放在对应符号旁边。

字体布局 1 表示的是罗马字体 cmr10 的字符对应关系，这是真正的 OT1 编码框架，是绝大多数具有同样符号的字体代表，因为可能在不同的样式中，这些符号具有相同的位置。文本斜体只是稍微有点儿区别。而大写及小大写字体 cmcsc10 则偏离得就有点儿多了，因为这时没有了连写。打字机字体则表明了其与布局 1 中的 OT1 标准顺序差得更远了。最后，数学和符号字体则具有截然不同的对应关系，因为其内容与文本字体并没有任何关联。所有这些有偏差的字体模式演示在布局 2–8 中。

注意：将来 TEX3.0 的字体中应包含 256 个字符。它们要遵从 T1 对应关系。

	0	1	2	3	4	5	6	7
’00x	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7
’01x	Φ 8	Ψ 9	Ω 10	ff 11	fi 12	fl 13	ffi 14	ffl 15
’02x	ı 16	ȷ 17	` 18	´ 19	ˇ 20	˘ 21	¯ 22	° 23
’03x	¸ 24	ß 25	æ 26	œ 27	ø 28	Æ 29	Œ 30	Ø 31
’04x	˙ 32	! 33	” 34	# 35	$ 36	% 37	& 38	’ 39
’05x	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
’06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
’07x	8 56	9 57	: 58	; 59	¡ 60	= 61	¿ 62	? 63
’10x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
’11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
’12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
’13x	X 88	Y 89	Z 90	[91	“ 92	] 93	^ 94	˙ 95
’14x	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103
’15x	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
’16x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119
’17x	x 120	y 121	z 122	– 123	— 124	˝ 125	˜ 126	¨ 127

字体布局 1 字符字体为 cmr10。这里给出的是字符与相应数值的标准对应关系。在下面的布局中给出的是非标准赋值。

©TEXGuru, August 16, 1999

	0	1	2	3	4	5	6	7
'00x	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7
'01x	Φ 8	Ψ 9	Ω 10	↑ 11	↓ 12	' 13	¡ 14	¿ 15
'02x	ı 16	ȷ 17	` 18	´ 19	ˇ 20	˘ 21	¯ 22	° 23
'03x	¸ 24	ß 25	æ 26	œ 27	ø 28	Æ 29	Œ 30	Ø 31
'04x	- 32	! 33	” 34	# 35	$ 36	% 37	& 38	’ 39
'05x	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
'06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
'07x	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
'10x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
'11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
'12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
'13x	X 88	Y 89	Z 90	[91	“ 92	] 93	^ 94	˙ 95
'14x	‘ 96	A 97	B 98	C 99	D 100	E 101	F 102	G 103
'15x	H 104	I 105	J 106	K 107	L 108	M 109	N 110	O 111
'16x	P 112	Q 113	R 114	S 115	T 116	U 117	V 118	W 119
'17x	X 120	Y 121	Z 122	– 123	— 124	˝ 125	~ 126	¨ 127

字体布局 2 字符字体为 `cmcsc10` 。与布局 1 的差别在于 11–15, 60 和 62 号符号。通常位于 11–15 号的连写被其它一些符号所取代。

	0	1	2	3	4	5	6	7
'00x	*Γ* 0	*Δ* 1	*Θ* 2	*Λ* 3	*Ξ* 4	*Π* 5	*Σ* 6	*Υ* 7
'01x	*Φ* 8	*Ψ* 9	*Ω* 10	*ff* 11	*fi* 12	*fl* 13	*ffi* 14	*ffl* 15
'02x	*ı* 16	*ȷ* 17	` 18	´ 19	ˇ 20	˘ 21	¯ 22	° 23
'03x	¸ 24	*ß* 25	*æ* 26	*œ* 27	*ø* 28	*Æ* 29	*Œ* 30	*Ø* 31
'04x	- 32	*!* 33	” 34	# 35	*£* 36	% 37	*&* 38	’ 39
'05x	*(* 40	*)* 41	* 42	+ 43	, 44	- 45	. 46	/ 47
'06x	*0* 48	*1* 49	*2* 50	*3* 51	*4* 52	*5* 53	*6* 54	*7* 55
'07x	*8* 56	*9* 57	*:* 58	*;* 59	*¡* 60	= 61	*¿* 62	*?* 63
'10x	@ 64	*A* 65	*B* 66	*C* 67	*D* 68	*E* 69	*F* 70	*G* 71
'11x	*H* 72	*I* 73	*J* 74	*K* 75	*L* 76	*M* 77	*N* 78	*O* 79
'12x	*P* 80	*Q* 81	*R* 82	*S* 83	*T* 84	*U* 85	*V* 86	*W* 87
'13x	*X* 88	*Y* 89	*Z* 90	*[* 91	“ 92	*]* 93	^ 94	˙ 95
'14x	‘ 96	*a* 97	*b* 98	*c* 99	*d* 100	*e* 101	*f* 102	*g* 103
'15x	*h* 104	*i* 105	*j* 106	*k* 107	*l* 108	*m* 109	*n* 110	*o* 111
'16x	*p* 112	*q* 113	*r* 114	*s* 115	*t* 116	*u* 117	*v* 118	*w* 119
'17x	*x* 120	*y* 121	*z* 122	– 123	— 124	˝ 125	~ 126	¨ 127

字体布局 3 字符字体为 `cmti10` 。与布局 1 的差别在于第 36 号 (*£* 代替了 $) 和第 38 号 (*&* 代替了 &) 符号。所有其它文本斜体都具有与之相同的模式。

	0	1	2	3	4	5	6	7
’00x	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7
’01x	Φ 8	Ψ 9	Ω 10	↑ 11	↓ 12	' 13	¡ 14	¿ 15
’02x	ı 16	ȷ 17	` 18	´ 19	˘ 20	¯ 21	˚ 22	¸ 23
’03x	ß 24	æ 25	œ 26	ø 27	Æ 28	Œ 29	Ø 30	Å 31
’04x	␣ 32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39
’05x	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
’06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
’07x	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
’10x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
’11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
’12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
’13x	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95
’14x	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103
’15x	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
’16x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119
’17x	x 120	y 121	z 122	{ 123	| 124	} 125	~ 126	¨ 127

字体布局 4 字符字体为 cmtt10 。所有的 tt 字体都具有同样的模式，只是 cmvtt10 例外，其模式与布局 1 一致。差别在于 11–15, 60, 62, 92, 123, 124 号。而且斜体打字机字体 cmit 用斜体字体 £ 和 ℘ 取代了第 36, 38 号符号。 cmtcsc10 字体在 97–122 号是更小的大写字母。

	0	1	2	3	4	5	6	7
’00x	· 0	↓ 1	α 2	β 3	∧ 4	¬ 5	∈ 6	π 7
’01x	λ 8	γ 9	δ 10	↑ 11	± 12	⊕ 13	ϖ 14	∂ 15
’02x	⊂ 16	⊃ 17	∩ 18	∪ 19	∀ 20	∃ 21	⊗ 22	↹ 23
’03x	← 24	→ 25	≠ 26	◇ 27	≤ 28	≥ 29	≡ 30	∨ 31
’04x	32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39
’05x	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
’06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
’07x	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
’10x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
’11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
’12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
’13x	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95
’14x	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103
’15x	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
’16x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119
’17x	x 120	y 121	z 122	{ 123	| 124	} 125	~ 126	$\int$ 127

字体布局 5 字符字体为 cmtex10 。与布局 4 的差别在于 0–31 号和 127 号字符，现在这里包含的是特殊数学符号。

©TEXGuru, August 16, 1999

	0	1	2	3	4	5	6	7
’00x	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7
’01x	Φ 8	Ψ 9	Ω 10	α 11	β 12	γ 13	δ 14	ϵ 15
’02x	ζ 16	η 17	θ 18	ι 19	κ 20	λ 21	μ 22	ν 23
’03x	ξ 24	π 25	ρ 26	σ 27	τ 28	υ 29	ϕ 30	χ 31
’04x	ψ 32	ω 33	ε 34	ϑ 35	ϖ 36	ϱ 37	ς 38	φ 39
’05x	$\leftharpoonup$ 40	$\leftharpoondown$ 41	$\rightharpoonup$ 42	$\rightharpoondown$ 43	$\triangleleft$ 44	$\triangleright$ 45	$\triangleright$ 46	$\triangleleft$ 47
’06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
’07x	8 56	9 57	. 58	, 59	< 60	/ 61	> 62	$\star$ 63
’10x	∂ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
’11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
’12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
’13x	X 88	Y 89	Z 90	$\flat$ 91	$\natural$ 92	$\sharp$ 93	$\smile$ 94	$\frown$ 95
’14x	ℓ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103
’15x	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
’16x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119
’17x	x 120	y 121	z 122	$\imath$ 123	$\jmath$ 124	$\wp$ 125	$\vec{}$ 126	$\frown$ 127

字体布局 6 字符字体为 cmmi10 。包括 cmmib10 在内的 cmmi 族字体， 11–39 号上是小写希腊字母， 40–47, 60–64 和 123–127 号上是一些数学符号。

	0	1	2	3	4	5	6	7
’00x	$-$ 0	$\cdot$ 1	$\times$ 2	$*$ 3	$\div$ 4	$\diamond$ 5	$\pm$ 6	$\mp$ 7
’01x	$\oplus$ 8	$\ominus$ 9	$\otimes$ 10	$\oslash$ 11	$\odot$ 12	$\bigcirc$ 13	$\circ$ 14	$\bullet$ 15
’02x	$\asymp$ 16	$\equiv$ 17	$\subseteq$ 18	$\supseteq$ 19	$\leq$ 20	$\geq$ 21	$\preceq$ 22	$\succeq$ 23
’03x	$\sim$ 24	$\approx$ 25	$\subset$ 26	$\supset$ 27	$\ll$ 28	$\gg$ 29	$\prec$ 30	$\succ$ 31
’04x	$\leftarrow$ 32	$\rightarrow$ 33	$\uparrow$ 34	$\downarrow$ 35	$\leftrightarrow$ 36	$\nearrow$ 37	$\searrow$ 38	$\simeq$ 39
’05x	$\Leftarrow$ 40	$\Rightarrow$ 41	$\Uparrow$ 42	$\Downarrow$ 43	$\Leftrightarrow$ 44	$\nwarrow$ 45	$\swarrow$ 46	$\propto$ 47
’06x	$\prime$ 48	∞ 49	$\in$ 50	$\ni$ 51	$\triangle$ 52	$\triangledown$ 53	/ 54	$\mid$ 55
’07x	$\forall$ 56	$\exists$ 57	$\neg$ 58	$\emptyset$ 59	$\Re$ 60	$\Im$ 61	$\top$ 62	$\bot$ 63
’10x	$\aleph$ 64	$\mathcal{A}$ 65	$\mathcal{B}$ 66	$\mathcal{C}$ 67	$\mathcal{D}$ 68	$\mathcal{E}$ 69	$\mathcal{F}$ 70	$\mathcal{G}$ 71
’11x	$\mathcal{H}$ 72	$\mathcal{I}$ 73	$\mathcal{J}$ 74	$\mathcal{K}$ 75	$\mathcal{L}$ 76	$\mathcal{M}$ 77	$\mathcal{N}$ 78	$\mathcal{O}$ 79
’12x	$\mathcal{P}$ 80	$\mathcal{Q}$ 81	$\mathcal{R}$ 82	$\mathcal{S}$ 83	$\mathcal{T}$ 84	$\mathcal{U}$ 85	$\mathcal{V}$ 86	$\mathcal{W}$ 87
’13x	$\mathcal{X}$ 88	$\mathcal{Y}$ 89	$\mathcal{Z}$ 90	$\cup$ 91	$\cap$ 92	$\uplus$ 93	$\wedge$ 94	$\vee$ 95
’14x	$\vdash$ 96	$\dashv$ 97	$\lfloor$ 98	$\rfloor$ 99	$\lceil$ 100	$\rceil$ 101	$\{$ 102	$\}$ 103
’15x	$\langle$ 104	$\rangle$ 105	$\mid$ 106	$\|$ 107	$\updownarrow$ 108	$\Updownarrow$ 109	$\backslash$ 110	$\wr$ 111
’16x	$\surd$ 112	$\amalg$ 113	∇ 114	$\int$ 115	$\sqcup$ 116	$\sqcap$ 117	$\sqsubseteq$ 118	$\sqsupseteq$ 119
’17x	§ 120	† 121	‡ 122	¶ 123	$\clubsuit$ 124	$\diamondsuit$ 125	$\heartsuit$ 126	$\spadesuit$ 127

字体布局 7 字符字体为 cmsy10 。在 cmsy 字体中有许多数学符号，以及 65–90 号位置上是花体字母 $\mathcal{A}\ldots\mathcal{Z}$ 。黑体版本 cmbsy10 具有完全一样的模式。

	0	1	2	3	4	5	6	7
'00x	0	1	2	3	4	5	6	7
'01x	8	9	10	11	12	13	14	15
'02x	16	17	18	19	20	21	22	23
'03x	24	25	26	27	28	29	30	31
'04x	32	33	34	35	36	37	38	39
'05x	40	41	42	43	44	45	46	47
'06x	48	49	50	51	52	53	54	55
'07x	56	57	58	59	60	61	62	63
'10x	64	65	66	67	68	69	70	71
'11x	72	73	74	75	76	77	78	79
'12x	80	81	82	83	84	85	86	87
'13x	88	89	90	91	92	93	94	95
'14x	96	97	98	99	100	101	102	103
'15x	104	105	106	107	108	109	110	111
'16x	112	113	114	115	116	117	118	119
'17x	120	121	122	123	124	125	126	127

字体布局 8 字符字体为 cmex10，由外观尺寸可以变化的各种数学符号组成。

©TEXGuru, August 16, 1999

	0	1	2	3	4	5	6	7
’00x	Њ 0	Љ 1	Ц 2	Э 3	І 4	Є 5	Ъ 6	Ћ 7
’01x	њ 8	љ 9	ц 10	э 11	і 12	є 13	ђ 14	ћ 15
’02x	Ю 16	Ж 17	Й 18	Ё 19	Ѵ 20	Ѳ 21	S 22	Я 23
’03x	ю 24	ж 25	й 26	ё 27	ѵ 28	ѳ 29	s 30	я 31
’04x	¨ 32	! 33	” 34	Ђ 35	◡ 36	% 37	´ 38	’ 39
’05x	(40	) 41	* 42	ђ 43	, 44	- 45	. 46	/ 47
’06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
’07x	8 56	9 57	: 58	; 59	« 60	ı 61	» 62	? 63
’10x	˘ 64	А 65	Б 66	Ц 67	Д 68	Е 69	Ф 70	Г 71
’11x	Х 72	И 73	Ј 74	К 75	Л 76	М 77	Н 78	О 79
’12x	П 80	Ч 81	Р 82	С 83	Т 84	У 85	В 86	Щ 87
’13x	Ш 88	Ы 89	З 90	[91	“ 92	] 93	Ь 94	Ъ 95
’14x	‘ 96	а 97	б 98	ц 99	д 100	е 101	ф 102	г 103
’15x	х 104	и 105	ј 106	к 107	л 108	м 109	н 110	о 111
’16x	п 112	ч 113	р 114	с 115	т 116	у 117	в 118	щ 119
’17x	ш 120	ы 121	з 122	– 123	— 124	№ 125	ь 126	ъ 127

字体布局 9 字符字体为 `wncyr10`，这是一种由华盛顿大学提供的 Cyrillic 字体。

§E.7 Cyrillic 字体

包含 Cyrillic 字母表的字体并不是通常 TEX/LATEX宏包的一部分。在 1980 年美国数学会设计了一组该类字体，用于综述以俄语和其它斯拉夫语出版的书籍，因为这时需要用原来的语言输出标题。在 1988 年，华盛顿大学人文与艺术计算中心对 $\mathcal{AMS}$的 Cyrillic 字体进行了重新设计，以适应通常斯拉夫语研究的需要，给其增加了革新前的以及有重音的字母。字母的全局外观也大大改善 (见字体布局 9) 。

华盛顿大学的 Cyrillic 字体名都是前缀 `wncy`, 后接样式标志 `r`(直立), `b`(黑体), `i`(斜体), `sc`(小大写), 或者 `ss`(直立 sans serif 字体)，然后是设计尺寸的点数。

可以如通常那样用 `\newfont` 定义 Cyrillic 字体命令。例如,

```
\newfont{\cyr}{wncyr10}
```

就会把 `\cyr` 定义成一个声明，激活 10 pt 的直立 Cyrillic 字体。也可以用其它任何字体样式或者尺寸来调用这条命令。

Cyrillic 字体的布局经过了精心选择，这样输入文本时就可以直接输入通常的英文翻译。因此，在 83 号位置上的 Cyrillic C 就是通常拉丁字母 S 所在的地方。当所用字体为 Cyrillic 时，输入 S 就会输出等价的 C 。数学和标点符号也可以在标准位置上找到，因此可以如常输入。 “Санкт-Петербург 10?”

就是用输入 {\cyr Sankt-Peterburg 10?} 生成的。

激活 Cyrillic 字体的更好方法就是在 NFSS 下只要选择字体编码 OT2 就可以了，见 206 页上 8.5.2 节的演示。相应于 Cyrillic 字体的 .fd(字体定义)文件在建立时所采取的方式也与拉丁字体完全一致。这也就是说， wncyr10 具有与 cmr10 完全一样的属性，差别就在于后者的编码： cmr 族， n 形状， m 序列。(当然，如果 cm 字体不是当前标准字体，那么所需要做的也就只是选择 OT2 编码以激活 Cyrillic 字体。)

由于在 Cyrillic 字母表中拥有比拉丁字母表多的字母，其中有些字母就必须翻译成多个字母的组合。这一点是利用 TEX的连写系统来自动用程序实现的。例如， Ch 就看成是 81 号符号 Ч 的连写，这就如同在拉丁字体中把 fi 看成连写 *fi* 一样。也就是说只要简单地输入多字母翻译就可以了。生成 “Хрущев” 的输入就是 {\cyr Khrushchev}，其中 Kh → Х， shch → щ 。这种翻译连写框架是针对于英语的；其它语言也可以用自己的系统来复制这种拼写。例如， “Горбачёв” 在德语中就是 *Gorbatschow*，在意大利语中是 *Gorbaciov*，而在英语中是 *Gorbachev* 。其它框架在这种字体中并不会有作用。

并不是所有的字母都能如此自动生成 (例如 Горбачёв 中的 ё)，因此 $\mathcal{AMS}$提供了一个名为 cyracc.def 的文件，由重音字母和其它特殊功能 (例如软硬符号) 的宏定义组成。当在拉丁字体中调用这些宏，就会出现另外的翻译连写符号。

不幸的是， cyracc.def 中有些重音命令在 LATEX 2_ε下无法使用。其需要一个类似于 OT1enc.def 和 T1enc.def 的编码定义文件 OT2enc.def，利用 8.5.9 节给出的特殊命令定义那些与编码有关的命令。

§E.8 字体文件

§E.8.1 基本名

在每个 TEX实现中都应该包含列在 E.3 至 E.5 节中各种不同设计尺寸的多达 75 种标准字体。这些字体的基本名汇总如下：

cmr5	cmti9	cmssq8	cmu10	cmmi5
cmr6	cmti10	cmss8	cmff10	cmmi6
cmr7	cmti12	cmss9	cmfi10	cmmi7
cmr8	cmbx5	cmss10	cmdunh10	cmmi8
cmr9	cmbx6	cmss12	cmtt8	cmmi9
cmr10	cmbx7	cmss17	cmtt9	cmmi10
cmr12	cmbx8	cmssqi8	cmtt10	cmmi12
cmr17	cmbx9	cmssi8	cmtt12	cmmib10
cmcsc10	cmbx10	cmssi9	cmtcsc10	cmsy5

```
cmsl8   cmbx12    cmssi10   cmsltt10  cmsy6
cmsl9   cmb10     cmssi12   cmitt10   cmsy7
cmsl10  cmfib8    cmssi17   cmtex8    cmsy8
cmsl12  cmbxsl10  cmssdc10  cmtex9    cmsy9
cmti7   cmbxti10  cmssbx10  cmtex10   cmsy10
cmti8   cminch    cmvtt10   cmex10    cmbsy10
```

$\mathcal{AMS}$提供了黑体数学斜体 `cmmib` 以及黑体符号字体 `cmbsy`，设计尺寸为 5–9 pt。安装 LaTeX 2ε时，就假定这些字体是存在的，虽然如果没有它们的话，也没有什么坏处。

```
cmmib5  cmmib6  cmmib7  cmmib8  cmmib9
cmbsy5  cmbsy6  cmbsy7  cmbsy8  cmbsy9
```

除了标准 TeX字体外，还有一些特殊的记号字体以及 LaTeX字体：

```
lasy5  lasy8   lasyb10    linew10     logo8   logobf10
lasy6  lasy9   line10     lcirclew10  logo9
lasy7  lasy10  lcircle10  logosl10    logo10
```

在基本名末尾的数字表示设计尺寸的点数。对这些字体中的每一个，都存在一个以此为基本名，扩展名为 `.tfm` 的文件，例如 `cmr10.tfm`。在 TeX和 LaTeX处理文本文件的过程中，只用到这些 `.tfm` 文件。其由尺寸信息组成，还有字体中的其它字符以及符号，见 356 页上的说明。但 `.tfm` 文件并没有说明符号到底是什么样子的。

只有要用驱动程序把输出送到打印机上才会需要如何构造这些字母和符号的信息。这些信息存贮在其它的文件中，除各种基本尺寸外，还可以进行不同的放大。

§E.8.2 字体放大

TeX字体通常的放大幅度是 1.2 的幂次以及 $\sqrt{1.2}$。在 LaTeX中，字体尺寸的选择是利用 4.1.2 节的命令实现的，这些命令会检测是否有要求尺寸的相应字体，如果没有，就会对另外一种设计尺寸进行适当的放大。LaTeX2.09 是从 `lfonts.tex` 文件中接受字体信息的，而 LaTeX 2ε用的则是字体定义文件 (`.fd`, 8.5.8 节)。对字体的放大也可以用下述命令来选择：

```
\newfont{\ 字体命令 }{ 基本名  scaled  放缩因子 }
```

这样就会把 `\`字体命令 定义成激活放大 *放缩因子*/1000 倍的 *基本名* 字体的命令。也就是说 *放缩因子* 是等于放大倍数 1000 倍的整数。如果用的就是设计尺寸，那该数就是 1000；对第一个放大幅度 $\sqrt{1.2}$，其值为 1095；然后接下来就是 1200, 1440, ...。TeX命令 `\magstep`n 可以生成 1.2^n，而 `\magstephalf` 就是 $\sqrt{1.2}$。

10 pt 的字体被放大 1.2 倍并不与设计尺寸为 12 pt 的字体相同，虽然这种差别是很难发现的，例如，

12pt unscaled 10pt scaled by 1200

当放大倍数是 1.2^3，即放缩因子为 1728 时差别就比较明显了：

17pt unscaled 10pt scaled by 1728

放大的字体虽然可能具有与原始字体同样的高度，但字符显得更黑，并且要宽一些。这是因为小设计尺寸字体在字母高度和宽度以及线粗等方面的关系与大尺寸字体的不同。然而，当把它们放大到同样高度时，所有其它度量都同比例地放大。当字体以设计尺寸显示时是最好看的。

包含字体像素式样的文件完整名称与打印机分辨率以及代码系统有关。这些文件的基本名仍然是前面给出的，但扩展名要反映放缩因子或者基本分辨率。对于 300 dpi(每英寸小点数) 的比较常见的激光打印机，扩展名为：

放大倍数	放缩因子	像素代码 源 300pxl	像素代码 源 200pxl	压缩代码
1.0	1000	1000pxl	1500pxl	300pk
$\sqrt{1.2}$	1095	1095pxl	1643pxl	329pk
1.2	1200	1200pxl	1800pxl	360pk
1.2^2	1440	1440pxl	2160pxl	432pk
1.2^3	1728	1728pxl	2592pxl	519pk
1.2^4	2074	2074pxl	3110pxl	622pk
1.2^5	2488	2488pxl	3732pxl	746pk
1.2^6	2986	2986pxl	4479pxl	896pk
1.2^7	3583	3583pxl	5373pxl	1075pk

对于 10 pt 罗马字体的字符集，在放大 1.2^2 倍后的像素文件的名称应为 `cmr10.1440pxl`, `cmr10.2160pxl` 或者 `cmr10.432pk` 。前两者的差别在于：所有像素文件都是针对于特定分辨率设计的，如果用的是 300 dpi，那么就会得到正确的输出。在 `pxl` 前面的数字就是放缩因子。然而，如果像素文件是针对于 200 dpi 的打印机创建的，那么乘上 2/3 因子后的输出结果就会太小。为了校正这个问题，需要再进行 1.5 倍的放大。

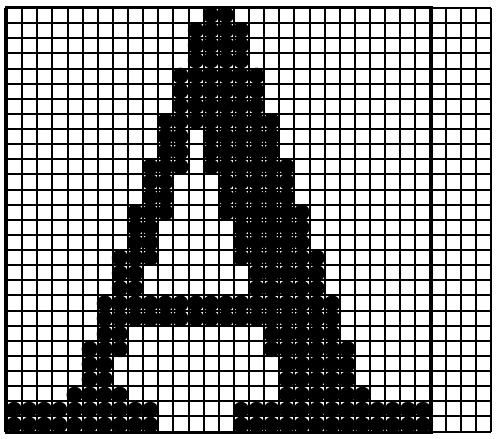
压缩代码的扩展名为 `.pk`，其前接设计尺寸的每英寸小点数。如果其为 300，输出是送到同分辨率的打印机上，那么就会得到正确的大小。如果创建的是 360 dpi 像素文件，还送到那个打印机上，结果就会比设计的大 1.2 倍，也就是说放大了这个倍数。

在有些计算机系统中，尤其是 PC 机上，像素文件的组织方式是不同的。这时它们都具有相同的扩展名，即 `.pxl` 或 `.pk`，具体与所用代码有关，但保存在不同的目录中，每个目录相应于一个放大倍数。因此目录名类似于 `360dpi` 的地方保存的就是所有放大倍数为 1.2 的像素文件 (假设打印机分辨率为 300 dpi) 。

©TEXGuru, August 16, 1999

§E.8.3 像素代码

用打印机输出每个字符, 都是用一系列的小点表示, 这些小点称为图形元素, 或者像素。像素的大小与打印机的分辨率有关。如果分辨率为 300 dpi , 那么一个像素的直径大约为 0.08 mm 。下面给出的是 10 pt 罗马字体中字母 A 的放大照。



左图每个小方框中的内容在计算机中都存贮为 0 或 1 。零表示白, 即空方框, 而 1 表示黑点。字母 A 由 28 行组成, 每行有 28 个点。顶部两行的编码为

0000000000000110000000000000

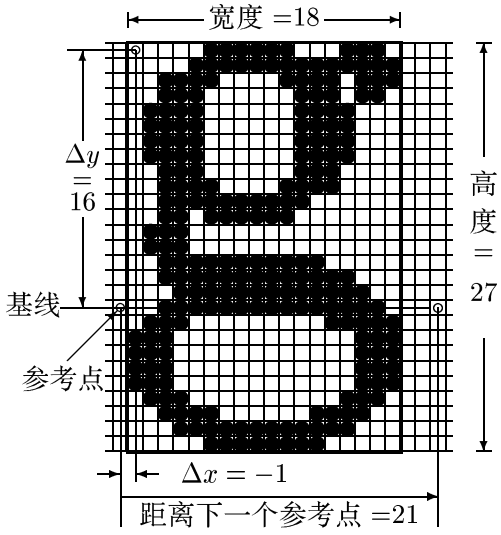
0000000000001111000000000000

而底行的编码为

1111111111000001111111111111

这就可以完整表示该字母。这样的二进制字符串不可能无穷长。计算机通常以 32 位 (即计算机中字的长度) 为一块。因此要把上面的行通过加 4 位零填补到 32 位。如果字符的宽度超过了 32 位, 那么每行就需要两组或多组 32 位。

按照计算机术语, 一组 8 位构成一个字节 (byte) 。因此 32 位的字中有 4 个字节。这里的字母 A 需要 28 个字或者说 128 个字节来存贮其式样。整个字体由 128 个字符组成, 因此完整的像素文件大概需要 14 000 个字节来表示。然而, 事情并不是到此为止, 每个字符还需要其它的信息, 从下面字母 g 的演示中可见一斑:



符号的像素式样可以包围在一个极小盒子中, 这个盒子恰好容纳了所有的黑点。对于字母 g , 这个盒子有 27 点高, 18 点宽。基线到顶行的点数用 Δy 表示, 等于 16 个像素。水平对齐用的参考点位于基线上, 距最左边列的点数用 Δx 表示; 负值意味着参考点在第一列的左边。这里 $\Delta x = -1$ 。最后, 这里同时要给出下一个参考点的位置, 对于 g , 其值为右边 21 个像素。

必须为每个符号在像素文件中保存这些值。极小盒子的高度和宽度, 以

及 Δy 和 Δx 的值，以像素为单位，每个需要 16 位，即半个字长。表示到下一个参考点距离的项要以 `tfm` 为单位，即其要与 `.tfm` 文件中的字符宽度一样。用四分之一的字长指定符号的像素式样在像素文件中的开头。

用 $128 \times 4 = 512$ 个字表示附加信息，构成一种字体中 128 个字符的一类目录表，其要位于像素文件的尾部。后接另外四个字，给出一般的字体信息，包括设计尺寸和放大倍数。像素文件的第一个和最后一个字都是 1001 的标志。因此一个像素文件是由像素式样外加 518 个字构成的附加信息。进一步的详情可能只有 `.dvi` 驱动程序的编写人员会感兴趣，可以在更专业的 TEX 文献中找到。

在输出时，如果对每个字符都要把其像素式样送到打印机上，那么打印速度会显著除低。实际上，激光打印机有自己的内存，因此要被传递的式样只需进行一次就可以了。这些被存贮起来的式样可以用该字符的代码号调用，这些代码号只有一个字节。它们就是列在 E.6 节字体布局中的代码号。

§E.8.4 压缩代码

从前一节内容可知，像素文件需要相当可观的存贮空间，尤其当处理的是放大字体时更加如此。例如，在放缩因子为 1440 时，相应于 `cminch` 的像素文件占据了总数为 705 376 的字节。

出于这个原因，Stanford 大学的 Tomas Rokicki 设计了一种压缩过程，用更紧致的形式保存像素信息。这个程序的基础就是点要么为黑，要么为白，而且一个点后接的更经常是同样点，而很少是相反的点，即像素的分布并不是随机的，而是按某种有序的方式出现的。

因此现在代码变成了一列数，其指定接续的有多少个同样的点。相邻点给出的相反颜色的连续点有多少个。当然，必须给出开始像素的颜色。对于前一节的字母 g 示例，开始颜色为白，代码的开头部分为

$$5/6/3/3/\underline{5}/14/2/4/4/1/3\ldots$$

该符号以五个白色像素开始，后接六个黑色的，再是三个白和三个黑。在第一行的最后那个像素以及第二行开头四个像素是白的，因此这里给出的是五个白的，其分散在两行上。在解码时这不会导致任何问题，因为极小盒子的宽度是已知的，所以 $\underline{5}$ 个像素是会正确地分散在两行上的。

数 0 从不会出现。对像素式样的分析可知在 90% 的情形中，同色的相邻像素数目要小于 14 。这也就是系列数中保存为 4 字节数，即 0–15 之间的值。0, 14 和 15 三个值有特殊意义。零用来表示一种颜色的像素点数大于 13 ，下面那个仍旧表示同色的像素。由于分析表明，在 37% 的情形中，一行后接的是同样模式的行，数 15 就表示这种情形。最后，数 14 表示有相同式样的行超过了两行，其后接的数表示到底有多少行。

即使是诸如极小盒子的高度和宽度，参考点的位置等附加信息也可以用

压缩方式存贮。这些详情只有在 .dvi 驱动程序的解压缩过程中才需要。

上述过程的效率是非常巨大的。 cminch.432pk 文件保存的是放大倍数为 1.44 的 cminch 字体，这时只需要 32 644 字节，只有等价像素文件大小的 5% 。这是一个极端例子，对 75 种标准字体有前四个放大幅度中，压缩形式所需的空间大约只有像素格式所需空间的 20% 。当放大倍数更大时，节省的比率就越大。

时至今日，已经很少能见到 pxl 代码了。其已经完全被 .pk 代码格式所代替。

§E.9 对 METAFONT的注释

METAFONT是一个设计和开发字符字体的程序，由 TEX 程序的发明者 Donald E. Knuth 编写。如果在你所用的计算机上有 METAFONT ，那么可以使用任何想要的 (更大或更小) 放大倍数，来重新生成已有的字体。

也可以用它创建诸如阿拉伯文或者希伯来文等全新的字体集合，虽然只有专家级人物才可能做到，因为其需要对 METAFONT的内部语言有相当深入的了解。本书并不是要讲解 METAFONT的，因此不会给出更多的信息。这里只是描写如何利用已存在的像素字体生成新的像素文件。

对 METAFONT程序的调用方式与所用的计算机操作系统有关。一旦启动了该程序，其就会在屏幕上显示出版本号以及上载的其它文件。然后就等待用户的反应，提示符为 ** 。反应必须是下面的形式：

\mode=localfont; mag=*nn*; input 文件

后接 ⟨回车⟩ 。这里 mag= 的值 *nn* 表示放大倍数；如果忽略该值，或者设为 1 ，那么用的就是设计尺寸。 input 的 文件 条目就是 E.8.1 节 75 种标准字体之一的基本名。做为 METAFONT基本实现的一部分，应该有 75 个同基本名，扩展名为 .mf 的文件。这些文件由具有 62 个参数的字符定义组成，这里不对其细节过多涉及。其它诸如相应于 Cyrillic 或特殊字符集的 .mf 文件也可以输入在 文件 所处的地方。

现在 METAFONT就会处理并生成选定字体在指定放大倍数下的像素文件，并在屏幕上显示字体名称和符号代码数。最后，程序显示出 * 并等待新的用户指令。反应 end⟨回车⟩ 就会结束该程序，把控制交还给操作系统。

METAFONT运行的结果是生成两个新文件，一个是字体的 .tfm 文件，另一个则具有相同的基本名，扩展名为 .xxxgf ，其中 gf 表示普通字体 (generic font) 。 *xxx* 反应了选定的放大倍数：如果放大倍数为 1 ，那么 *xxx* 就是以每英寸点数 (dpi) 表示的打印机分辨率。 METAFONT是从 \mode=localfont 中获取打印机信息及其分辨率的。在原始的版本中，这个分辨率应为 200 dpi ，但在任何安装中，应把 localfont 改成适应于所用打印机。

一个 .tfm 文件可以用于所有同基本名，放大倍数不同的字体。在有些地方， localfont 被进行了修改，只生成设计尺寸的 .tfm 文件，如同选定的是 mag=1 。

现在还需要把普通字体代码转化成驱动程序 (如上面描述的压缩代码) 能识别的形式。这是利用附属程序 gftopk 实现的。假设我们已经利用 METAFONT为 300 dpi 打印机生成了放大倍数为 1.44 的 cmr10 字体，这样我们就拥有了 cmr10.432gf 文件。

```
gftopk cmr10.432gf
```

的调用就进行了转化，生成压缩代码文件 cmr10.432pk 。

对于那些只能用原来像素代码的老驱动程序, 可以用附属程序 gftopxl 。还是上面的那个例子,

```
gftopxl cmr10.432gf
```

就生成 cmr10.1440pxl ，遵从 376 页表格中描述的命名约定，这里假定打印机分辨率为 300 dpi 。

通常 TEX的放大幅度为 $\sqrt{1.2}$ 或者 1.2 的整数次方。在命令行 mag= 中既可以输入十进制小数表示放大倍数，也可以用 magstep n 表示 1.2^n 。半幅度 $\sqrt{1.2}$ 可以输入为 magstep0.5 。

METAFONT允许对基本字体进行相当大的变体。无畏的用户可以尽情地对其进行操作。无论进行了怎样的操作，不要改动原来的 .mf 文件，要把新生成的字体起个别的名字。列在 374 页上的 75 种 TEX字体应保持为标准。

关于如何使用与开发 METAFONT程序的详情, 可以阅读 Knuth 的书 *The METAFONTbook* 。警告：玩耍 METAFONT会上瘾，并且耗时巨大。

为了适应于 TEX3.0 中多语言应用的发展和每种字体有 256 个字符，Donald E. Knuth 把 METAFONT扩展到了 2.0 版。现在他不再希望对这两个程序进行新改进了，所做的只是对已有错误进行修正。为了强调这一决定，从现在开始，他让 TEX的版本号收敛到 $\pi(3.14159\ldots)$ ， METAFONT的版本号收敛到 $e(2.71828\ldots)$ 。现在 TEX版本为 3.1415 ， METAFONT版本为 2.71 。这个决定的后果就是，如果有用户组织对这两个程序进行了大发展，那么就必须给它起个新名称，因为 Donald E. Knuth 保留对这些程序名称的版权。

©TEXGuru, August 16, 1999