



Tutorials and worked examples for simulation,
 curve fitting, statistical analysis, and plotting.
<http://www.simfit.org.uk>

This document explains the fonts available for PostScript plots, then describes the options provided by SIMFIT to display titles, legends, and plot labels containing superscripts, subscripts, and mathematical equations.

Standard fonts

Printers that support PostScript (i.e. most modern printers) have a basic set of 35 fonts and it can be safely assumed that graphics using these fonts will display in GSview/Ghostscript and print on all such printers. Of course there may be a wealth of other fonts available, but details for some that are widely used will be given. The Times and Helvetica fonts are well known, and the monospaced Courier family of typewriter fonts are sometimes convenient for tables.

Times-Roman

!"#\$%&'()*+,-./0123456789:;<=>?@
 ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
 abcdefghijklmnopqrstuvwxyz{|}~

Times-Bold

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Times-Italic

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Times-BoldItalic

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica

!"#\$%&'()*+,-./0123456789:;<=>?@
 ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
 abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-Bold

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-Oblique

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Helvetica-BoldOblique

!"#\$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`
abcdefghijklmnopqrstuvwxyz{|}~

The StandardEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	"	#	\$	%	&	'
\05x	()	*	+	,	-	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	@	A	B	C	D	E	F	G
\11x	H	I	J	K	L	M	N	O
\12x	P	Q	R	S	T	U	V	W
\13x	X	Y	Z	[\]	^	_
\14x	'	a	b	c	d	e	f	g
\15x	h	i	j	k	l	m	n	o
\16x	p	q	r	s	t	u	v	w
\17x	x	y	z	{		}	~	
\20x								
\21x								
\22x								
\23x								
\24x		ı	ç	£	/	¥	f	§
\25x	ı	'	“	«	<	>	fi	fl
\26x		—	†	‡	·		¶	•
\27x	,	„	”	»	...	‰		ı
\30x		`	^	^	~	-	˘	˙
\31x	˚		°	˘		˚	˛	˜
\32x	—							
\33x								
\34x		Æ		ª				
\35x	Ł	Ø	Œ	º				
\36x		æ				ı		
\37x	ı	ø	œ	ß				

The ISOLatin1Encoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	"	#	\$	%	&	'
\05x	()	*	+	,	-	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	@	A	B	C	D	E	F	G
\11x	H	I	J	K	L	M	N	O
\12x	P	Q	R	S	T	U	V	W
\13x	X	Y	Z	[\]	^	_
\14x	'	a	b	c	d	e	f	g
\15x	h	i	j	k	l	m	n	o
\16x	p	q	r	s	t	u	v	w
\17x	x	y	z	{		}	~	
\20x								
\21x								
\22x	ı	`	˘	ˆ	˜	-	˘	·
\23x	¨		°	´	˝	˝	˘	˘
\24x		ı	¢	£	¤	¥	¦	§
\25x	¨	©	ª	«	¬	-	®	-
\26x	°	±	²	³	´	µ	¶	·
\27x	¸	¹	º	»	¼	½	¾	¿
\30x	À	Á	Â	Ã	Ä	Å	Æ	Ç
\31x	È	É	Ê	Ë	Ì	Í	Î	Ï
\32x	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×
\33x	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
\34x	à	á	â	ã	ä	å	æ	ç
\35x	è	é	ê	ë	ì	í	î	ï
\36x	ð	ñ	ò	ó	ô	õ	ö	÷
\37x	ø	ù	ú	û	ü	ý	þ	ÿ

The SymbolEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x		!	∀	#	∃	%	&	ə
\05x	()	*	+	,	-	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	≅	A	B	X	Δ	E	Φ	Γ
\11x	H	I	∅	K	Λ	M	N	O
\12x	Π	Θ	P	Σ	T	Υ	ς	Ω
\13x	Ξ	Ψ	Z	[∴]	⊥	—
\14x	—	α	β	χ	δ	ε	φ	γ
\15x	η	ι	φ	κ	λ	μ	ν	ο
\16x	π	θ	ρ	σ	τ	υ	ϖ	ω
\17x	ξ	ψ	ζ	{		}	~	
\20x								
\21x								
\22x								
\23x								
\24x		Υ	'	≤	/	∞	<i>f</i>	♣
\25x	♦	♥	♠	↔	←	↑	→	↓
\26x	°	±	"	≥	×	∞	∂	•
\27x	÷	≠	≡	≈	...		—	⌋
\30x	⌘	ℑ	℔	∅	⊗	⊕	∅	∩
\31x	∪	⊃	⊇	∄	⊂	⊆	∈	∉
\32x	∠	∇	®	©	™	Π	√	·
\33x	¬	^	∨	↔	←	↑	⇒	↓
\34x	◇	⟨	®	©	™	Σ	∫	
\35x	⌈	⌊		⌊	⌊	{	⌊	
\36x		⟩	⌋	⌋		⌋	⌋	
\37x	⌋	⌋		⌋		⌋	⌋	

The ZapfDingbatsEncoding Vector

octal	0	1	2	3	4	5	6	7
\00x								
\01x								
\02x								
\03x								
\04x								
\05x								
\06x								
\07x								
\10x								
\11x								
\12x								
\13x								
\14x								
\15x								
\16x								
\17x				,	,	“	”	
\20x								
\21x								
\22x								
\23x								
\24x								
\25x					①	②	③	④
\26x	⑤	⑥	⑦	⑧	⑨	⑩	①	②
\27x	③	④	⑤	⑥	⑦	⑧	⑨	⑩
\30x	①	②	③	④	⑤	⑥	⑦	⑧
\31x	⑨	⑩	①	②	③	④	⑤	⑥
\32x	⑦	⑧	⑨	⑩	→	→	↔	↕
\33x	▲	→	▼	→	→	→	→	→
\34x	→	→	→	→	→	→	→	→
\35x	→	→	→	→	→	→	→	→
\36x	→	→	→	→	→	→	→	→
\37x	→	→	→	→	→	→	→	→

Simfit character display codes

Each character plotted by SIMFIT is actually a pair consisting of the actual letter entered from the keyboard or character selection table, and a code which has one of the following meanings.

- 0** Standard font
- 1** Standard font subscript
- 2** Standard font superscript
- 3** Maths/Greek
- 4** Maths/Greek subscript
- 5** Maths/Greek superscript
- 6** Bold Maths/Greek
- 7** ZapfDingbats (PostScript) Wingding (Windows)
- 8** ISOLatin1Encoding (PostScript), Standard (Windows, almost)
- 9** Special (PostScript) Wingding2 (Windows)
- A** Grave accent
- B** Acute accent
- C** Circumflex/Hat
- D** Tilde
- E** Macron/Bar/Overline
- F** Dieresis
- G** Maths/Greek-hat
- H** Maths/Greek-bar
- I** Bold maths/Greek-hat
- J** Bold Maths/Greek-bar
- K** Symbol font
- L** Bold Symbol font

You will need non-keyboard characters from the standard font for such characters as a double dagger (‡) or upside down question mark (¿), e.g. typing `\277` in a text string would generate the upside down question mark (¿) in the PostScript output. If you want to include a single backslash in a text string, use `\\`, and also cancel any unpaired parentheses using `\(` and `\)`. Try it in program **simplot** and it will then all make sense. The ISOLatin1Encoding vector is used for special characters, such as `\305` for Angstrom (Å), `\361` for n-tilde (ñ), or `\367` for the division sign (÷), and, apart from a few omissions, the standard Windows font is the same as the ISOLatin1Encoding.

*If you type four character octal codes as character strings for plotting non-keyboard characters, you do not have to worry about adjusting the character display codes, program **simplot** will make the necessary corrections. The only time you have to be careful about the length of character display code vectors is when editing in a text editor. If in doubt, just pad the character display code vector with question marks until it is the same length as the character string.*