

XMGrace

Fancy characters and stuff

In XMGrace it is possible to write Greek letters, do superscripts and subscripts and the like. This tex-file/PDF will hopefully keep a list of what I have learnt (starting from: <http://blog.louic.nl/?p=249>).

Also check (from Googling ‘xmgrace greek character list’):

<http://agendafisica.wordpress.com/2010/12/01/grace-typesetting-for-titles-legends-tick-marks/>

Greek Characters

For Greek characters (for example, sigma):

```
1 \f{Symbol}s\f{}
```

or

```
1 \x s\f{}
```

Note that you need a capital ‘S’ on Symbol! The `\f{}` returns the original font. For different characters after the `{Symbol}` part you get different output as described in the table below:

a	α	A	A
b	β	B	B
c	χ	C	X
d	δ	D	Δ
e	ϵ	E	E
f	ϕ	F	Φ
g	γ	G	Γ
h	η	H	H
i	ι	I	I
j	φ	J	ϑ
k	κ	K	K
l	λ	L	Λ
m	μ	M	M
n	ν	N	N
o	o	O	O
p	π	P	Π
q	θ	Q	Θ
r	ρ	R	P
s	σ	S	Σ
t	τ	T	T
u	υ	U	Y
v	ϖ	V	ς
w	ω	W	Ω
x	ξ	X	Ξ
y	ψ	Y	Ψ
z	ζ	Z	Z

Table 1: Different Greek characters available in XMGrace

Superscripts

For example x^2 is:

```
1 x\S2\N
```

Note that the capital ‘S’ makes it a superscript. The \N returns the text to the default state

Subscripts

For example a_b is:

```
1 a\sb\N
```

Note that the lower case ‘s’ makes it a subscript. The \N returns the text to the default state

Overlines

Overlines are created in the following way:

```
1 \oA\O
```

This would give: \overline{A} as an output

Rotations

To rotate the next set of elements by x degrees anticlockwise relative to the current orientation (note that the effect is cumulative):

```
1 \r{x} rotated font
```

to return to normal:

```
1 \r{-x}
```

Slanting

To slant (to the right) by a factor of x (slant back to the left by -x to cancel):

```
1 \l{x}
```

Vertical Shift

To vertically shift by x units (note that 1 appears to be roughly double line spacing):

```
1 \v{x}
```

Horizontal Shift

To horizontally shift by x units:

```
1 \h{x}
```

Underlines

To begin and end an underling, use:

```
1 \u asd\U
```

To get: asd

Italics

```
1 \q italics\Q NOT
```

Gives: “*italics* NOT”. Note that this is equivalent to slanting by 0.25 (e.g. \l{0.25})

Colours

1 `\R{colourname}`

Gives a colour of the type ‘colourname’. Suitable colournames include (note no capitalisation): red, blue, yellow, green, cyan, brown, grey, violet, magenta, orange, green4, indigo, maroon, turquoise.

Font Size

Increase or decrease font size using:

1 `\+ \-`

Where `\+` increases font size and `\-` decreases font size. Note that `\N` returns the text to the default state

Curly Font

To use a *curly font* try (note, if your system doesn’t have the URW Chancery L font, this will do nothing):

1 `\f{URWChanceryL–MediumItalic}`

to leave this font type use `\f{}`

Other Useful Characters

For interest’s sake: ‘`\x`’ puts you into symbol font (the same as ‘`\f{Symbol}`’) and ‘`\c`’ enters you into the upper 128 characters of a set while ‘`\C`’ removes you from it. Note to put yourself back in the default font just type ‘`\f{}`’.

<code>\x\c”\C</code>	$!$
<code>\x\c%\C</code>	∞
<code>\x\c^\C</code>	\Rightarrow
<code>\x\c.\C</code>	\rightarrow
<code>\x\c1\C</code>	\pm
<code>\x\c3\C</code>	\geq
<code>\x\c6\C</code>	∂
<code>\x\c9\C</code>	\neq
<code>\x\ce\C</code>	Σ
<code>\x\cr\C</code>	\int
<code>\x\cQ\C</code>	∇
<code>\x\cU\C</code>	Π
<code>\x\cO\C</code>	\notin
<code>\x\c;\C</code>	\approx
<code>\x\cB\C</code>	Real Part
<code>\x\cA\C</code>	Im Part
<code>\x\c#\C</code>	\leq
<code>\x\cN\C</code>	\in
<code>\x”\C</code>	\forall
<code>\c0\C</code>	\circ
<code>\x\ca\C</code>	\langle
<code>\x\cq\C</code>	\rangle
<code>\f{ZapfDingbats}=</code>	\dagger

Table 2: Useful characters available in XMGrace

Note, there may be issues with spaces turning into Euro symbols if you are still in the ‘\c’ environment.

Fractions

Not a nice thing, but by playing around with the horizontal and vertical placement of what you’ve got and adding in some underscores you can jerry rig a fraction:

```
1 \+ \v{0.7}1\v{-0.3}\h{-0.7}--\v{-1.1}\h{-0.7}3\N
```

This gives $\frac{1}{3}$. Note that for longer numbers you’ll have to adjust the number of underscores and horizontal placement.

PRINTING PDFs

It is possible to adjust the page size in ‘View’→‘Page Setup’ using the ‘Orientation’ and ‘Size’ options.

Once an appropriate size has been chosen (note: Letter will do!), go to ‘Plot’→‘Graph Appearance’ and change options in ‘Viewport’ (i.e. $Xmin$, $Xmax$, $Ymin$, $Ymax$) - I have used the values 0.08, 1.39, 0.09, 0.98 (respectively) as these remove whitespace effectively.

It is worth noting that having scaled up the graph size, the fonts seem a little small so it may be useful to increase text size (this would then require changing the ‘Viewport’ values again).

Export this as a postscript (.ps) file then *ps2pdf* it. Be aware that if Greek characters (or other characters perhaps) are included in the graph then *ps2pdf* may have trouble printing them so you need to add the following flags: `-dPDFSETTINGS=/printer -dAutoRotatePages=/All`

NOTE: it is not necessary to crop the graphs manually, one can also export them as .ps, then convert to pdf and use the *pdfcrop* command line function.

1 Graph-ception

To make a sub-graph (i.e. inset), first have the original graph you want to include the inset with. Then go: *Data* » *Import* » *ASCII* (as usual) to bring up the *Read Sets* menu. In the section *Read To Graph*, right click in the scroll box and select the option *Create new*, this now enables a second graph to be created and have data read into it. Then going to *Edit* » *Arrange Graphs* to bring up the menu that lets you adjust the *Page Offsets* of each graph to make it fit in the right place.

In order to edit a graph make sure that the correct graph is selected from *Edit* » *Arrange Graphs* (i.e. the corners have black squares on them).

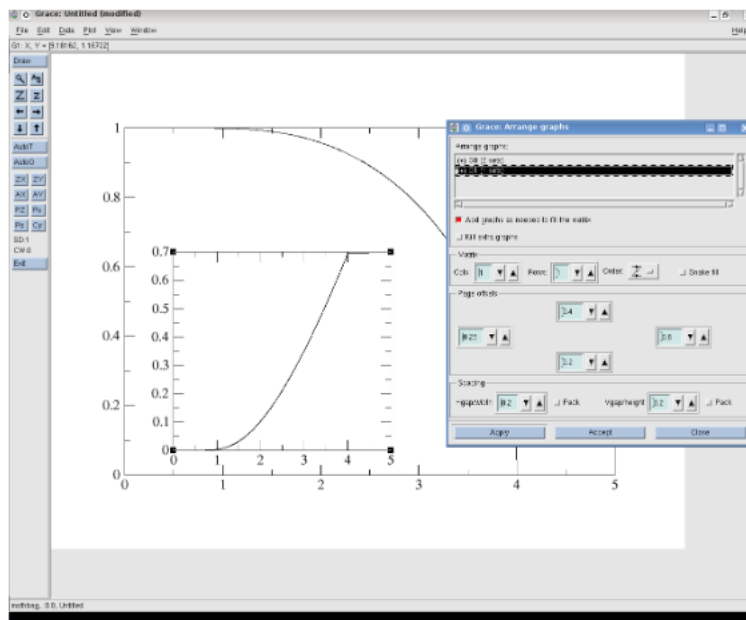


Figure 1: The *Arrange Graphs* menu with the *Page Offsets* options set to produce an inset