High Quality Labels on Included Graphics, using XY-pic

Ross Moore
Mathematics Department
Macquarie University
Sydney, Australia 2109
`ross@mpce.mq.edu.au`
URL: `http://www-math.mpce.mq.edu.au/~ross/`

Abstract

The XY-pic suite of graphics macros, through the `\xyimport` command, offers the ability to easily annotate and label imported graphics using TeX's full capabilities for handling mathematics and special fonts.

Introduction

Many modern software applications exist for constructing elegant graphics to present information of all kinds. This includes scientific data, maps, charts, tables and business graphics. Almost without exception, the handling of text for titles and labelling is rather limited. Typically just a single font may be used, superscript and subscripts are often not supported, and all but the simplest mathematical expressions can be constructed. Questions are continually asked, on Internet newsgroups, about how to include publication-quality typeset labels on the graphics produced using such programs.

A common solution to this problem is to first save the graphic in Encapsulated PostScript[1] (EPS) format, then use Adobe's Illustrator™, or other program, to render the `.eps` file, and edit it to add the annotations and labels. For many purposes this is effective, producing high-quality results. It is a strategy adopted with many scientific journals.

However this technique is not always sufficient (especially for mathematics) and has several drawbacks when the intention is to include the graphic within a TeX or LaTeX document, quite apart from the extra expense of a piece of commercial software. It can be quite difficult to achieve the following:

S1 use same or similar fonts in the imported graphic as in the surrounding typeset text, at compatible sizes and styles;

S2 ability to resize the graphic (to fit available space in the typeset document) while retaining font-compatibility as in S1;

S3 inclusion of properly typeset mathematics; e.g. for axis-labels on graphs.

Adjustment of font-sizes or style, and the overall size of the graphic, are decisions of style which can be quite independent of the main content of the graphic. Such decisions should be made late in the overall process of preparing a manuscript for typesetting. Thus to accommodate any changes, reediting within Illustrator™ is required, and perhaps even regeneration of the original graphic using the specialised software. This may be impractical or even impossible to achieve, once the manuscript has been submitted for publication.

The correct place for such style-decisions to be made is within the TeX (or LaTeX) source of the document within which the graphic is to appear. This can be realised effectively using XY-pic. Note that as a first step to achieve font-style compatibility, the original graphic should be created so that *it contains no explicit text or labelling*; for these will be added later during the TeX/LaTeX processing. For most graphs it is not even necessary to include the axes, for these also may be added using XY-pic, as the examples show. (Indeed with some programs there are distinct advantages in leaving out the axes and any frame. We comment further about this near the end of the article.)

The `\xyimport` command

To use the `\xyimport` command of XY-pic, first the xyimport feature must have been loaded. This is done by including import among the options given to LaTeX's `\usepackage` command when XY-pic is initially loaded, or by explicit use of `\xyoption`.

`\usepackage[...,import,...]{xy}`	(LaTeX 2_ε)
`\xyoption{import}`	useable with any format.

The `\xyimport` command is used within an XY-pic picture or diagram in one of the following ways:

[1] PostScript® is a registered Trademark of Adobe, Inc. (Adobe Systems Incorporated, 1990).

```
\begin{xy}                                using LaTeX
\xyimport(1,1){⟨graphics⟩}
... further XY-pic commands ...
\end{xy}

\xy\xyimport(1,1){⟨graphics⟩}         in any format
... further XY-pic commands ...
\endxy
```

Here ⟨*graphics*⟩ denotes the TEX/LATEX command to include a graphic within the document; e.g. LATEX's `\includegraphics[...]{...}` command, or use of `\epsbox[...]{...}` from the `epsf` package, or use of `\psfig{...}`. Actually ⟨*graphics*⟩ may be *any* TEX source that creates an `\hbox`. In the examples below we use `\xyimport{\emptybox}` with a definition:

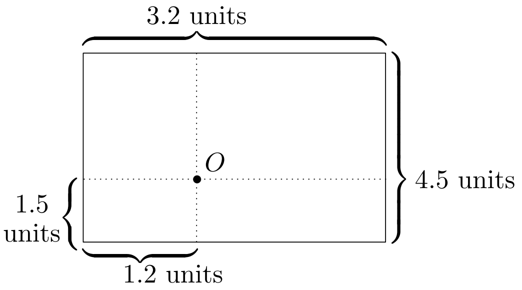
```
\providecommand{\emptybox}{\hbox to4cm{%
 \vrule height2.5cm width0cm\hfill}}
```

The `(1,1)` may be replaced by any pair of positive numbers. Furthermore a second pair of numbers may be optionally given:

```
\xyimport(3.2,4.5)(1.2,1.5){\emptybox}
```

The significance of each number is shown in figure 1. One sees that the first pair establishes the length of coordinate units, independently in horizontal and vertical directions. A second pair gives the offset, in these units, from the bottom-left corner to a point O, which becomes the origin for the coordinate system. When no second pair is given, O is located at the bottom-left corner.

Figure 1: Establishing an origin and coordinates.



With such a coordinate system, any location on the graphic (or even outside of it) can be specified as a pair of numbers (x,y). Using XY-pic kernel commands, any ⟨*object*⟩ (e.g. text-strings, mathematics or anything typeset using TEX) may be positioned at that location.

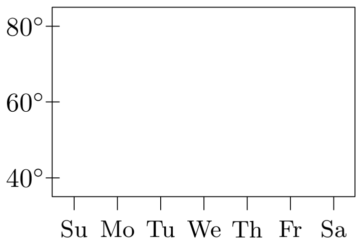
In XY-pic parlance, the `\xyimport` command "drops an ⟨*object*⟩", consisting of the ⟨*graphics*⟩, into a diagram. This specifies a ⟨*pos*⟩; i.e. a rectangle with a distinguished point, usually inside or on its boundary. This information alone is sufficient to drop further ⟨*object*⟩s into the diagram at locations determined using the edges of the ⟨*graphics*⟩ object. With the extra information provided by the number-pairs, the coordinate basis for the XY-pic "graphics-state" can be adapted to whatever is most appropriate for the logical meaning of the (visual) content of the ⟨*graphics*⟩.

frames, ticks and labels

As an example, suppose the graphic displays data collected for different days of the week. One wishes to add labels on the axis to mark each day. The following code produces the picture in figure 2.

```
$$\def\degrees{^{\circ}}\def\low{35}
\begin{xy}
\xyimport(7,50)(-.5,-\low){\emptybox}*\frm{-}
,(.5,40)*@{|}*+!R{40\degrees}
,(.5,60)*@{|}*+!R{60\degrees}
,(.5,80)*@{|}*+!R{80\degrees}
,0;/r2pc/**{} % set horiz direction
,(1,\low)*@_{|}*++!U\txt\small{Su}
,(2,\low)*@_{|}*++!U\txt\small{Mo}
,(3,\low)*@_{|}*++!U\txt\small{Tu}
,(4,\low)*@_{|}*++!U\txt\small{We}
,(5,\low)*@_{|}*++!U\txt\small{Th}
,(6,\low)*@_{|}*++!U\txt\small{Fr}
,(7,\low)*@_{|}*++!U\txt\small{Sa}
\end{xy}$$
```

Figure 2: Axis labels on a weekly chart.



Note the following features which are apparent in figure 2 and the preceding code.

- the origin of coordinates can be outside the area of the ⟨*graphics*⟩, by having negative offsets from the bottom-left corner.
- a solid frame is placed using `*\frm{-}`; many alternate styles of frame are possible; see the XY-pic Reference Manual (Rose, Kristoffer and Moore, Ross, 1997).
- tick-marks on the frame (or axes) may be positioned above or below (inside or outside) or through the edges. That strange combination `;p+/r2pc/**{}` is a technical device to switch the current direction to horizontal, so that the subsequent `*@_{|}` produce vertical tick-marks.

(The initial direction is vertical, so then `*@{|}`s come out as horizontal; i.e. rotated 90°.)

- text-labels are normally given a margin, e.g. as in `*++!U\txt` (add more `+`s for a wider margin), then positioned with an appropriate edge at the coordinate location. `!U` positions the U(pper) edge of the day-name abbreviations, while `!R` positions the R(ight) edge of the temperatures. Similarly L(eft), D(own), C(enter) and 2-letter combinations of these, are allowable. See the XY-pic Reference Manual (Rose, Kristoffer and Moore, Ross, 1997) for more details.

Also note how a coordinate may be specified from expanding a macro rather than typing explicit numbers. Indeed XY-pic kernel commands generally can be given as expansions of macros, provided `\drop` and `\POS` commands are used to ensure that appropriate parsers are activated. So if the repetition of data in the above code is of concern, then the following code produces exactly the same diagram as in figure 2.

```
$$\def\low{35}\def\off{.5}
\def\temp#1{\POS(\off,#1)*@{|}*+!R{#1^{\circ}}}
\def\wkday#1{\ifcase #1{}%
 \or Su\or Mo\or Tu\or We\or Th\or Fr\or Sa\fi}
\def\day#1{\POS
 (#1,\low)*@_{|}*+++!U\txt\small{\wkday{#1}}}
\begin{xy}
\xyimport(7,50)(-\off,-\low){\emptybox}*\frm{-}
\temp{40}\temp{60}\temp{80}\POS 0;/r2pc/**{}
\day1 \day2 \day3 \day4 \day5 \day6 \day7
\end{xy}$$
```

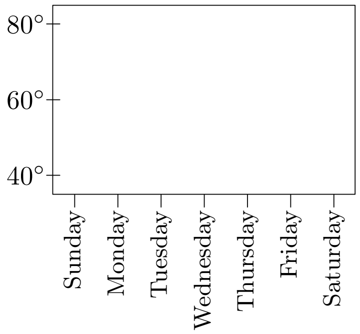
Rotated labels. To rotate text in TEX or LATEX requires some POSTSCRIPT® trickery. It can be done using LATEX's graphics package, provided an appropriate dvi-driver (usually dvips) is loaded. Alternatively XY-pic has a rotate feature, which similarly requires support. Including an `\xyoption` command, as in the first of these lines, causes the required driver-code to be loaded and activated.

```
\xyoption{dvips,rotate}
\usepackage[dvips,frame,rotate,import]{xy}
```

Alternatively multiple features can be loaded with a single `\usepackage` command in LATEX. For all the diagrams in this article the lower line above is sufficient to request the necessary options.

```
$$\def\low{35}\def\off{.5}
\def\temp#1{\POS(\off,#1)*@{|}*+!R{#1^{\circ}}}
\def\weekday#1{\ifcase#1\or Sun\or Mon\or Tues%
 \or Wednes\or Thurs\or Fri\or Satur\fi day}
\def\day#1{\POS (#1,\low)*@_{|}
 *[@^]++!U\txt\small{\weekday{#1}}}
\begin{xy}
\xyimport(7,50)(-\off,-\low){\emptybox}*\frm{-}
\temp{40}\temp{60}\temp{80}\POS 0;/r2pc/**{}
\day1 \day2 \day3 \day4 \day5 \day6 \day7
\end{xy}$$
```

Figure 3: ...with rotated labels.



The `[@^]` ⟨*modifier*⟩ rotates the text through 90° counter-clockwise; so that that it stands vertical, pointing upwards. Alternatively `[left]` achieves the same effect. Other such modifiers are `[right]` and `[flip]` (180°). After the rotation, adding first the margin and then aligning (`++!U`) gets the text positioned exactly as desired.

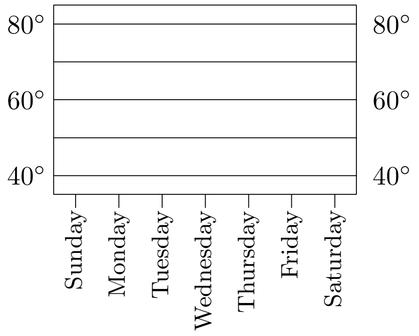
Axes and grid-lines

Grid-lines, stretching across the graph, from one side to the other or from top to bottom, are handled using XY-pic ⟨*connection*⟩s Given the coordinatisation, these are particularly easy. For example, a gridline at 60° could be placed into figure 3 with the code:

```
(.5,60);(7.5,60)**@{-}
```

Earlier code need change only slightly, using a macro `\Grline` to place grid-lines, labelled either side and without ticks. Another macro `\grline` places just lines. (The `\day` macro is as in figure 3.)

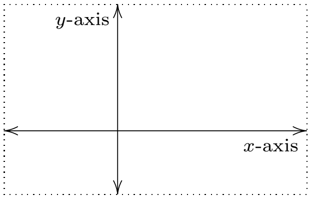
Figure 4: ...with horizontal grid-lines.



```
$$\def\low{35}\def\off{.5}
\def\grline#1{\POS(\off,#1);(7.5,#1)**@{-}}
\def\Grline#1{\POS(\off,#1);p*+!R{#1^{\circ}},
  (7.5,#1)**@{-},*++!L{#1^{\circ}}}
\begin{xy}
\xyimport(7,50)(-\off,-\low){\emptybox}*\frm{-}
\Grline{40}\Grline{60}\Grline{80}
\grline{50}\grline{70}\POS 0;/r2pc/**{}
\day1 \day2 \day3 \day4 \day5 \day6 \day7
\end{xy}$$
```

Coordinate Axes. The axes could be treated as a special form of grid-line. However, as usually they are required to pass through the origin of coordinates, then in Xy-pic they can be specified logically, without reference to coordinates at all.

Figure 5: Axes with arrows and labels.



```
$$\begin{xy}
\xyimport(3.2,4.5)(1.2,1.5){\emptybox}="a"
,*\frm{.} %  dotted outline
,"a"+L;"a"+R,**{}
 ?(0)*@{<};?(1)*@{>}**@{-} % arrow-tips
,?(1)*+!UR\txt\scriptsize{$x$-axis},
,"a"+D;"a"+U,**{}
 ?(0)*@{<};?(1)*@{>}**@{-} % arrow-tips
,?(1)*+!UR\txt\scriptsize{$y$-axis}
\end{xy}$$
```

After naming the ⟨*graphics*⟩ (using ="a") the idea is to set an empty ⟨*connection*⟩ (**{}) between the left ("a"+L) and right ("a"+R) edges. This establishes the correct direction for dropping the arrow-tips at either end (?(0)*@{<} and ?(1)*@{>}). Then put a solid ⟨*connection*⟩, using **@{-}. Labels are positioned along each axis using the parametrization provided by this latter ⟨*connection*⟩. To shift the arrow-tips just outside the area of the ⟨*graphics*⟩, simply replace the arrow-tips lines with these:

```
?(0)+/l1\jot/*@{<};?(1)+/r1\jot/*@{>}**@{-}
?(0)+/d1\jot/*@{<};?(1)+/u1\jot/*@{>}**@{-}
```

Figure 6: ...with arrow-heads further out.

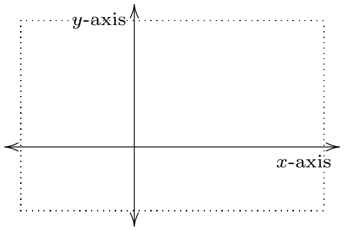
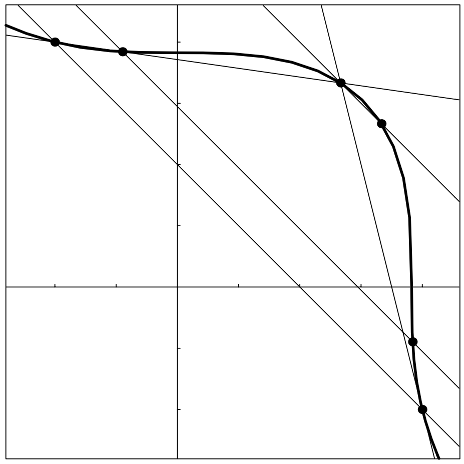
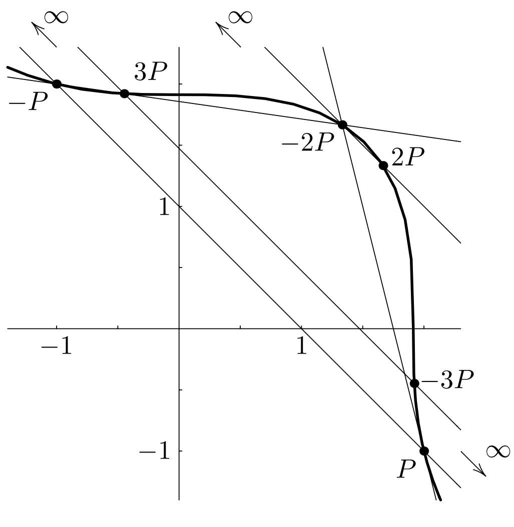


Figure 7: Original .eps graphic, with added frame...



framed contents of file:import1.eps

Figure 8: ...with mathematical embellishments.



Rational points on the elliptic curve: $x^3 + y^3 = 7$

Real-world example

The graphic in figure 7 was created using the *Mathematica* (Wolfram, 1993) program and appears in the XY-pic Reference Manual (Rose, Kristoffer and Moore, Ross, 1997). It is a variation of a diagram prepared for the book "Lectures on Fermat's Last Theorem"(van der Poorten, 1996)[2], for which I was TEX-nical editor. Many of its diagrams were typeset using `\xyimport`, as discussed here.

```
\xy\xyimport(3.7,3.7)(1.4,1.4){\ellipA}
,!D+<2pc,-1pc>*+!U\txt{Rational points
 on the elliptic curve: $x^3+y^3=7$}
,(1,0)*+!U{1},(-1,0)*+!U{-1}
,(0,1)*+!R{1},(0,-1)*+!R{-1}
,(2,-1)*+!RU{P},(-1,2)*+!RU{-P}
,(1.3333,1.6667)*+!UR{-2P}
,(1.6667,1.3333)*!DL{\;2P}
,(-.5,1.9)*++!DL{3P},(1.9,-.5)*!DL{\;-3P}
,(-1,2.3)*+++!D{\infty}*=0{},{\ar+(-.2,.2)}
,(.5,2.3)*+++!D{\infty}*=0{},{\ar+(-.2,.2)}
,(2.3,-1)*+++!L{\infty}*=0{},{\ar+(.2,-.2)}
\endxy
```

Warning. The appropriate numbers to use for the required argument to `\xyimport` can usually be read directly from the graphic being imported. Another way is to refer to the actual code which was used to generate the graphic, from its "plot range" say. However some software packages add an extra margin to this range, in particular when there are axes or a frame at the edges. Thus it can be necessary to estimate (read 'guess') the best numbers to use, when extreme accuracy is required in the placement of labels. A bit of 'trial-and-error' can be done from TEX or LATEX, without any need to regenerate the graphic.

Code for Figure 1. Here is the XY-pic code that was used to typeset figure 1. Notice how the edges and corners of the imported ⟨*graphics*⟩ are located logically, and saved (e.g. `"a"+R="aR"`) for repeated reuse. The extended braces are placed as a special type of "moustache" frame, on narrow rectangles derived from these corners and edges.

```
\xy\xyimport(3.2,4.5)(1.2,1.5){\emptybox}
 ="a",*\frm{-},*=0@{*},*+!DL{O}
,"a"+L="aL";"a"+R="aR"**@{.}
,"a"+D="aD";"a"+U="aU"**@{.}
,"a"+RD="aRD","a"+RU="aRU"
,"a"+LU="aLU","a"+LD="aLD"
,"aR"+/r5pt/."aRU"."aRD"*\frm{\}}
 ,*++!L\txt{4.5 units}
,"aU"+/u5pt/."aRU"."aLU"*\frm{^\}}
 ,*++!D\txt{3.2 units}
,"aL"+/l5pt/."aLD"!C*\frm{\{}
 ,*++!U(.2)!R\txt{1.5\\units}
,"aD"+/d5pt/."aLD"!C*\frm{_\}}
 ,*++!L(.5)!U\txt{1.2 units}
\endxy
```

[2] This book was awarded the prize of "Outstanding Professional/Scholarly Title in Mathematics, for 1996" by the Association of American Publishers.

References

Adobe Systems Incorporated. *PostScript Language Reference Manual.* Addison–Wesley, second edition, 1990.

Rose, Kristoffer and Moore, Ross. *XY-pic Reference Manual.* DIKU, University of Copenhagen, Universitetsparken 1, DK–2100 København Ø, 3.5 edition, 1997.

van der Poorten, Alf. *Lectures on Fermat's Last Theorem.* Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley Interscience, 1996.

Wolfram, Stephen. *Mathematica: A System for Doing Mathematics by Computer.* Addison–Wesley, 2nd edition, 1993.