Instructions for using the macro package `rlepsf.tex`

Colin Rourke

Mathematics Institute
University of Warwick
Coventry, CV4 7AL, UK

Email: `cpr@maths.warwick.ac.uk`
www: `http://www.maths.warwick.ac.uk/~cpr/`

Abstract

This set of instructions is a cut-down version of [1]. The following topics are covered here:

(1) Including `.eps` files in a `tex` or `latex` document with the macro package `epsf.tex`

(2) Relabelling `.eps` files in a `tex` or `latex` document using the macro package `rlepsf.tex`

(3) Adding labels using the package `labelfig.tex`

AMS Classification numbers Primary: 00-01, 00-02, 68-00, 68-01

Secondary: 51-00, 51-01, 55-00, 55-01, 57-00, 57-01

Keywords: Graphics, `tex`, `latex`, eps files, labelling, relabelling

Introduction

The macro package `rlespsf.tex` is an extension of the Rockiki–Knuth package `epsf.tex` [RoKn]. It allows the postscript labels in an `.eps` file to be replaced by `tex` labels. This means that you can draw your figure in any drawing program which is capable of `eps` output (or even draw the figure "by hand") using dummy postscript labels, which are then replaced by the correct `tex` labels.

Section 1 gives a brief introduction to `epsf.tex`; section 2 gives instructions for using `rlepsf.tex` and section 3 is an overview of the Siebenmann–Séroul package `labelfig.tex` which allows `tex` labels to be superimposed on an existing `.eps` file and in particular may be used in conjunction with `rlepsf.tex`.

Note The package `rlepsf.tex` can only be used in systems employing the Rokicki–Knuth `dvips` converter. (This is the standard converter normally installed with Unix or IbmPC `tex` instalations.)

1 Including .eps files in a tex document

This section is a brief introduction to the macro package `epsf.tex` [3]. Complete detailed instructions are given at the start of [3], my purpose here is merely to publicise this package and particularly to show how easy it is to use.[1]

Loading `epsf.tex`

You need the file `epsf.tex` and the line

```
\input epsf
```

somewhere in your file near the beginning.

(In `latex` you can alternatively load `epsf` by adding it to your `\usepackage` command; you need to make a copy of `epsf.tex` renamed as `epsf.sty` to do this. Note that this is *instead* of inserting the line `\input epsf`.)

Both files `epsf.tex` and `epsf.sty` are usually installed with `tex` installations.

Using `epsf.tex`

If your figure is the `.eps` file called say `spheres.eps` then to insert the figure in your file you merely type

```
\epsfbox{spheres.eps}
```

which will paste the figure into your file at the current point. To display it you would type for example:

```
\centerline{\epsfbox{spheres.eps}}
```

These instructions will insert the figure at its natural size. But you can also rescale it to any desired size very easily. The line

```
\centerline{\epsfxsize 4.4truein\epsfbox{spheres.eps}}
```

will display the figure but with width scaled to 4.4 inches (and height scaled appropriately to preserve the shape of the figure). A similar command `\epsfysize` will scale using height as the measure. Figure 1 was inserted in this file by using this command.

[1] The `epsf.tex` package can only be used with the Rokicki–Knuth `dvips` converter normally installed with Unix or IbmPC `tex` instalations.

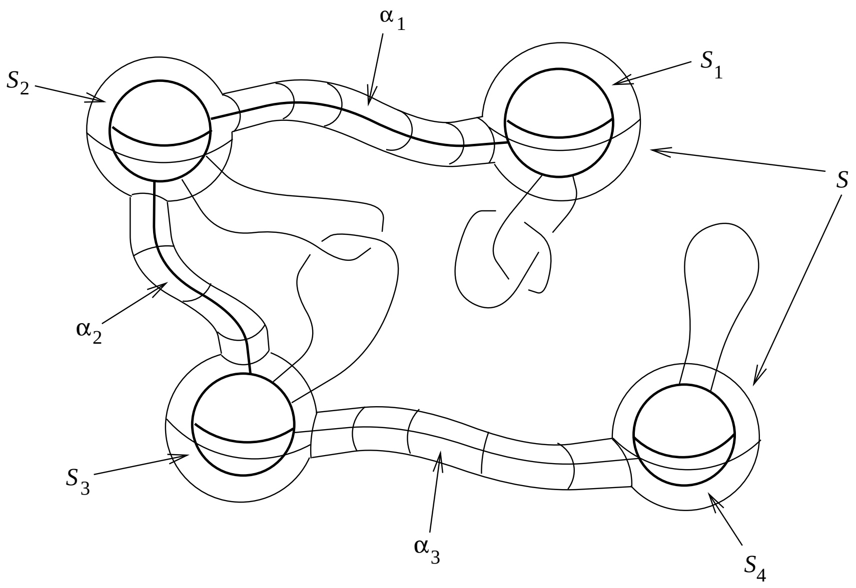


Figure 1: from [4]

2 Relabelling `.eps` files using `rlepsf.tex`

The macro package `rlespsf.tex` is an extension of `epsf.tex`. It allows the postscript labels in the `.eps` file to be replaced by `tex` labels. This means that you can draw your figure in any drawing program which is capable of `eps` output (or even draw the figure "by hand") using dummy postscript labels, which are then replaced by the correct `tex` labels.

Loading `rlepsf.tex`

You need both the files `epsf.tex` and `rlepsf.tex` and the line

```
\input rlepsf
```

somewhere in your file near the beginning. Do *not* also type `\input epsf` (if you already have this line then edit it to `\input rlepsf`). This is because `rlepsf.tex` loads `epsf.tex` automatically so it should not be loaded separately.

(In `latex` you can alternatively load `rlepsf` by adding it to your `\usepackage` command—you need a copy of `rlesps.tex` renamed to `rlepsf.sty`, and you also need `epsf.tex` for this. Note that this is *instead* of inserting the line `\input rlepsf`. Also note that the package `rlepsf` includes the package `epsf`, which should not be loaded separately.)

Using `rlepsf.tex`

Draw your figure in any drawing program and use "dummy" labels, ie choose labels which are simple strings but which remind you of the correct label; for example you could type `babcq2` to stand for the label $\bar babcq^2$. Now export your figure as an encapsulated postscript (`eps`) file. Place your `eps` file inside a `\relabelbox ... \endrelabelbox`. Insert the `eps` file exactly as usual with `epsf.tex` as explained in the last section. Type your relabelling instructions after the `\epsfbox{..}` command. A typical relabelling instruction (for the dummy label suggested above) is:

```
\relabel {babcq2}{$\bar babcq^2$}
```

There are three (re)labelling macros:

`\relabel` {pslabel}{TeX label}
`\adjustrelabel` <dx , dy> {pslabel}{TeX label}
`\extralabel` <x , y> {extra label}

`\relabel` uses the position of the dummy ps label for the new `tex` label (which can be any `tex` output). The new label is positioned with same reference point (left-hand end of baseline) as the dummy ps label. Usually this means that the new label position is perfect, but if it needs adjusting then `\adjustrelabel` can be used to adjust the position by (dx, dy). `\extralabel` allows extra `tex` labels to be inserted. The positions x, y are measured from the bottom right-hand corner of the diagram (so x is typically negative)! The re/extra-labelling instructions can be given in any order and any `tex` dimensions can be used for x, y, dx, dy.

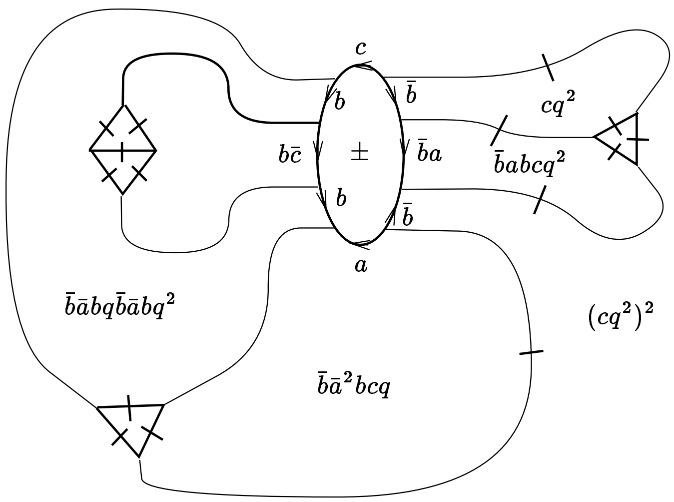
Preview the diagram (with ghostview, gsview or some similar ps previewer) after inserting the `\relabel` instructions but before inserting the `\extralabel` instructions. Decide if the figure needs scaling and if so choose the scaled width or height. Decide which labels need adjusting (if any) and measure the positions of any extra labels which are needed from the bottom right. Add the appropriate scaling command to the `\epsfbox` command as explained in the last section. Edit the `\relabel` instructions which need adjusting into `\adjustrelabel` instructions inserting appropriate <dx , dy> and add the `\extralabel` instructions.

Preview again and make any final adjustments to the various variables. This all sounds complicated, but it is in fact very straightforward and the amount of adjustment is usually minimal.

The result is a box with the same dimensions as the original epsfbox but with new `tex` labels. This can then be inserted in `$$ ... $$` or `\centerline{ ... }` for display, or combined with other boxes to form a larger display.

Here is a specific example. This is the code for relabelling figure 2, which was drawn using `xfig` and exported as an `eps` file.

```
\centerline{\relabelbox\small
\epsfxsize 3.5truein \epsfbox{KDiag.eps}
\relabel {c}{$c$}
\adjustrelabel <-2pt, 0pt> {a}{$a$}
\relabel {b}{$b$}
\adjustrelabel <-1pt, 0pt> {b1}{$b$}
\adjustrelabel <-1pt, -1pt> {bb}{$\bar b$}
\relabel {bb1}{$\bar b$}
\relabel {bc}{$b\bar c$}
\relabel {ba}{$\bar ba$}
\relabel {cq2}{$cq^2$}
\relabel {babcq2}{$\bar babcq^2$}
\relabel{(cq2)2}{$(cq^2)^2$}
\relabel {ba2bcq}{$\bar b\bar a^2bcq$}
\relabel {babqbabq2}{$\bar b\bar abq\bar b\bar abq^2$}
\relabel {where q=babc etc}{where $q=\bar b\bar abc$ and
unlabelled regions carry $q^3$, $a^2$ or $c^3$}
\extralabel <-1.7truein, 2.15truein> {$\pm$}
\endrelabelbox}
```



where $q = \bar b\bar abc$ and unlabelled regions carry q^3, a^2 or c^3

Figure 2: from [5]

Remarks (1) The `\relabelbox ... \endrelabelbox` forms a `tex` group; thus the instruction `\small` which sets the new labels in smaller type remains local to this group and does not affect the subsequent text.

(2) If the `eps` file has two or more labels the same, then only one will be replaced. Therefore it is necessary to use different dummy labels even if you wish the final labels to be the same. In the above example the dummy labels `b` and `b1` were both replaced by b and the dummies `bb` and `bb1` were both replaced by $\bar{b}$.

(3) The macros are sensitive to spaces. Be careful to leave a space after the $<dx\,, dy>$ adjustment but not between the {pslabel} and the {TeX label}.

(4) All the existing ps labels are deleted from the `eps` digram, whether or not instructions for relabelling are given. Thus you can use the macros to relabel an existing diagram by choosing the appropriate subset of labels to replace. However if you want to replace two or more ps labels, which are the same, then you have to use `\extralabel` to replace the one(s) which do not get relabelled at the first attempt. (Alternatively if you know how to find the labels in the `eps` file—look for strings enclosed in round brackets—then you can edit the `eps` file to make the repeated labels different.)

(5) The output will not view correctly on a dvi viewer which will typically show the new labels superimposed at the bottom right-hand corner. However those produced by `\extralabel` will be shown correctly positioned.

Health warning The package `rlepsf.tex` is still under development. The present version is a "beta test" version. It has been thoroughly tested with `xfig` and Adobe Illusrator output and also with 'hand-drawn' figures. However before committing any time to using the package, test it with a simple test figure to check that it will work properly with your chosen drawing program.

3 Relabelling `.eps` files using `labelfig.tex`

The macro package `labelfig.tex` [6] allows `tex` labels to be superimposed onto any picture, or in fact onto any box. Essentially this is a sophisticated version of the `\extralabel` facility in `rlepsf.tex` and indeed you can combine the two methods—replace the convenient ps labels by `tex` labels using `rlepsf.tex`—then add extra labels using `labelfig.tex`, see the example given below. The sophistication comes from the fact that the new labels attached by `labelfig.tex` are given using "scalable" coordinates, ie coordinates which are fixed to the picture and scale with it; moreover you can superimpose a temporary grid over the picture in order to read off the coordinates of the new label positions.

There is excellent documentation in the file `labelfig.tex` both as comments near the beginning and as an appended document; for full details on using the package, see this documentation. Here we shall just give an overview together with an example of use.

Loading `labelfig.tex`

You need the file `labelfig.tex` and the line

```
\input labelfig
```

somewhere in your file near the beginning.

Using `labelfig.tex`

Place your figure in an `\AffixLabels{ ... }` command preceeded by a `\ShowGrid` command, eg:

```
\ShowGrid
\AffixLabels{\epsfbox{figure.eps}}
```

Now preview the figure. The figure will be shown with a 10×10 grid superimposed over it. Read off the coordinates of the positions where you want the labels and then construct a label list of the form:

```
\SetLabels
(x1*y1) first label \\
(x2*y2) second label \\
...
\endSetLabels
```

which should come before the `\Showgrid`. You can prefix the coordinates with extra orientation commands (such as `\L`) which cause `labelfig.tex` to attach the new labels to the given coordinate by particular points. For example `\L` means: attach by left-hand end of baseline. (The default is centre of baseline.) For a complete list of these prefix commands see [6]; see also the worked example below. Preview again. When you are happy with the new labels, comment out the `\ShowGrid` command to remove the grid.

Here is a worked example. I used both `rlepsf.tex` and `labelfig.tex` to relabel an existing diagram (from [2]) with `tex` labels. The diagram with grid attached is shown in figure 3. The commands for displaying this figure were:

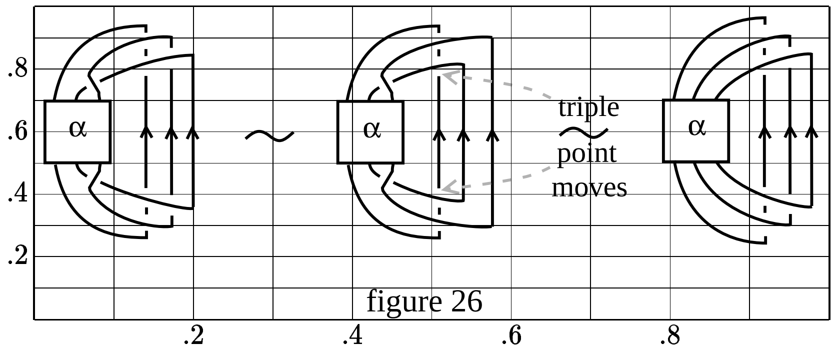


Figure 3: from [2] (drawn with Adobe Illustrator)

```
\centerline{\ShowGrid
\AffixLabels{\epsfbox{Closure.eps}}}
```

After previewing figure 3, I decided to replace the labels "triple", "point", "moves" and "figure 26" using `rlepsf.tex`, which is needed in any case in order to kill the postscript labels, and use `labelfig.tex` to replace the three α's. I read off the coordinates for the centre of the three α's as $(.05, .6)$, $(.42, .6)$, $(.83, .6)$ and typed the `\SetLabel` instructions. I previewed the figure a second time, made small adjustments to two of the `\relabels` and commented out the `\Showgrid`. The result is shown in figure 4, and the full set of instructions are as follows:

```
\centerline{%
\small
\SetLabels
\E(.05*.6) $\alpha$\\
\E(.42*.6) $\alpha$\\
\E(.83*.6) $\alpha$\\
\endSetLabels
%\ShowGrid
\AffixLabels{\relabelbox
\epsfbox{Closure.eps}
\relabel{triple}{triple}
\relabel{point}{point}
\adjustrelabel <2pt, 0pt> {moves}{moves}
\adjustrelabel <-0.5in,0pt> {figure 26}{Figure \RClosure:
figure \Closure\ after relabelling}
\endrelabelbox}}
```

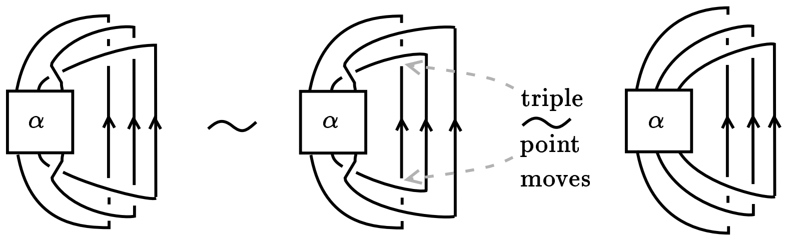


Figure 4: figure 3 after relabelling

Notes

(1) The `\E` commands prefixed to the coordinates for the α's are orientation commands which cause `labelfig.tex` to attach these labels by their centres to the point with given coordinates ("E" is for 'equator').

(2) It is not recommended to attach the caption to the figure as part of the figure. This was merely done in the example because the caption was part of the original figure and, if nothing had been substituted, there would have been an unsightly blank line at the bottom of the figure.

(3) The `\SetLabels .... \endSetLabels` pair forms a `tex` group and I could have typed `\small` after `\SetLabels` to set the new labels for `labelfig.tex` in the correct point size without affecting any other part of the text. However, by placing the `\small` command just after the `\centerline{`, both sets of new labels were set in the correct point size.

References

[1] **Geometry and Topology**, *Further topics for authors*, Geometry and Topology, `http://www.maths.warwick.ac.uk/gt/info/samples/topics.ps`

[2] **Sofia Lambropoulou and Colin Rourke**, *Markov's theorem in 3–manifolds*, Topology and its Applications (to appear) (available from my home page)

[3] **Tomas Rokicki (and Donald Knuth)**, *The macro package* `epsf.tex`, CTAN

[4] **Colin Rourke**, *Algorithms to disprove the Poincaré conjecture*, Talk given at the Gökova Topology Conference, May 1994 (available from my home page)

[5] **Colin Rourke**, *Dunce hats and the Kervaire conjecture*, Papers presented to EC Zeeman, Warwick (1988) 221–230 (available from my home page)

[6] **Ray Séroul and Larry Siebenmann**, *The macro package* `labelfig.tex`, CTAN