`pst-eps`:
Export of `PSTricks` environments

Herbert Voß

November 5, 2006

Abstract

It is relatively easy to save single `PSTricks` graphics as PostScript files. Important is that one

- puts a frame using `\fbox` around the `PSTricks` object,
- sets the line color to `white`,
- sets `\fboxsep` to `0pt` to avoid getting any additional space,
- chooses `\pagestyle{empty}`, to suppress the page number.

Contents

1 introduction 2

2 `\TeXtoEPS` 3

3 `\PSTtoEPS` 3

4 Parameters 4

1 introduction

Creating a `EPS` file from the dvi output is possible with

```
dvips spirale.dvi -E -o spirale.eps
```

which has the correct bounding box (for figures 1 `%%BoundingBox: 148 456 364 668`) on one hand and on the other can be included as normal a graphic in the document without problems thereafter. Figure 1 shows a graphic created this way and listing 1 the according source code.

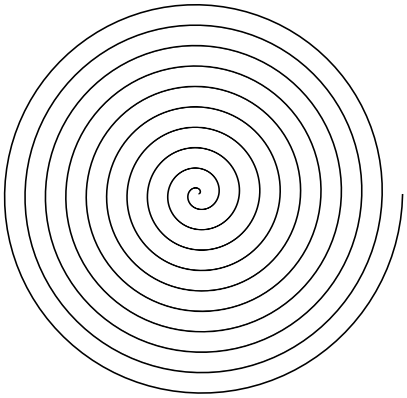


Figure 1: With the “`-E`”- option created `EPS` file

Listing 1: Source code for figure 1

```
\documentclass{article}
\usepackage{pstricks}% automatically loads xcolor
\usepackage{pst-plot}
\pagestyle{empty}
\begin{document}

\color{white}% fbox invisible
\fboxsep=0pt
\fbox{%
  \begin{pspicture}(-4,-4)(4,4)
  \parametricplot[plotpoints=1000]{0}{3600}{t dup cos 1000 div mul t dup sin 1000 div mul}
\end{pspicture}
}
\end{document}
```

With this method, one is forced to work with `\fbox`, since `dvips` is unable to determine a correct bounding box otherwise, because `dvips` does not regard graphical elements as boundaries. As an example for this, simply convert the above example without using `\fbox`. Since `\fbox` as a text element represents a clear boundary on text layer, **`dvips`** has no problems to definitely determine the bounding box. For converting single graphics this method is surely very efficient, but very time-consuming for a larger number. This is where the package **`pst-eps`** comes in, which tries to automate this process.

2 \TeXtoEPS

This macro has the task of rendering the trick with `\fbox` shown above superfluous, and therefore give **`dvips`** a possiblity to correctly determine the bounding box.

```
\TeXtoEPS%         TeX
 ...
\endTeXtoEPS
\begin{TeXtoEPS}%  LaTeX
 ...
\end{TeXtoEPS}
```

The same example as in listing 1 is picked up again, yielding the source code in listing 2.

Listing 2: Alternative source code to figure 1

```
\documentclass{article}
\usepackage{pst-eps}
\usepackage{pst-plot}
\pagestyle{empty}
\begin{document}

\begin{TeXtoEPS}
  \begin{pspicture}(-3.7,-3.7)(3.7,3.7)
    \parametricplot[plotpoints=1000]{0}{3600}{t dup cos 1000 div mul t
      dup sin 1000 div mul}
  \end{pspicture}
\end{TeXtoEPS}

\end{document}
```

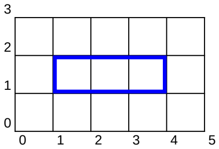
Again the `DVI` file is converted with **`dvips`** as described above, whereas this time a correct bounding box is yielded: `%%BoundingBox: 71 509 286 721`, which differs only in absolute, but not in relative values from the values given above.

3 \PSTtoEPS

With `PSTtoEPS` the `PSTricks` environment can be saved in an external file without detours.

`\PSTtoEPS[<parameters>]{<filename>}{<graphic object>}`

With this macro the problem of the bounding box not being determined correctly arises again. It can be specified with according parameters (table 2) The file is created immediately, so that it can be read direcly afterwards as `EPS` file, as in the following example.



```
\psset{checkfile=true}
\PSTtoEPS[bbllx=-0.5,bblly=-0.5,bburx=5.3,bbury=3.4,
  checkfile,headers=all,makeeps=all*]{frame.eps}{%
  \psgrid[subgriddiv=0](5,3)
  \psframe[linecolor=blue,linewidth=0.1](1,1)(4,2)%
}
\includegraphics[scale=0.5]{frame}
```

4 Parameters

Table 2 shows a compilation of all special parameters valid for `pst-eps`.

The parameters shall not be discussed in detail here, since the package `pst-eps` can be substituted by other possiblities meanwhile.

A practical use of `pst-eps` arises, when the calculation of single objects requires intense processor time, for instance three dimensional objects, like cylinders or spheres. Instead of conducting those calculation at every compile of the document again, one could create the graphic as `EPS` file in the first place and only read it in consequent LaTeX runs.

References

[1] Denis Girou. Présentation de PSTricks. *Cahier GUTenberg*, 16:21–70, April 1994.

[2] Michel Goosens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Graphics Companion*. Addison-Wesley Publishing Company, Reading, Mass., 1997.

[3] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.

[4] Herbert Voß. *`PSTricks` – Grafik für TeX und LaTeX*. DANTE – Lehmanns, Heidelberg/Hamburg, third edition, 2006.

Figure 2: Summary of all parameters for `pst-eps`

name	values	default	meaning
`bbllx`	`<value[unit]>`	`0pt`	bounding box lower left x
`bblly`	`<value[unit]>`	`0pt`	lower left y
`bburx`	`<value[unit]>`	`0pt`	upper right x
`bburx`	`<value[unit]>`	`0pt`	upper right y
`makeeps`	`none|new|all|all*`	`new`	none: do nothing new: create, when non exists all: create allways all*: ask before overwriting
`checkfile`	`<true|false>`	`false`	check before overwriting
`headerfile`	`<filename>`	`{}`	filename of header to include
`headers`	`none|all|user`	`none`	none: no headers all: include all PSTricks header files user: include only the header `headerfile`
`GraphicsRef`	`<x,y>`	`{}`	reference point
`Translation`	`<x,y>`	`{}`	set another origin
`Rotation`	`<value>`	`{}`	rotation angle
`Scale`	`<value1 value2>`	`{}`	scaling

[5] Timothy Van Zandt. *PSTricks - PostScript macros for generic TEX*. http://www.tug.org/application/PSTricks, 1993.

[6] Timothy Van Zandt. ***multido.tex** - a loop macro, that supports fixed-point addition*. CTAN:/graphics/pstricks/generic/multido.tex, 1997.

[7] Timothy Van Zandt. ***pst-eps**: Exporting eps images*. CTAN:graphics/pstricks/generic/, 1999.

[8] Timothy Van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index

`bbllx`, 4
`bblly`, 4
`bburx`, 4
`bbury`, 4
bounding box, 2

`EPS`, 2

`GraphicsRef`, 4

`headerfile`, 4
`headers`, 4

`makeeps`, 4

Parameter
    `GraphicsRef`, 4
    `Rotation`, 4
    `Scale`, 4
    `bbllx`, 4
    `bblly`, 4
    `bburx`, 4
    `bbury`, 4
    `headerfile`, 4
    `headers`, 4
    `makeeps`, 4
`\PSTtoEPS`, 3

`Rotation`, 4

`Scale`, 4
Syntax
    `\PSTtoEPS`, 3
    `\TeXtoEPS`, 3

`\TeXtoEPS`, 3