Trivial Experiments with psTricks manipulation

Radhakrishnan CV and Rajagopal CV
River Valley Technologies, Trivandrum, India
http://www.river-valley.com

Antoine Chambert-Loir
Ecole polytechnique, Palaiseau Cedex, France
http://www.math.polytechnique.fr/~chambert

August 10, 2003

1 Objectives

psTricks macros cannot be used with pdfTEX, since the high level PostScript language commands generated by the psTricks are unknown to pdfTEX. As such, a package *viz.*, pdfTricks.sty has been written to circumvent this limitation, so that the extensive facilities offered by the powerful psTricks package can be made use of in a pdfTEX document. This is brought by making use of the shell escape function available in the web2c TEX compiler, while this package is of no use to other commercial implementations.

Shell escape provides us the facility of stopping a TEX run midway, perform the functions that we do with a shell command, complete those functions, return to the TEX run and finish the job. Within a package, this facility can be invoked by using \write18{shell command}. So if you use a line:

```
write18 ls -l
```

TEX will stop at the moment when it encounter the \write18 command, escapes to shell, executes ls -l command and return to the compilation process.

2 Shell escape

In all the web2c TEX implementations, shell escape is turned off by default. You can either turn it on by changing the line in your texmf.cnf (configuration file of your TEX system) to:

```
shell_escape = t
```

which will be 'f' by default. You have to recreate all the formats like `pdflatex.fmt` or whatever you might need. This is not a wise step though, since it makes you open to the assault of Trojan macros without your knowledge.

The more elegant way will be to invoke shell escape with a switch, each time you run the compiler like:

```
pdflatex -shell-escape <file name>
```

This eliminates the vulnerability to macros/packages written by intelligent criminal minds.

3 Usage

The usage is very simple, as usual you have to load the package with a package loading line in the preamble of your document:

```
\usepackage{pdftricks}
```

All the packages that are needed for the compilation of the `pstricks` code shall be enclosed within an environment `\begin{psinputs}` ... `\end{psinputs}` in order to make it available to all the figure documents during compilation. An example of this will look like

```
\begin{psinputs}
 \usepackage{pstricks}
 \usepackage{color}
 \usepackage{pstcol}
 \usepackage{pst-plot}
 \usepackage{pst-tree}
 \usepackage{pst-eps}
 \usepackage{multido}
 \usepackage{pst-node}
 \usepackage{pst-eps}
 ...
\end{psinputs}
```

Each `pspicture` environment along with any macro code shall be enclosed within an environment *viz.*, `pdfpic`. This will enable pdfLaTeX to write external files like `fig1.tex`, `fig2.tex`, ..., depending on the number of `psTricks` figures you have included in your document. `pdftricks` will intelligently find out whether your pdfTeX implementation has shell escape facility enabled or not. Depending on this, there are two ways of approaching the problem which are given in the succeeding sections.

3.1 With shell escape

With shell escape option invoked, one need not bother anything. The usual compilation with pdfLaTeX will result in a `pdf` document with all the figures appearing in the correct positions. pdfTeX will do the necessary job of writing all the `pstricks` code to self standing external `*.tex` documents, escapes to shell, compiles the `*.tex` document, converts to `*.eps`, translates to `*.pdf` and includes in the location where the `pstricks` code appeared.

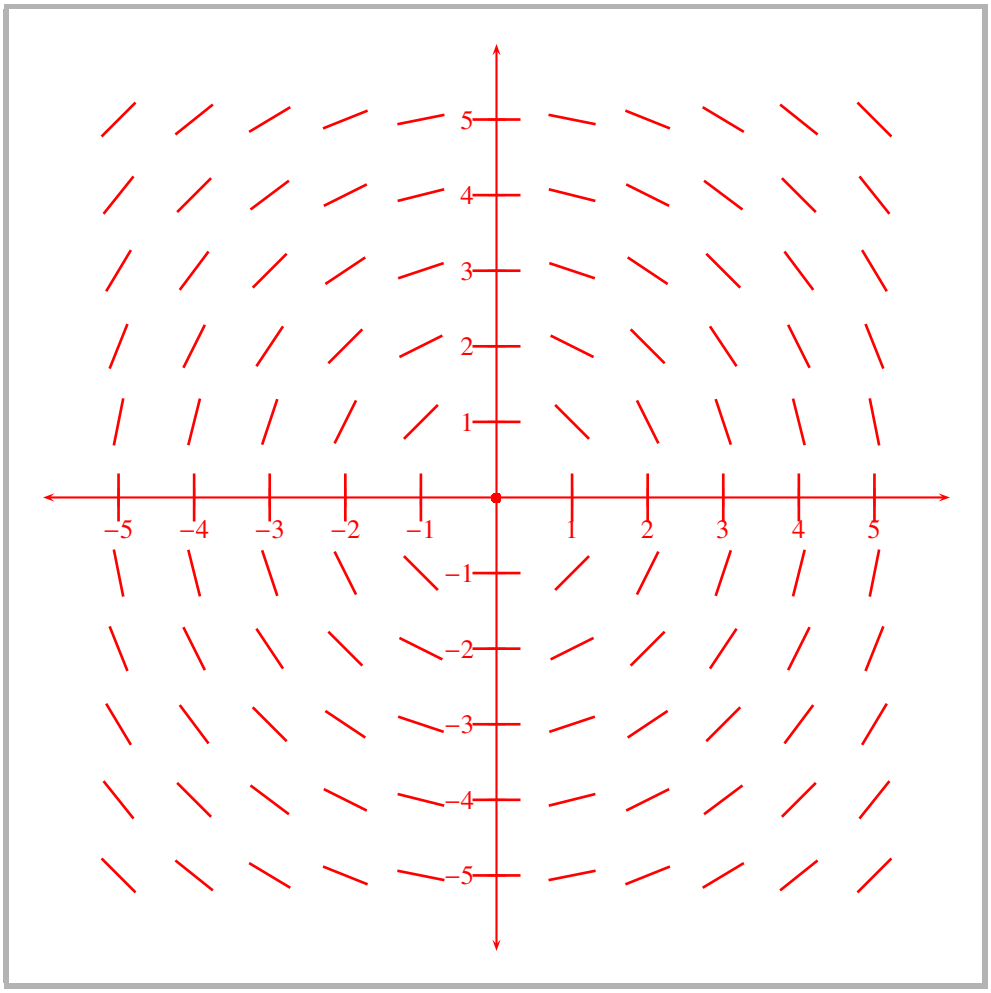
3.2 Without shell escape

A special shell script viz., `pst2pdf` is also provided. Now as the second step, you have to execute this shell script, `pst2pdf` which will compile all the newly written `fig1.tex`,..., and respective `*.dvi` will be generated, which will be translated into respective `*.eps` files and finally translated to `*.pdf` by calling `epstopdf`. As a final step, you shall run pdfLaTeX again, so that the `*.pdf` figures will be included with the usual `\includegraphics` command automatically in the locations wherever the `pstricks` code appeared.

See an example below:

```
\begin{pdfdisplay}
\definecolor{lightblue}{rgb}{0,0,.5}
\definecolor{Navy}{rgb}{0,0,0.5}
\definecolor{LemonChiffon}{rgb}{1,0.98,0.8}
\definecolor{ForestGreen}{rgb}{0.13,0.55,0.13}
\SpecialCoor
\begin{pspicture}(-6,-6)(6,6)
  \psaxes{<<->>}(0,0)(-6,-6)(6,6)
  \psset{arrows=->>}
  \multido{\ia=-5+1}{11}{%
\multido{\ib=-5+1}{11}{%
 \pstVerb{/x \ia\space def
    /y \ib\space def
    y 0 eq
      {/ValueTempA 0 def
       /ValueTempB 0.5 def}
      {/ValueTempZ 2 1 x x mul y y mul div add sqrt mul def
       /ValueTempA 1 ValueTempZ div def
       /ValueTempB x y ValueTempZ mul div def}
      ifelse}
 \psline(! x ValueTempA sub y ValueTempB sub)
        (! x ValueTempA add y ValueTempB add)}}
\end{pspicture}
\end{pdfdisplay}
```

This psTricks code will result in the following graphic:



The following are the steps in the sequential order:

1. latex <your document>
2. pst2pdf
3. latex <your document>

4 Options and Hooks

shell This option will invoke the shell escape functions.

`noshell` This option will suppress the shell escape functions and will assume that there is no shell escape facility.

`NoProcess` A new command `\NoProcess` has been introduced in this version to facilitate the suppressing of pdf generation of those figures whose pdf's are already available. This might prove helpful when you have more figures to process and many of them are perfected and don't need recompilation and translation everytime you run pdfLaTeX. The usage is:

```
\NoProcess{<comma separated and/or hyphen separated ranges>}
```

If you have ten figures and if you want to suppress the processing of the figure numbers 1, 2, 4 to 8 you can issue the command at the top of the document as:

```
\NoProcess{1,2,4-8}
```

There are few hooks to resize the graphic thus translated and included.

```
\configure[pdfgraphic][width=2in]
\configure[pdfgraphic][height=3in]
\configure[pdfgraphic][width=2in,height=3in]
```

the functionality is same as the `width` and `height` options as in the `\includegraphics` command. But the graphic will be restricted to aspect ratio.

5 Limitations/Dependencies

- Usage is limited to `web2c` TeX implementations.
- The shell escape commands used in this package are typical Unix commands and as such it best works in Linux/Unix flavours.
- The package needs, `graphicx` and `keyval` packages; `epstopdf`, the Perl script that translates `eps` files into `pdf`.

6 Licence

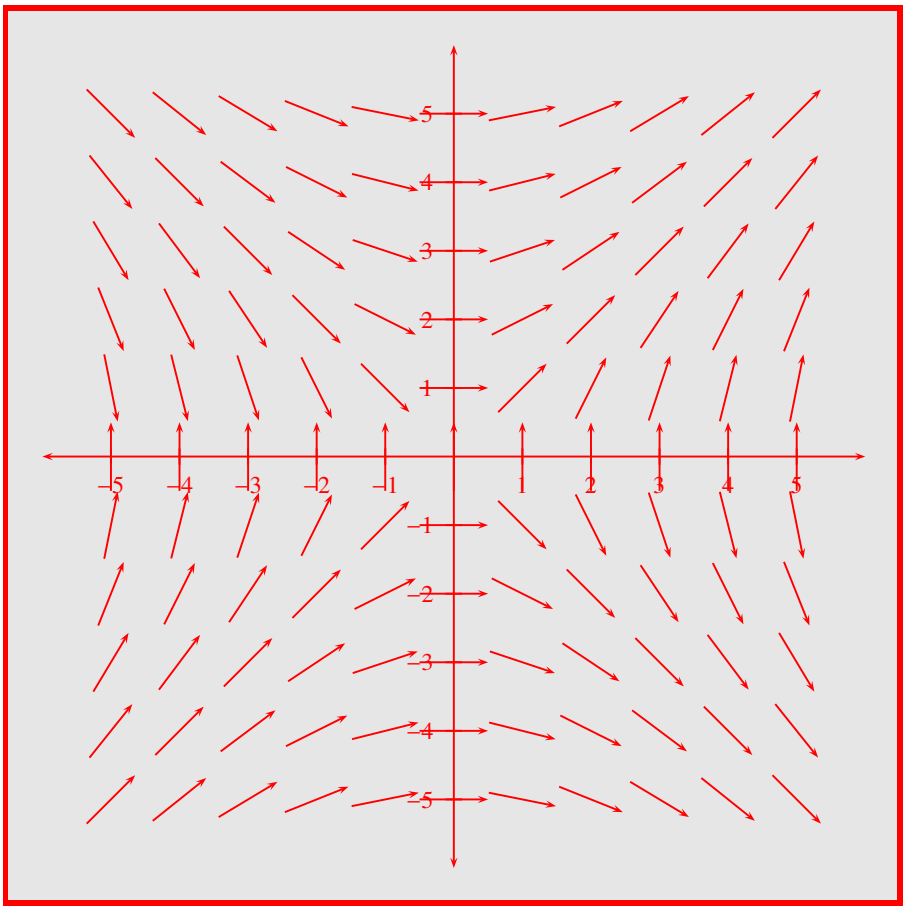
The package is distributed under gnu General Public Licence.

7 Examples

Here are some examples of pstricks code and the figures generated by pdfLaTeX.

7.1 Example 1

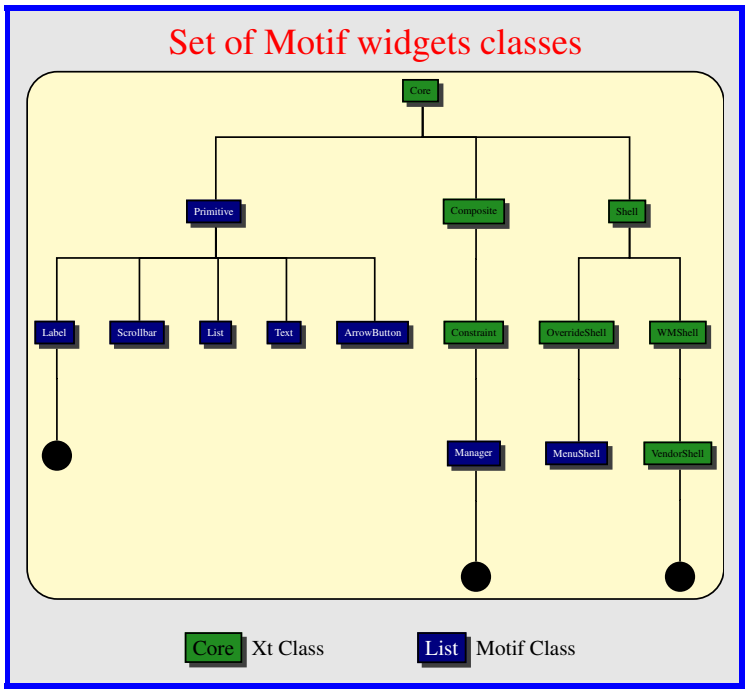
```
\configure[pdfgraphic][width=2in,linecolor=red,background=gray10]
\begin{pdfdisplay}
\definecolor{lightblue}{rgb}{0,0,.5}
\definecolor{Navy}{rgb}{0,0,0.5}
\definecolor{LemonChiffon}{rgb}{1,0.98,0.8}
\definecolor{ForestGreen}{rgb}{0.13,0.55,0.13}
\SpecialCoor
\begin{pspicture}(-6,-6)(6,6)
  \psaxes{<<->>}(0,0)(-6,-6)(6,6)
  \psset{arrows=->>}
  \multido{\ia=-5+1}{11}{%
\multido{\ib=-5+1}{11}{%
 \pstVerb{/x \ia\space def
    /y \ib\space def
    y 0 eq
      {/ValueTempA 0 def
       /ValueTempB 0.5 def}
      {/ValueTempZ 2 1 x x mul y y mul div add sqrt mul def
       /ValueTempA 1 ValueTempZ div def
       /ValueTempB x y ValueTempZ mul div def}
      ifelse}
 \psline(! x ValueTempA sub y ValueTempB sub)
        (! x ValueTempA add y ValueTempB add)}}
\end{pspicture}
\end{pdfdisplay}
```



7.2 Example 2

```
\configure[pdfgraphic][linecolor=blue,background=gray10,width=4in]
\begin{pdfdisplay}
\definecolor{lightblue}{rgb}{0,0,.5}
\definecolor{Navy}{rgb}{0,0,0.5}
\definecolor{LemonChiffon}{rgb}{1,0.98,0.8}
\definecolor{ForestGreen}{rgb}{0.13,0.55,0.13}
\newcommand{\MyNode}[2]{%
  \Tr{\psshadowbox[fillstyle=solid,fillcolor=#1]{\tiny #2}}}
\newcommand{\NoeudXt}[1]{\MyNode{ForestGreen}{#1}}
\newcommand{\NoeudMotif}[1]{\MyNode{Navy}{\textcolor{white}{#1}}}
\psset{armB=5mm,angleA=90,angleB=-90,levelsep=2cm,treesep=5mm}
\renewcommand{\psedge}[2]{\ncangle{#2}{#1}}
\TeXtoEPS
```

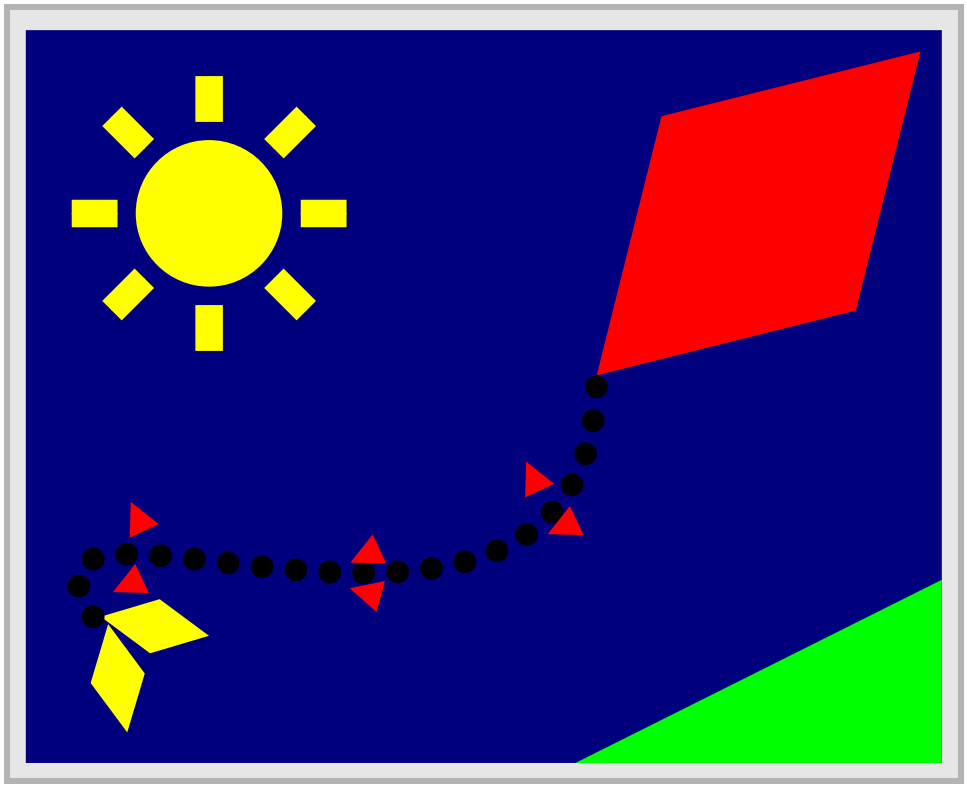
```
\begin{pspicture}(-8cm,-9.5cm)(8cm,1cm)
\rput(0,0){\LARGE\textcolor{red}{Set of Motif widgets classes}}
\rput(0,-4.8){%
  \psframebox[fillstyle=solid,fillcolor=LemonChiffon,linearc=5mm,
              cornersize=absolute]
    {\pstree{\NoeudXt{Core}}
    {\pstree{\NoeudMotif{Primitive}}
            {\pstree{\NoeudMotif{Label}}
                    {\TC*}
             \NoeudMotif{Scrollbar}
             \NoeudMotif{List}
             \NoeudMotif{Text}
             \NoeudMotif{ArrowButton}}
     \pstree{\NoeudXt{Composite}}
            {\pstree{\NoeudXt{Constraint}}
                    {\pstree{\NoeudMotif{Manager}}
                            {\TC*}}}
     \pstree{\NoeudXt{Shell}}
            {\pstree{\NoeudXt{OverrideShell}}
                    {\NoeudMotif{MenuShell}}
             \pstree{\NoeudXt{WMShell}}
                    {\pstree{\NoeudXt{VendorShell}}
                            {\TC*}}}}}}
\rput(-2,-10){%
  \psshadowbox[fillstyle=solid,fillcolor=ForestGreen]{Core} Xt Class}
\rput(2,-10){%
  \psshadowbox[fillstyle=solid,fillcolor=Navy]{%
     \textcolor{white}{List}} Motif Class}
\end{pspicture}
\endTeXtoEPS
\end{pdfdisplay}
```

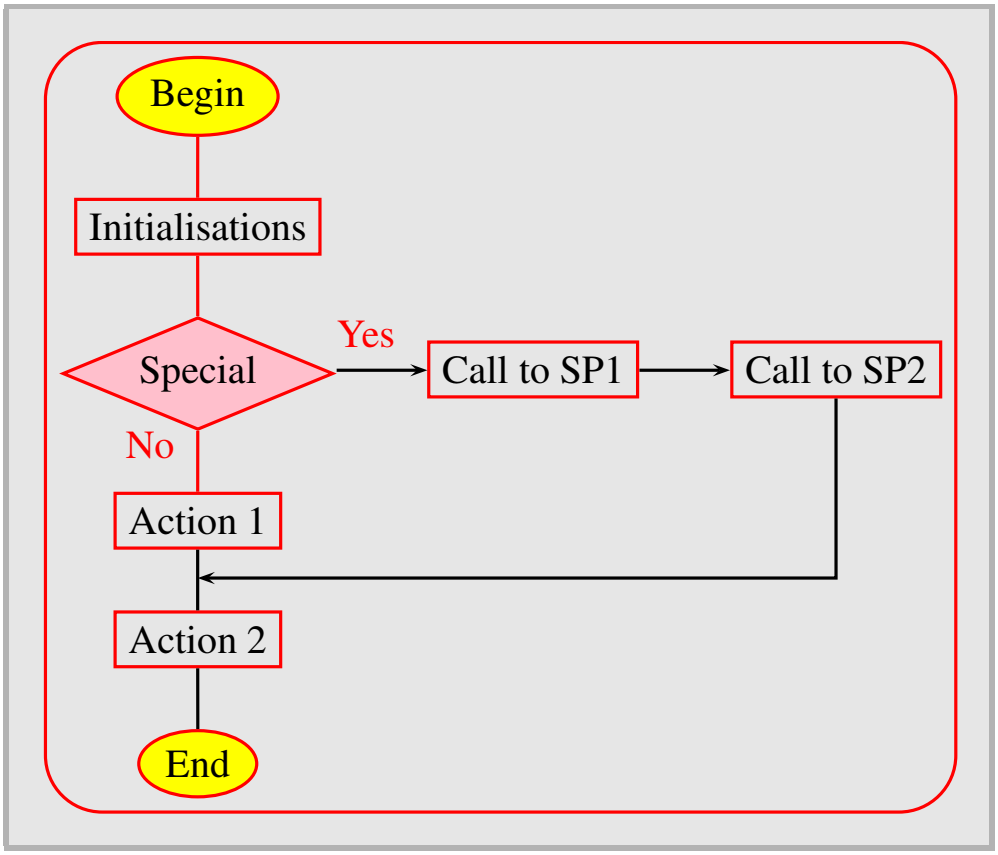


7.3 Example 3

```
\configure[pdfgraphic][linecolor=black,background=gray30]
\begin{pdfdisplay}
\TeXtoEPS
\definecolor{lightblue}{rgb}{0,0,.5}
\definecolor{Navy}{rgb}{0,0,0.5}
\definecolor{LemonChiffon}{rgb}{1,0.98,0.8}
\definecolor{ForestGreen}{rgb}{0.13,0.55,0.13}
\psset{unit=.825cm}
\begin{pspicture}(10,8)
  \psset{fillstyle=solid,linestyle=none,linewidth=0}
  \psframe[fillcolor=lightblue](10,8)
  \pscircle[fillcolor=yellow](2,6){.8} % Sun
  {% Rays
   \psset{linecolor=yellow,linestyle=solid,linewidth=.3}
   \degrees[8]
   \multido{\i=1+1}{8}{\rput{\i}(2,6){\psline(1,0)(1.5,0)}}
  }%
  \pspolygon[fillcolor=green](6,0)(10,2)(10,0)% Grass
  \psdiamond[fillcolor=red,gangle=-45](8,6)(1.5,2.5)% Kite
  \rput{45}(8,6){\pnode(-2.5,0){Kitetail}}
```

```
  \rput{-10}(.8,1.5){\psdiamond[fillcolor=yellow](.6,.1)(.6,.3)}
  \rput{-80}(.8,1.5){\psdiamond[fillcolor=yellow](.6,.1)(.6,.3)}
  \pnode(.8,1.5){Tailend}
  \nccurve[fillstyle=none,angleA=270,angleB=125,ncurvB=.9,ncurvA=1.4,
           linestyle=dotted,dotstyle=square,linewidth=.25]{Kitetail}{Tailend}
  \newcommand{\bunting}{\pstriangle(.35,.35)}
  \psset{fillcolor=red,labelsep=.01}
  \naput[nrot=115,npos=.15]{\bunting}
  \nbput[nrot=25,npos=.15]{\bunting}
  \naput[nrot=75,npos=.4]{\bunting}
  \nbput[nrot=115,npos=.4]{\bunting}
  \naput[nrot=115,npos=.7]{\bunting}
  \nbput[nrot=25,npos=.7]{\bunting}
\end{pspicture}
\endTeXtoEPS
\end{pdfdisplay}
```






```
\begin{pdfdisplay}

\definecolor{Pink}{rgb}{1.,0.75,0.8}
%  Flow diagram with the psmatrix environment\\[5mm]

  \psframebox[linearc=5mm,cornersize=absolute]{%
    \begin{psmatrix}[rowsep=0.5cm,colsep=0.8cm]
      \psovalbox[fillstyle=solid,fillcolor=yellow]{Begin} \\
      \psframebox{Initialisations} \\
      \psdiabox[fillstyle=solid,fillcolor=Pink]{Special} &
         \psframebox{Call to SP1} & \psframebox{Call to SP2} \\
      \psframebox{Action 1} \\
      \psframebox{Action 2} \\
      \psovalbox[fillstyle=solid,fillcolor=yellow]{End}
      % Links
      \ncline{1,1}{2,1}
      \ncline{2,1}{3,1}
```

```
      \ncline{3,1}{4,1}<{\textcolor{red}{No}}
      \ncline{4,1}{5,1}
      \ncline{5,1}{6,1}
      \ncline{->}{3,1}{3,2}^{\textcolor{red}{Yes}}
      \ncline{->}{3,2}{3,3}
      \ncbar[angleA=-90,armB=0,nodesepB=2.5mm]{->}{3,3}{4,1}
    \end{psmatrix}}
\end{pdfdisplay}
```