

Zpravodaj Československého sdružení uživatelů TeXu

Herbert Voss

The current state of the PSTricks project

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 20 (2010), No. 1-2, 39–67

Persistent URL: <http://dml.cz/dmlcz/149997>

Terms of use:

© Československé sdružení uživatelů TeXu, 2010

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Abstrakt

Článek přináší přehled novinek a rozšíření balíčku PSTricks. Je to zkratka pro PostScript Tricks, přeloženo do jazyka českého jako triky s programovacím jazykem PostScript. Jedná se o balíček, který široce využívá *starý* programovací jazyk PostScript. Ten je také znám pod zkratkou PCL (Page Code Language) vyvíjený firmou Adobe Systems od roku 1984. PDF je odnoží PostScriptu s několika důležitými rozšířeními, ale bez možnosti matematických výpočtů.

Klíčová slova: Balíček PSTricks, PostScript, PDF.

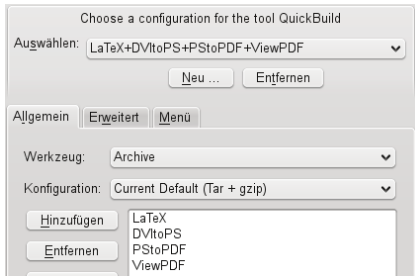
doi: 10.5300/2010-1-2/39

Creating a graphic with PSTricks

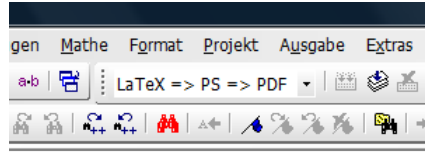
$\text{T}_{\text{E}}\text{X}$ as a typesetting machine cannot make full use of the possibilities of PostScript. There cannot be a direct interaction between $\text{T}_{\text{E}}\text{X}$ and PostScript, it is more of a one way communication from $\text{T}_{\text{E}}\text{X}$ to PostScript. The interface between these two systems is the DVI converter `dvips` which converts the DVI output of $\text{T}_{\text{E}}\text{X}$ into the PostScript format. Here we use DVI output as an intermediate format only, it is not really of interest. On the $\text{T}_{\text{E}}\text{X}$ side the user has to reserve some space (a box) which is filled on the PostScript side with a graphic or some text. If this space is not reserved then everything will be printed over the text, depending on the current point before the PostScript related code. This PostScript code must be transferred to from $\text{T}_{\text{E}}\text{X}$ via the DVI output with the macro `\special`. Its contents are ignored by $\text{T}_{\text{E}}\text{X}$ and passed to PostScript where it will be executed by a PostScript interpreter such as Ghostscript or Distiller.

Figure 2 on page 65 shows the important flow from the $\text{T}_{\text{E}}\text{X}$ source to the destination format PDF. When using a graphical user interface (GUI) for editing the $\text{T}_{\text{E}}\text{X}$ source one can choose the output format PDF, but has to ensure that it follows the path `!TEX-dvips-ps2pdf`. For three often-used GUIs, Figure 1 on the next page shows the preferences to select for this method of compiling a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ source document.

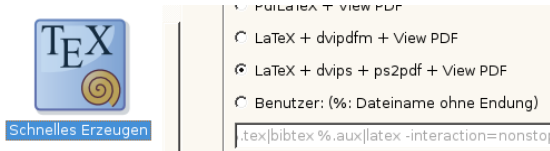
The first example shows the use of PostScript related specials without reserving any space on $\text{T}_{\text{E}}\text{X}$ level. In the beginning and in the end there is a named node which can be connect by a line or a curve, in this by a curve with an arrow and stroke opacity. It is drawn over the text, there was no space reserved on the $\text{T}_{\text{E}}\text{X}$ level. In examples like these it may be useful to draw directly over the main text part of the document.



(a) Kile (Linux)



(b) T_EXnicCenter (Windows)



(c) T_EXmaker (Linux/Windows/MacOSX)

Figure 1: Preferences for the destination format PDF via the intermediate format DVI (DeVice Independent).

The beginning of all: Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like »Huardest gefburn«. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language. And this is the end my friend.

```
\usepackage{blindtext,pst-node}
\SpecialCoor

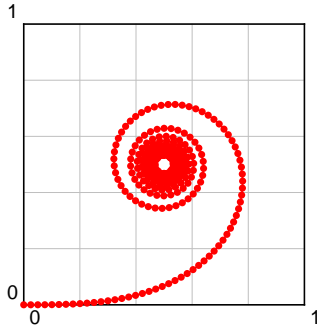
\rnode{A}{The beginning of all: }
\blindtext \rnode{E}{ and this is
the end my friend.}
\ncurve[linewidth=5pt,
strokeopacity=0.4,ncurvB=0.9,
arrowscale=1.5,arrows=-D>]{A}{E}
```

The advantage of PSTricks in comparison to METAPOST or TikZ is the possibility of using all features of a powerful programming language with very good support for graphic operations. This is the reason why *any* calculation with mathematical functions or large external data sets can be done before the output is printed. Solving a differential equation on the fly is as possible as drawing three dimensional solids with hidden lines and surfaces; everything is done on the PostScript side. In the T_EX or L^AT_EX source one has only to define the space of the box and to describe the code with T_EX or L^AT_EX macros which are then

passed as specials to PostScript. The next example shows the output of the solution of the differential equation system of first order:

$$\dot{x} = \cos \frac{\pi}{2} \cdot x^2 \quad (1)$$

$$\dot{y} = \sin \frac{\pi}{2} \cdot x^2 \quad (2)$$



```
\usepackage{pstricks-add}
\psset{unit=5}
\begin{pspicture}(-0.04,-0.04)(1,1)
\psgrid[subgriddiv=5,
subgridcolor=lightgray]
\psplotDiffEqn[whichabs=0,
whichord=1,linecolor=red,
method=rk4,algebraic,
plotpoints=400,showpoints=true]%
{0}{10}{0 0}%
{cos(Pi*x^2/2)|sin(Pi*x^2/2)}
\end{pspicture}
```

The parameters (coordinates) of the environment `pspicture` have a different meaning for \TeX and PostScript; for \TeX they define width and height or the box which has its lower left side at the current point in \TeX . When there is no shift defined, then the lower side of this box is always on the baseline of the current text line. It is for \TeX just the same as a box for a single letter. \TeX needs the coordinates only for its formatting; what will be inserted later into this box is not of interest to \TeX .

For PostScript the coordinates define a two dimensional area with the lower left and upper right corner of the rectangle. The origin of this cartesian coordinate system may be inside or outside of this rectangle; it depends on the values of the coordinates. An example: `begin{pspicture}(-1,-2)(4,4)` defines for \TeX a box with a width of five length units ($4 - (-1)$) and a height of six length units ($4 - (-2)$). For PostScript the origin of this box is one length unit to the right and two length units up, measured from the current point, which is the lower left of the \TeX box.

The next example defines, on the \TeX level, a box with a width and height of $2.5\text{cm} \times 1.5\text{cm}$.

The box with a reserved space of $2,5\text{cm} \times 1,5\text{cm}$ is by definition with its lower left side at the current point. The lower side is on the baseline, which can easily be



seen on the this box: its internal origin maybe somewhere, also outside this box. In this example the origin is at $(1, 0.5)$ units, measured from the lower left corner of the box.

```
\usepackage{pstricks,pst-plot}
```

\raggedright The box with a reserved space of 2,5cm \cdot 1,5cm is by definition with its lower left side at the current point.

The lower side is on the baseline, which can easily be seen on the this box:

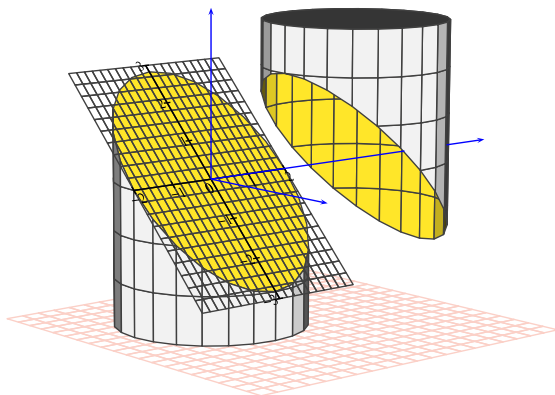
```
\psframebox[framesep=0]{%
  \begin{pspicture}(-1,-0.5)(1.5,1)
    \psaxes[labels=none]{->}(0,0)(-1,-0.5)(1.5,1)
  \end{pspicture}}
```

its internal origin mybe somewhere, alaso outside this box. In this example the origin is at $\texttt{(1,0.5)}$ units, measured from the lower left corner of the box.

There are several optional arguments for the main environment `pspicture`:

```
\pspicture * [Options] (xMin,yMin)(xMax,yMax) ... \endpspicture
\pspicture * [Options] (xMax,yMax) ... \endpspicture
\begin{pspicture * } [Options] (xMin,yMin)(xMax,yMax)... \end{pspicture * }
\begin{pspicture * } [Options] (xMax,yMax) ... \end{pspicture * }
```

Another main focus of `PSTricks` is three dimensional graphic objects, with support for the hidden line algorithm for lines and surfaces. The powerful package `pst-solides3d` allows combining any three dimensional solid, given by coordinates or a mathematical expression. The following example shows the book cover image from [2], a cylindrical object divided by a plane into two parts, one of which is moved and rotated.



```
\usepackage[dvipsnames]{pstricks}
\usepackage{pst-solides3d}

\begin{pspicture}[solidmemory](-4,-5)(7,4)
\psset{viewpoint=50 -40 10 rtp2xyz,Decran=50,linecolor=darkgray,lightsrc=viewpoint}
\psSolid[object=grille,action=draw,base=-3 5 -3 5,linecolor=Salmon!40](0,0,-3)
```

```

\psSolid[object=cylindre,r=2,h=6,ngrid=6 24,plansepare={{[0.707 0 0.707 0]}},
  name=Zylinder,action=none](0,0,-3)
\psSolid[object=load,load=Zylinder1,fillcolor=black!5,fc0=(Goldenrod)]
\psSolid[object=load,load=Zylinder0,RotZ=90,fillcolor=black!5,rm=0,hollow,
  incolor=Goldenrod](0,4,0)
\psSolid[object=plan,action=draw,definition=equation,
  args={{[0.707 0 0.707 0] 90},base=-2 2 -3 3,planmarks]
\psSolid[object=line,args=0 0 0 5.5 0,linicolor=blue]% erste Haelfte y-Achse
\color{white}
\axesIID[showOrigin=false,linicolor=blue](0,6.8,0)(3.5,8,3.5)
\end{pspicture}

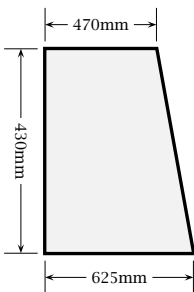
```

PSTricks project background

The first version of the main package `pstricks` was written by Timothy Van Zandt and published nearly 20 years ago and is still the base package for the so-called PSTricks project. All additional packages published since 1991 are listed at the PSTricks web page <http://PSTricks.tug.org>. In [1] and especially [2] the packages are described and shown with a lot of examples. Here, we will list the packages with only one significant example, to give a glimpse at what the package provides. More examples or some more information can be found on the PSTricks web page (<http://PSTricks.tug.org>), CTAN (<http://mirrors.ctan.org>) or your local \TeX distribution's documentation, e. g., using the `texdoc` program.

All PSTricks packages load by default the main package `pstricks`, which itself loads the package `xcolor`, which has a better support for colors than the package `color`.

`pstricks`: Main package with the base macros for lines, curves, areas, etc.

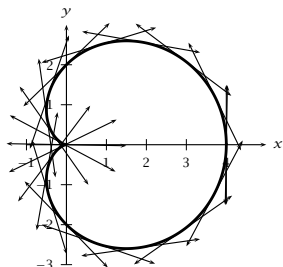


```

\usepackage{pstricks}
\psset{unit=0.05mm}% 1:20,or=0.1mm for 1:10
\begin{pspicture}(-100,-100)(1000,650)
  \pspolygon[linewidth=2pt,fillcolor=gray!10,
    fillstyle=solid](0,0)(0,470)(860,625)(860,0)
  \psset{linewidth=0.2pt,arrowscale=2,tbarsize=10pt}
  \psline{|<->|}(0,-100)(860,-100)
  \rput*(430,-100){430mm}
  \psline{|<->|}(960,0)(960,625)
  \rput*{90}(960,312.5){625mm}
  \psline{|<->|}(-100,0)(-100,470)
  \rput*{90}(-100,235){470mm}
\end{pspicture}

```

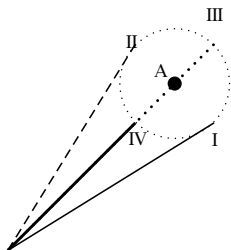
pstricks-add: Extended base macros for the packages pstricks, pst-node, and pst-plot



```
\usepackage{pstricks-add}
\usepackage{multido}

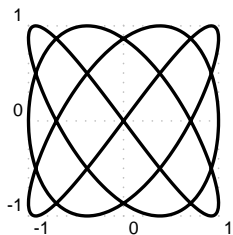
\psscalebox{0.75}{%
\begin{pspicture}(-1,-3)(4.75,3)
\psaxes{->}(0,0)(-1,-3)(5,3)
\psplot[polarplot,linewidth=2pt,algebraic,
plotpoints=500]{0}{6.289}{2*(1+cos(x))}
\multido{\r=0.000+0.314}{21}{ %
\psplotTangent[polarplot,Derive=-2*sin(x),
algebraic,arrows=<->]{\r}{1.5}{2*(1+cos(x))}}
\end{pspicture}}
```

pst-node: Nodes and node connections in text and a matrix



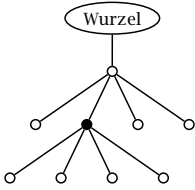
```
\usepackage{pst-node}
\SpecialCoor
\begin{Example}[1txps]{\xLcs{pnode}\xLcs{uput}}
\begin{pspicture}(4,4)
\pnode(3,3){A}\psdot{dotscale=2}(A)\uput[135](A){A}
\pscircle[linestyle=dotted](A){1}
\psline([nodesep=1,angle=-45]A)\uput[0](3.5,2){I}
\psline[linestyle=dashed]([nodesep=-1,angle=-45]A)
\uput[-45](2,4){II}
\psline[linestyle=dotted,linewidth=1.5pt]%
([offset=1,angle=-45]A)\uput[-225](4,4){III}
\psline[linewidth=1.5pt]([offset=1,angle=135]A)
\uput[0](2,2){IV} \nccurve{->}{A}{0,0}
\end{pspicture}
```

pst-plot: Plotting of mathematical functions or external data sets



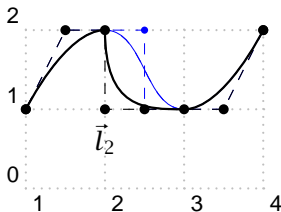
```
\usepackage{pst-plot}
\psset{xunit=1.5cm,yunit=1.5cm}
\begin{pspicture}[showgrid=true](-1.1,-1.1)(1.1,1.1)
\psparametricplot[plotstyle=curve,linewidth=1.5pt,
plotpoints=200]{-360}{360}%
{t 1.5 mul sin t 2 mul 60 add sin}
\end{pspicture}
```

pst-tree: Trees



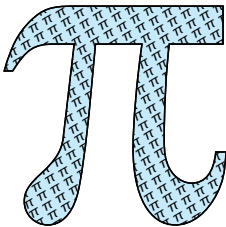
```
\usepackage{pst-tree}
\pstree[levelsep=1cm,radius=3pt]{\Toval{Wurzel}}{%
  \pstree{\TC}{%
    \TC
    \pstree{\TC*}{\TC\TC\TC\TC}%
    \TC\TC}}
```

pst-bezier: Bézier curves



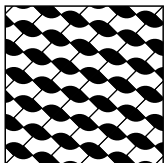
```
\usepackage{pst-bezier}
\pspicture[showgrid=true](5,3)\psset{showpoints=true}
\psbcurve[linecolor=blue,linewidth=0.01](1,1)%
(2,2)(3,1)(4,2)
\psbcurve(1,1)(2,2)l(2,1)(3,1)(4,2)
\uput[-90](2,1){$\vec{l}_2$}
\endpspicture
```

pst-text: Character and text manipulation



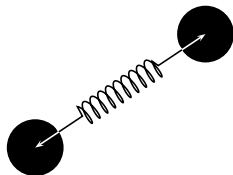
```
\usepackage{pst-text}\usepackage[tiling]{pst-fill}
\DeclareFixedFont{\ps}{U}{psy}{m}{n}{8cm}
\psboxfill{\footnotesize$\pi$}
\begin{pspicture}(0,0)(.25\linewidth,8)
\pscharpath[fillstyle=solid,fillcolor=cyan!20,
  addfillstyle=boxfill,fillangle=30,
  fillsep=0.6mm]{\rput[b](-0.5,0){\ps\char112}}
\end{pspicture}
```

pst-fill: Filling and tiling



```
\usepackage[tiling]{pst-fill}
\newcommand\FArcW{%
  \begin{pspicture}(-0.25,-0.25)(0.25,0.25)%
    \pswedge*(-0.25,-0.25){0.25}{0}{90}
    \pswedge*(0.25,0.25){0.25}{180}{270}
    \psframe[linewidth=0.1pt](-0.25,-0.25)(0.25,0.25)
  \end{pspicture}}
\begin{pspicture}(3.1,3.1) \psboxfill{\FArcLW}
\psframe[fillstyle=boxfill,fillcyclex=2,fillangle=45](3,3)
\end{pspicture}
```


pst-coil: Coils and zigzag lines



```
\usepackage{pst-node,pst-coil} \SpecialCoor
\begin{pspicture}(4,3)
\cnode*(0.5,0.5){0.5}{A}\cnode*(3.5,2.5){0.5}{B}
\pccoil[coilwidth=0.4,coilaspect=35,coilheight=0.5,
linecolor=white]{<->}(A)(B)
\nccoil[coilwidth=0.4,coilaspect=35,coilheight=0.5]{A}{B}
\end{pspicture}
```

pst-grad: Color gradients



```
\usepackage{pst-grad}
\begin{pspicture}(3,2.25)
\psframe[fillstyle=gradient](3,2)
\end{pspicture}
```

pst-slpe: Extended color gradients



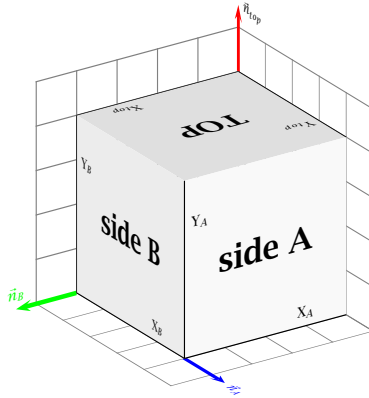
```
\usepackage{pst-slpe}
\def\st#1{\makebox[2.75cm]{\vcenter to55pt}{#1$}}
\begin{pspicture}(2.5,2.5)
\psset{fading,endfading=0.75,linecolor=black!40}
\psframe*(-0.3,-0.25)(3.5,20pt)
\psframebox[fillstyle=slope]{\LARGE\st{slope}}
\end{pspicture}
```

pst-blur: Shadows

PSTricks

```
\usepackage{pst-blur,pst-text}
\DeclareFixedFont{\RM}{T1}{ptm}{b}{n}{1.75cm}
\psset{shadow=true,blur=true,shadowsize=10pt,
blurradius=5pt}
\pscharpath{\RM PSTricks}
```

pst-3d: Basic macros for tilting and three dimensional objects

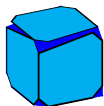
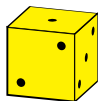


```

\usepackage{pst-3d}
\begin{pspicture}(-4.5,-3)(3,4.75) \psset{viewpoint=1 1.5 0.8}
{\psset{gridlabels=0pt,subgriddiv=0,gridcolor=black!50}
\ThreeDput[normal=0 0 1]{\psgrid(5,5)}\ThreeDput[normal=0 -1 0]{\psgrid(5,5)}
\ThreeDput[normal=1 0 0]{\psgrid(5,5)}
\ThreeDput[normal=0 0 1]{\psline[linewidth=3pt,linecolor=blue]{->}(4,4)(4,5.5)
\uput[90](4,5.5){\psrotateleft{\textcolor{blue}{\vec{n}_A}}}}}%$xy
\ThreeDput[normal=0 -1 0]{\psline[linewidth=3pt,linecolor=green]{->}(4,0)(5.5,0)
\uput[90](5.5,0){\psscalebox{-1 1}{\textcolor{green}{\vec{n}_B}}}}}%$xz
\ThreeDput[normal=1 0 0]{\psline[linewidth=3pt,linecolor=red]{->}(0,4)(0,5.5)
\uput[0](0,5.5){\vec{n}_top}} %yz
\ThreeDput[normal=0 0 1](0,0,4){\psframe*[linecolor=gray!25](4,4)
\rput(2,2){\Huge\textbf{TOP}}}
\ThreeDput[normal=0 1 0](4,4,0){\psframe*[linecolor=gray!5](4,4)
\rput(2,2){\Huge\textbf{side A}}}
\ThreeDput[normal=1 0 0](4,0,0){\psframe*[linecolor=gray!15](4,4)
\rput(2,2){\Huge\textbf{side B}}}
% Die kleinen Achsen
\ThreeDput[normal=0 0 1](0,0,4){\psline(4,0)\uput[90](3,0){X$_{top}$}
\psline(0,4)\uput[0](0,3){Y$_{top}$}}
\ThreeDput[normal=0 1 0](4,4,0){\psline(4,0)\uput[90](3,0){X$_{A}$}
\psline(0,4)\uput[0](0,3){Y$_{A}$}}
\ThreeDput[normal=1 0 0](4,0,0){\psline(4,0)\uput[90](3,0){X$_{B}$}
\psline(0,4)\uput[0](0,3){Y$_{B}$}}
\end{pspicture}

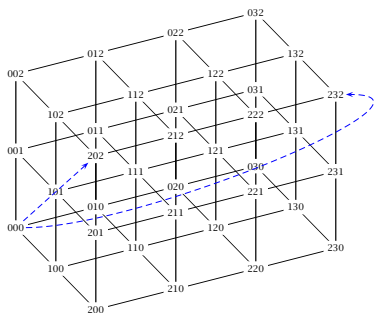
```

pst-ob3d: Simple three dimensional objects



```
\usepackage{pst-ob3d}\SpecialCoor
\begin{pspicture}(-1,-0.5)(3,1.7)
\psset{fillstyle=solid,fillcolor=yellow,
RandomFaces=true} \PstDie[viewpoint=1 -3 1]
\rput(1.5,0){\PstCube[Corners=true,CornersColor=blue,
fillstyle=solid,fillcolor=cyan,
viewpoint=1 2 1]{1}{1}{1}}
\end{pspicture}
```

pst-gr3d: Simple three dimensional grids



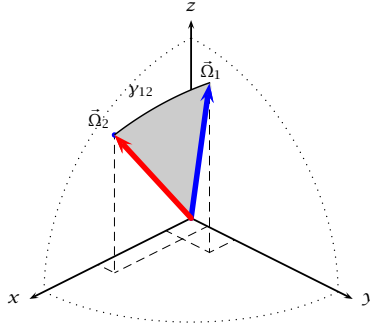
```
\usepackage{pst-gr3d,pst-node,multido}
\SpecialCoor
\psscalebox{0.5}{\footnotesize%
\PstGridThreeD[GridThreeDNodes,unit=2.5](2,3,2)
\multido{\ix=0+1}{3}{%
\multido{\iy=0+1}{4}{%
\multido{\iz=0+1}{3}{%
\rput*(Gr3dNode\ix\iy\iz){$\ix\iy\iz$}}}
\psset{linecolor=blue,linestyle=dashed,
linewidth=0.3pt,arrowscale=2,nodesep=8pt}
\pcline{->}(Gr3dNode000)(Gr3dNode202)
\pccurve{->}(Gr3dNode000)(Gr3dNode232)}
```

pst-fr3d: Three dimensional buttons



```
\usepackage{pst-fr3d}
\PstFrameBoxThreeD[FrameBoxThreeDOn=false]{\Large Off}
\quad\PstFrameBoxThreeD{\Large On}% the default
```

pst-3dplot: Three dimensional graphic objects in parallel projection

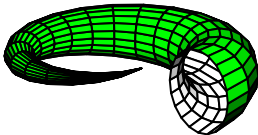


```

\usepackage{pst-3dplot}
\def\oA{\pstThreeDLine[linecolor=blue,linewidth=3pt,arrows=c->](0,0,0)(1,60,70)}
\def\oB{\pstThreeDLine[linecolor=red,linewidth=3pt,arrows=c->](0,0,0)(1,10,50)}
\def\oAB{\pstThreeDEllipse[beginAngle=58,endAngle=90](0,0,0)(1,140,40)(1,10,50)}
\begin{pspicture}(-4.8,-1.75)(4.8,3.75)
\psset{unit=4cm,drawCoor,beginAngle=90,endAngle=180,linestyle=dotted}
\pstThreeDCoor[drawing,linewidth=1pt,linecolor=black,%
linestyle=solid,xMin=0,xMax=1.1,yMin=0,yMax=1.1,zMin=0,zMax=1.1]
\pstThreeDEllipse(0,0,0)(-1,0,0)(0,1,0)\pstThreeDEllipse(0,0,0)(-1,0,0)(0,0,1)
\pstThreeDEllipse[beginAngle=0,endAngle=90](0,0,0)(0,0,1)(0,1,0)
\psset{SphericalCoor,linestyle=solid}
\pstThreeDDot[dotstyle=none](1,10,50)\pstThreeDDot[dotstyle=none](1,60,70)
\pscustom[fillstyle=solid,fillcolor=black!20,linestyle=none]{\oB\oAB\oA}\oA\oB\oAB
\pstThreeDPut[origin=lb](1.1,60,70){$\vec{\Omega}_1$}
\pstThreeDPut[origin=rb](1.2,10,50){$\vec{\Omega}_2$}
\pstThreeDPut[origin=lb](1,10,65){$\gamma_{12}$}
\end{pspicture}

```

pst-solides3d: Three dimensional graphic objects in central projection

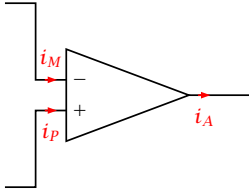


```

\usepackage{pst-solides3d}
\psset{viewpoint=100 50 20 rtp2xyz,Decran=60,
lightsrc=10 15 10}
\defFunction[algebraic]{corne}(u,v) %
{(2 + u*cos(v))*sin(2*pi*u)} %
{(2 + u*cos(v))*cos(2*pi*u) + 2*u} %
{u *sin(v)}
\begin{pspicture}(-2,-2)(2,2)
\psSolid[object=surfaceparametree,base=0 1 0 2 pi mul,
function=corne,ngrid=20]
\end{pspicture}

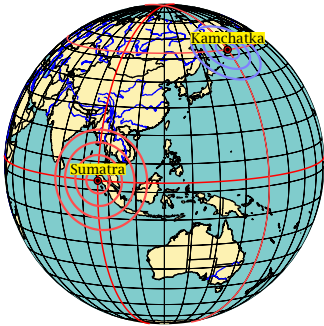
```

pst-circ: Electrical circuits for electronic and microelectronic



```
\usepackage{pst-circ}
\begin{pspicture}(4,3.5)
\node(0,3){A}\node(0,0){B}\node(4,1.5){C}
\OA[OAperfect=false,OAiplus,OAiminus,OAiout,
  OAipluslabel=$i_P$,OAiminuslabel=$i_M$,
  OAioutlabel=$i_A$,intensitycolor=red,
  intensitylabelcolor=red](A)(B)(C)
\end{pspicture}
```

pst-geo: Two and three dimensional geographical objects



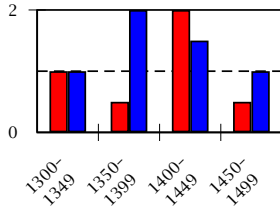
```
\usepackage{pst-map3d}
\psset{unit=0.75,Radius=5,Dobs=200,Decran=200,
  path=data/pstricks,PHI=10,THETA=120,
  circlewidth=1.5pt}
\begin{pspicture}(-5,-5)(5,5)
\WorldMapThreeD[circles=false,australia=true]
\psmeridien{95.98} \psparallel{3.30}
\psepicenter[circlecolor=red!70,waves=4,
  Rmax=2000](95.98,3.30){Sumatra}
\psmeridien[meridiencolor=red!70]{160}
\psparallel[parallelcolor=red!70]{52.76}
\psepicenter[circlecolor=blue!50]%
  (160,52.76){Kamchatka}
\end{pspicture}
```

pst-barcode: Barcodes



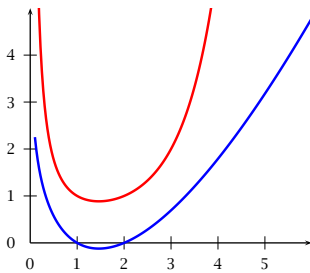
```
\usepackage{pst-barcode}
\begin{pspicture}(1in,1in)
\psbarcode{Herbert Voss Wasgenstraße 21 14129 Berlin
  http://www.dante.de/}{rows=52 columns=52}{datamatrix}
\end{pspicture}
```

pst-bar: Bar diagrams



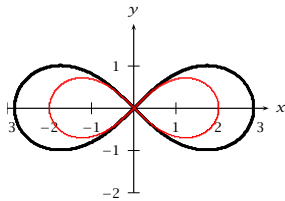
```
\usepackage{pst-plot,pst-bar}
\begin{filecontents*}{data1T.csv}
1300--1349, 1350--1399, 1400--1449, 1450--1499
1, 0.5, 2, 0.5
1, 2, 1.5, 1
\end{filecontents*}
\readpsbardata{\data}{data1T.csv}
\begin{pspicture}(-0.5,-2)(4,2)\footnotesize
\psline[linestyle=dashed](0,1)(4,1)
\psaxes[axesstyle=frame,Dy=2,labels=y](0,0)(4,2)
\psbarchart[barstyle={red,blue},barlabelrot=45,
chartstyle=cluster]{\data}
\end{pspicture}
```

pst-math: Extended PostScript functions



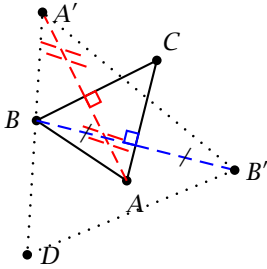
```
\usepackage{pst-plot,pst-math}
\begin{pspicture*(-0.75,-.75)(6,5) \psaxes{->}(6,5)
\psset{linewidth=1.5pt,plotpoints=200}
\psplot[linecolor=red]{.1}{6}{x GAMMA}
\psplot[linecolor=blue]{.1}{6}{x GAMMALN}
\end{pspicture}
```

pst-func: Special mathematical functions like polynomials, distributions, implicitly defined functions, etc.



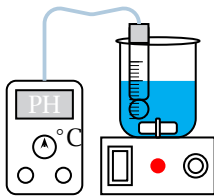
```
\usepackage{pst-func,pstricks-add}
\begin{pspicture*(-3,-2.2)(3.6,2.5)
\psaxes{->}(0,0)(-3,-2)(3.2,2)[x$,0][y$,90]
\psplotImp[linewidth=2pt,algebraic](-5,-2.2)(5,2.4){
(x^2+y^2)^2-8*(x^2-y^2)}
\rput*(2,1.5){$\left(x^2+y^2\right)^2-8(x^2-y^2)=0$}
\psplotImp[linewidth=1pt,linecolor=red,
algebraic](-5,-2.2)(5,2.4){(x^2+y^2)^2-4*(x^2-y^2)}
\end{pspicture}
```

pst-eucl: Euclidian geometry



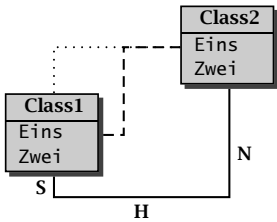
```
\usepackage{pst-eucl}
\psset{unit=0.4}%
\begin{pspicture}(0,-2)(8,7)
\pstTriangle(1,3){B}(5,5){C}(4,1){A}
\pstOrtSym{A}{B}{C}{D} \psset{CodeFig=true}
\pstOrtSym[CodeFigColor=red]{C}{B}{A}
\pstOrtSym[SegmentSymbol=pstslash,dotsep=3mm,
linestyle=dotted,CodeFigColor=blue]{C}{A}{B}
\pspolygon[linestyle=dotted,linewidth=1pt](A')(B')(D)
\end{pspicture}
```

pst-labo: Chemical objects



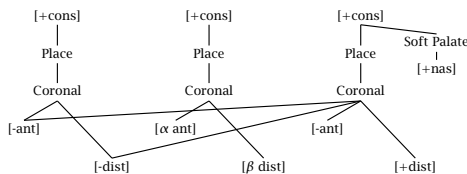
```
\usepackage{pst-labo}
\psset{unit=0.5cm,glassType=becher,burette=false}
\pstDosage[phmetre]
```

pst-uml: UML diagrams



```
\usepackage{pst-uml}
\begin{pspicture}(5,4)
\rput(1,1.5){\rnode{A}{\umlClass{Class1}{Eins}\Zwei}}
\rput(4,3){\rnode{B}{\umlClass{Class2}{Eins}\Zwei}}
\end{pspicture} \psset{linewidth=1pt}%
\ncNE[linestyle=dotted]{A}{B}\ncEVE[linestyle=dashed]{A}{B}
\ncSHN{A}{B}\nbut[npos=0.5]{\textbf{S}}
\nbut[npos=1.5]{\textbf{H}}\nbut[npos=2.5]{\textbf{N}}
```

pst-asr: “Autosegmental representations” for linguistic



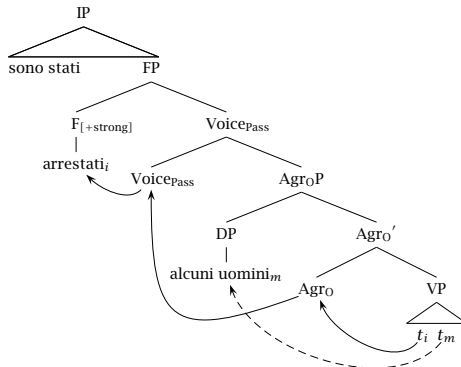
```

\usepackage{pst-asr}

\newsstyle{dotted}{linestyle=dotted,linewidth=1.2pt,dotsep=1.6pt}
\newsstyle{crossing}{xed=true,xedtype=\xedcirc,style=dotted}
\newsstyle{dotted}{linestyle=dotted,linewidth=1.2pt,dotsep=1.6pt}
\newsstyle{crossing}{xed=true,xedtype=\xedcirc,style=dotted}
\newtier{softpal,ant,dist,nasal} \tiershortcuts
\psset{xbgap=1.5in,yunit=3em,ts=0 (Pg),sy=1 (I),ph=-1 (Cg),softpal=.3 (Sg),nasal=-.4 (I),
  ant=-2 (I),dist=-3 (I),tssym=Place,sysym=\textrm{[+cons]},everyph=Coronal}
\DefList{\softpalA{2.5},\antoffset{- .22},\distoffset{.36}} \quad \asr \1{\1}{\1{}}
\@(\softpalA,softpal){Soft Palate} \-(2,sy) % softpal features
\@(\softpalA,nasal){\textrm{[+nas]}} \-(\softpalA,softpal) % ant features
\@(\antoffset,ant){\textrm{[-ant]}} \-(0,ph) \-[style=crossing](2,ph)
\@[1](\antoffset,ant){\textrm{[\alpha$ ant]}} \-(1,ph)
\@[2](\antoffset,ant){\textrm{[-ant]}} \-(2,ph) % dist features
\@(\distoffset,dist){\textrm{[-dist]}} \-(0,ph) \-[style=crossing](2,ph)
\@[1](\distoffset,dist){\textrm{[\beta$ dist]}} \-(1,ph)
\@[2](\distoffset,dist){\textrm{[+dist]}} \-(2,ph)
\endasr

```

pst-jtree: Linguistic trees

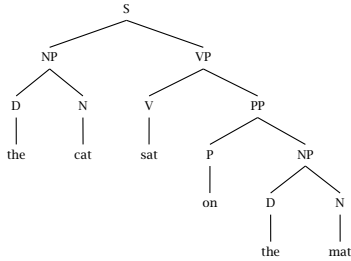


```

\usepackage{pst-jtree}
\jtree[xunit=5em,yunit=2em]
\! = {IP}
<tri>{\triline{sono stati}\hfil} ^<tri>[triratio=.95]{FP}
: {F$_{\rlap{\scriptstyle\rm [+strong]}}$}!a {Voice$_{\rlap{\scriptstyle\rm Pass}}$}
: {Voice\rlap{$_{\rm Pass}}$}@A2 {$\rm Agr\_OP$}
: {DP}!b {$\rm Agr\_O'}$
: [scaleby=.8 1]{\rm Agr\_O$}@A3 [scaleby=.8 1]{VP}
<tri>[scaleby=.4 .7]{\rnode{A5}{t_i}\hskip1ex \rnode{A6}{t_m}}.
\!a = <shortvert>{arrestati_i$}@A1 . \!b = <shortvert>{alcuni uomini_m$}@A4 .
\psset{arrows=>}\nccurve[angleA=225,angleB=-45]{A2}{A1}
\nccurve[angleA=200,angleB=-90,ncurv=1.5]{A3}{A2}\nccurve[angleA=-130,angleB=-70]{A5}{A3}
\nccurve[angleA=-130,angleB=-70,linestyle=dashed]{A6}{A4}
\endjtree

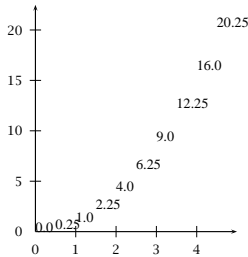
```


pst-qtreet: Interface for the package pst-tree



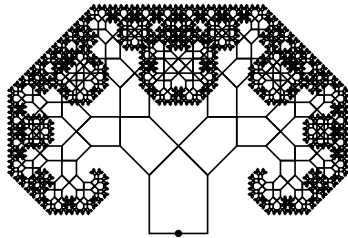
```
\usepackage{pst-qtreet}
\begin{Example}[ltxps]{\xLcs{pst-qtreet}{Tree}}
\Tree
[.S [.NP [.D the ] [.N cat ]
] [.VP [.V sat ] [.PP [.P on ]
[.NP [.D the ] [.N mat ] ] ] ] ] ]
```

infix-RPN: Converting an algebraic expression (infix) in a PostScript expression (postfix)



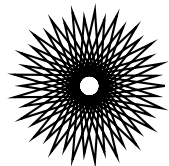
```
\usepackage{infix-RPN,pst-func,multido}
\SpecialCoor
\psset{yunit=0.25}
\begin{pspicture}(-0.25,-2)(5,22.5)
\infixtoRPN{x*x}
\multido{\rx=0.0+0.5}{10}{\rput(!/x \rx\space def
\RPN\space x exch ){\psPrintValue{\RPN}}}
\psaxes[dy=5,Dy=5]{->}(5,22.5)
\end{pspicture}
```

pst-fractal: Fractals



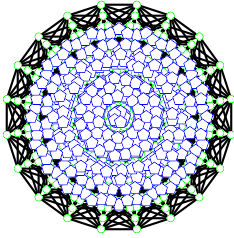
```
\usepackage{pst-fractal}
\begin{pspicture}(-3,0)(3,4)
\psPTree\psdot*(0,0)
\end{pspicture}
```

pst-poly: Polygons



```
\usepackage{pst-poly}
\PstPolygon[PolyNbSides=21, PolyOffset=2,
PolyIntermediatePoint=-0.9]
```

pst-coxeterp: Regular polytopes



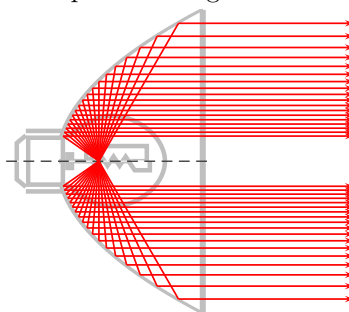
```
\usepackage{pst-coxeterp}
\begin{pspicture}(-2,-2)(2,2)
\psset{unit=0.4cm,colorCenters=blue,
styleCenters=pentagon,sizeCenters=0.2}
\gammappn[P=5,dimension=4]
\end{pspicture}
```

pst-lens: Lens magnification



```
\usepackage{pst-lens}
\def\Wishes{%
\rput[1b](0,0){%
\Large\begin{minipage}{3cm}
\centering
\textbf{Alles Gute}\\\ \textbf{Jana},\\
all the best for the year\\ \Huge 2010!
\end{minipage}}}\begin{pspicture}(0,-1.5)(3,4)
\Wishes\PstLens[LensMagnification=2](1.5,2.5){\Wishes}
\end{pspicture}
```

pst-optic: Two dimensional optical arrangement



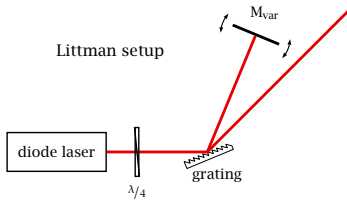
```
\usepackage{pst-optic}
\psset{unit=0.5}\begin{pspicture}(-1.5,-5.5)(10,5.5)
\rput(0,0){\beamLight[drawing=false,mirrorDepth=4.75,mirrorWidth=0.1,
mirrorHeight=10,linewidth=lightgray]}
\makeatletter \pst@getcoor{Focus}\pst@tempf \psset{linewidth=red}
\multido{\n=60+5}{18}{\mirrorCVGRay[linewidth=red,mirrorDepth=4.75,
mirrorHeight=10](Focus){! /XF \pst@tempf pop \pst@number\psxunit div def
```

```

\space cos XF add \space sin neg}{Endd1}
\psOutLine[arrows=->,length=.25](Endd1)(Endd1')}{Endd2}
\mirrorCVGRay[linecolor=red,mirrorDepth=4.75,mirrorHeight=10](Focus)%
(! /XF \pst@tempf pop \pst@number\psxunit div def
\space cos XF add \space sin )}{End1}
\psOutLine[arrows=->,length=.25](End1)(End1')}{End2}
\makeatother
\end{pspicture}

```

pst-optexp: Experimental optic

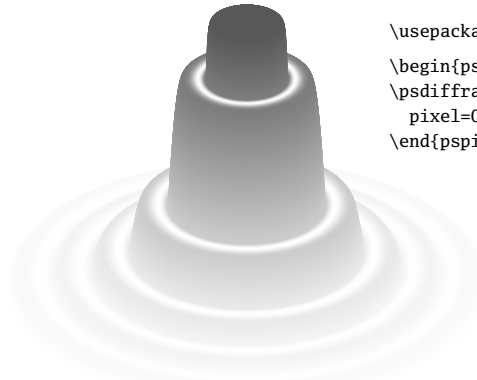


```

\usepackage{pst-optexp, nicefrac}
\begin{pspicture}(-4,-1)(3,3)
\addtopsstyle{Beam}{linewidth=2\pslinewidth,
linecolor=red!90!black}
\psset{labeloffset=0.5}
\node(-2,0){LaserOut}\node(0,0){Grat}
\node(4;45){Out}\node(2.5;67.5){Mvar}
\optbox[optboxwidth=2,labeloffset=0,
endbox](Grat)(LaserOut){diode laser}
\mirror[variable,conn=o-] %
(Grid)(Mvar)(Grid){M$_{\mathrm{var}}$}
\optgrid[beam](LaserOut)(Grat)(Out){grating}
\optretplate[position=0.3,labeloffset=0.8] %
(LaserOut)(Grat){$\nicefrac{\lambda}{4}$}
\rput[1](-3,2){Littman setup}
\end{pspicture}

```

pst-diffraction: Generating of a diffraction

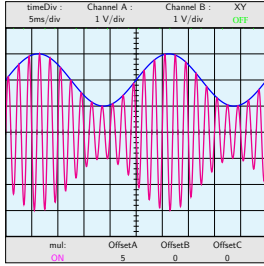


```

\usepackage{pst-diffraction}
\begin{pspicture}(-3.5,-1.5)(3.5,3.5)
\psdiffractionCircular[IIID,r=0.5e-3,f=10,
pixel=0.5,lambda=520,colorMode=0]
\end{pspicture}

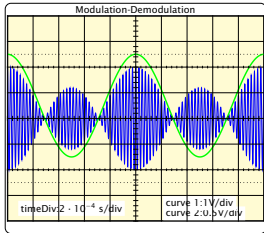
```

pst-osci: Simulating the output of an oscilloscope



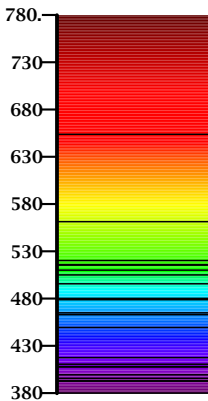
```
\usepackage{pst-osci}
\Oscillo[amplitude1=1,amplitude2=1,CC2=2,period2=25,
period1=2,combine=true,operation=mul,offset1=5]
```

pst-am: Modulation and demodulation



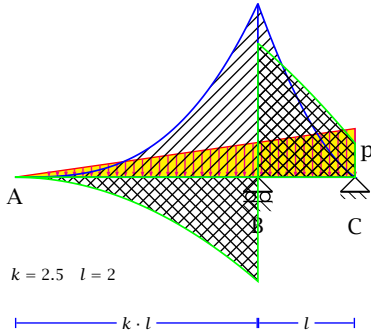
```
\usepackage{pst-am}
\psAM[SignalModulant,SignalModule,timeDiv=2e-4,U0=0.5,
frequencePorteuse=4e4,Up=4,Um=2,voltDivY2=0.5,values]
```

pst-spectra: Spectral lines



```
\usepackage{pst-spectra,pstricks-add}
\begin{pspicture}(0,-0.75)(2,4.75)
\rput{90}(1,0){\psspectrum[axe,Dl=50,wlangle=-90,
wlcmd={\scriptsize\bfseries},begin=780,end=380,
element=Es,absorption](5,2)(0,0)}
\end{pspicture}
```

pst-stru: Structural schemes in civil engineering

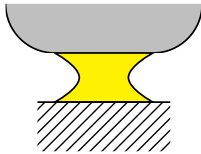


```

\usepackage{pst-stru}
\begin{pspicture}(-1,-3)(12,4)
\psset{arrowsize=0.8mm,
arrowsinset=0}
\trilead[K=2.5,P=8,L=2]
\end{pspicture}

```

pst-pad: Attachment models

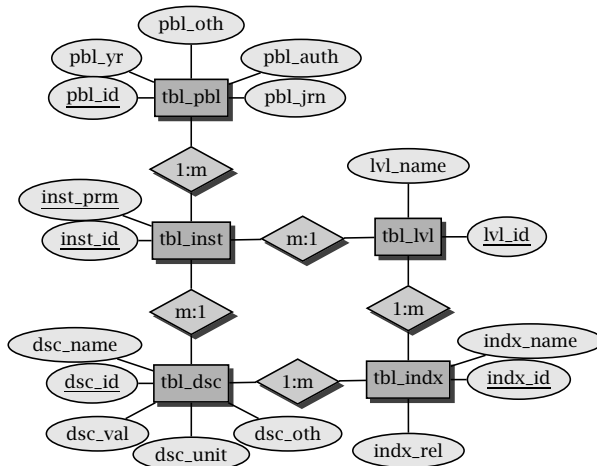


```

\usepackage{pst-pad}
\begin{Example}[ltxps]{\xLcs{pst-pad}{PstPad}}
\begin{pspicture}(4,4)
\PstPad(2,2)
\end{pspicture}
\end{Example}

```

pst-dbicons: Entity-Relationship diagrams

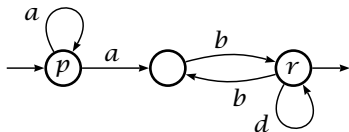


```

\usepackage{pst-dbicons}
\seticonparams{entity}{shadow=true,fillcolor=black!30,fillstyle=solid}
\seticonparams{attribute}{fillcolor=black!10,fillstyle=solid}
\seticonparams{relationship}{shadow=true,fillcolor=black!20,fillstyle=solid}
\begin{tabular}{cc}
\begin{tabular}{c}
\entity{tblpbl}[tbl\_pbl]\\[2cm]
\entity{tblinst}[tbl\_inst]\\[2cm]
\entity{tbldsc}[tbl\_dsc] \\[2cm]
\end{tabular}\hspace{6em}
\begin{tabular}{c}~\\[2cm]
\entity{tbllvl}[tbl\_lvl]\\[2cm]
\entity{tblindx}[tbl\_indx]\\[2cm]
\end{tabular}
\attributeof{tblpbl}[3em]{0}{pbljrn}[pbl\_jrn]
\attributeof{tblpbl}[3em]{90}{pblloth}[pbl\_oth]
\attributeof{tblpbl}[3em]{30}{pblauth}[pbl\_auth]
\attributeof{tblpbl}[3em]{150}{pblyr}[pbl\_yr]
\attributeof{tblpbl}[3em]{180}{key}{pblid}[pbl\_id]
\attributeof{tblinst}[3em]{150}{key}{instprm}[inst\_prm]
\attributeof{tblinst}[3em]{180}{key}{instid}[inst\_id]
\attributeof{tbldsc}[3em]{180}{key}{dscid}[dsc\_id]
\attributeof{tbldsc}[3em]{150}{dscname}[dsc\_name]
\attributeof{tbldsc}[3em]{220}{dscval}[dsc\_val]
\attributeof{tbldsc}[3em]{270}{dscunit}[dsc\_unit]
\attributeof{tbldsc}[3em]{320}{dscoth}[dsc\_oth]
\attributeof{tbllvl}[3em]{0}{key}{lvlid}[lvl\_id]
\attributeof{tbllvl}[3em]{90}{lvlname}[lvl\_name]
\attributeof{tblindx}[3em]{0}{key}{indxid}[indx\_id]
\attributeof{tblindx}[3em]{30}{indxname}[indx\_name]
\attributeof{tblindx}[3em]{270}{indxrel}[indx\_rel]
\relationshipbetween{tblpbl}{tblinst}{1:m}%relationships
\relationshipbetween{tblinst}{tbldsc}{m:1}
\relationshipbetween{tblinst}{tbllvl}{m:1}
\relationshipbetween{tbldsc}{tblindx}{1:m}
\relationshipbetween{tbllvl}{tblindx}{1:m}
\end{tabular}

```

pst-vaucanson-g: Drawing automata and graphs

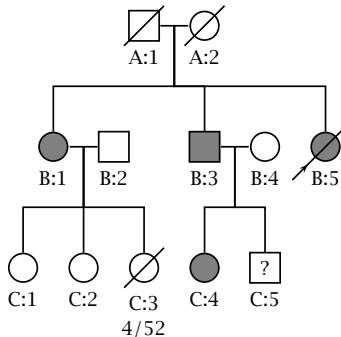


```

\usepackage{vaucanson-g}
\begin{VCPicture}{(0,-2)(5.5,2)}
\State[p]{(0,0)}{A} \State{(2.5,0)}{B}
\State[r]{(5.5,0)}{C}\Initial{A} \Final{C}
\EdgeL{A}{B}{a} \ArcL{B}{C}{b}
\ArcL{C}{B}{b} \LoopN{A}{a} \LoopS{C}{d}
\end{VCPicture}

```

pst-pdgr: “Pedegrees” and medical structures



```

\usepackage{pst-pdgr}
\begin{pspicture}(6,6) \psset{belowextrp=t,armB=1}
\rput(2.5,5.5){\pstPerson[male,deceased,belowtext=A:1]{A:1}}
\rput(3.5,5.5){\pstPerson[female,deceased,belowtext=A:2]{A:2}}
\pstRelationship[descentnode=A:1_2]{A:1}{A:2}
\rput(1,3.5){\pstPerson[female,affected,belowtext=B:1]{B:1}}
\pstDescent{A:1_2}{B:1}\rput(2,3.5){\pstPerson[male,belowtext=B:2]{B:2}}
\pstRelationship[descentnode=B:1_2]{B:1}{B:2}
\rput(3.5,3.5){\pstPerson[male,affected,belowtext=B:3]{B:3}}
\pstDescent{A:1_2}{B:3}\rput(4.5,3.5){\pstPerson[female,belowtext=B:4]{B:4}}
\pstRelationship[descentnode=B:3_4]{B:3}{B:4}
\rput(5.5,3.5){\pstPerson[female,affected,deceased,proband,belowtext=B:5]{B:5}}
\pstDescent{A:1_2}{B:5}
\rput(0.5,1.5){\pstPerson[female,belowtext=C:1]{C:1}}\pstDescent{B:1_2}{C:1}
\rput(1.5,1.5){\pstPerson[female,belowtext=C:2]{C:2}}\pstDescent{B:1_2}{C:2}
\rput(2.5,1.5){\pstPerson[female,deceased,
belowtext={\tabular{c}{C:3\4/52\endtabular}}]{C:3}}
\pstDescent{B:1_2}{C:3}\rput(3.5,1.5){\pstPerson[female,affected,belowtext=C:4]{C:4}}
\pstDescent{B:3_4}{C:4}
\rput(4.5,1.5){\pstPerson[male,insidetext=?,belowtext=C:5]{C:5}}
\pstDescent{B:3_4}{C:5}
\end{pspicture}

```

pst-light3d: Three dimensional light effects

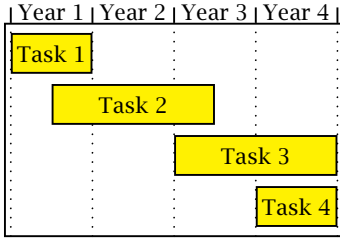


```

\usepackage{pst-light3d}
\DeclareFixedFont{\RM}{T1}{ptm}{m}{n}{2cm}
\psset{linestyle=none,fillstyle=solid,
fillcolor={rgb}{1,0.84,0}}
\PstLightThreeDText[LightThreeDXLength=0.5]{\RM\TeX}

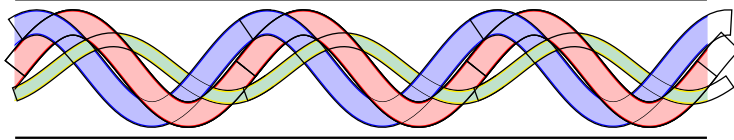
```

pst-gantt: Gantt charts



```
\usepackage{pst-gantt}
\begin{PstGanttChart}[yunit=1.5,
ChartUnitIntervalName=Year,
ChartUnitBasicIntervalName=Month,
TaskUnitIntervalValue=12,TaskUnitType=Year,
ChartShowIntervals]{4}{4}
\PstGanttTask[TaskInsideLabel={Task 1}]{0}{1}
\PstGanttTask[TaskInsideLabel={Task 2},
TaskUnitType=Month]{6}{24}% 24 mon start at 6
\PstGanttTask[TaskInsideLabel={Task 3}]{2}{2}
\PstGanttTask[TaskInsideLabel={Task 4}]{3}{1}
\end{PstGanttChart}
```

pst-thick: Very thick lines and curves



```
\usepackage{pst-thick}
\newsstyle{thicklinejaune}{fillstyle=solid,fillcolor=yellow!50!cyan!50,
linecolor=yellow,plotpoints=360}
\newsstyle{thicklinevert}{fillstyle=solid,fillcolor=green!50,
linecolor=green,plotpoints=360}
\newsstyle{onlycurvejaune}{linecolor=yellow,plotpoints=360}
\def\SinusPhase#1#2#3{%
/P #1 def /A #2 def /F #3 DegtoRad def
/O 360 P div def /x0 t def
/y0 t F add 0 mul sin A mul def % A*sin(0*t)
/dx dt def /dy t F add dt add 0 mul sin
t F add 0 mul sin sub A mul def }

\psset{unit=0.5}
\begin{pspicture}(0,-4)(30,4)
\def\motif{\psclip{\psframe[linestyle=none,dimen=inner](0,-3)(10,3)}
\psthick[stylethick=thicklineblue]{-1}{11}{\SinusPhase{10}{2}{90}}
\psthick[stylethick=thicklinejaune,E=0.5]{-1}{11}{\SinusPhase{10}{1.25}{-100}}
\psthick{-1}{11}{\SinusPhase{10}{2}{0}}
\psclip{\psthick[stylethick=vide,E=1.1]{-1}{11}{\SinusPhase{10}{2}{0}}
\psthick[stylethick=thicklineblue]{0}{3}{\SinusPhase{10}{2}{90}}
\endpsclip%
\psclip{\psthick[stylethick=vide,E=1.1]{0}{11}{\SinusPhase{10}{2}{90}}
\psthick{5}{9}{\SinusPhase{10}{2}{0}}
\endpsclip%
\psclip{\psthick[stylethick=vide,E=0.6]{0}{11}{\SinusPhase{10}{1.25}{-100}}
\psthick[stylethick=thicklineblue]{7}{9}{\SinusPhase{10}{2}{90}}
```



```

\endpsclip%
\psclip{\pstthick[stylethick=vide,E=1.1]{0}{10}{\SinusPhase{10}{2}{0}}}
  \pstthick[stylethick=thicklinejaune,E=0.5]{7}{11}{\SinusPhase{10}{1.25}{-100}}
\endpsclip%
\psclip{\pstthick[stylethick=vide,E=1.1]{0}{11}{\SinusPhase{10}{2}{0}}}
  \pstthick[stylethick=thicklinejaune,E=0.5]{-0.5}{1}{\SinusPhase{10}{1.25}{-100}}
\endpsclip%
\endpsclip}%
\motif\rput(10,0){\motif}\rput(20,0){\motif}
\psline[linewidth=0.1](0,3)(30,3)\psline[linewidth=0.1](0,-3)(30,-3)
\end{pspicture}

```

pst-mirror: Projections on a spherical mirror

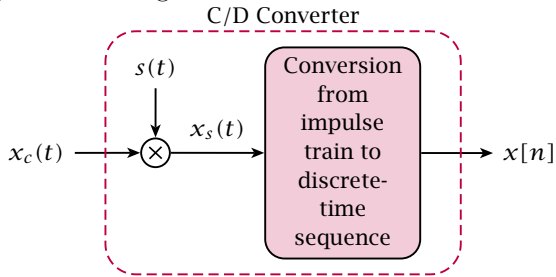


```

\usepackage{pst-mirror,pst-grad}
\begin{Example*}[ltxps]{ }
\psscalebox{0.7}{\begin{pspicture}(-7,-7)(7,7)
  \newsstyle{GradWhiteYellow}{fillstyle=gradient,gradbegin=yellow,gradend=yellow!20,
  linecolor=yellow!50,GradientCircle=true,gradmidpoint=0,GradientPos={(1,1)}}%
  \pscicle[style=GradWhiteYellow]{7.07}
  \pstSphereGrid[linecolor=red,grille=10,Ymin=-50,Ymax=50,Xmax=80,Xmin=-80,
  normale=0 0](20,0,0)
  \pstSphereGrid[linecolor=blue,grille=10,Ymin=-40,Ymax=-20,Xmax=80,Xmin=-80,
  normale=0 90](40,0,-10)%
  \pstTextSphere[fillstyle=solid,fillcolor=red,normale=0 0,fontscale=40,
  PSfont=Time-Roman,y0=0](20,0,10){pst-mirror}
  \pstTextSphere[fillstyle=solid,fillcolor=black,normale=0 0,fontscale=20,
  PSfont=Helvetica,y0=0](20,0,35){PSTricks}
  \pstTextSphere[fillstyle=solid,fillcolor=blue,normale=0 90,fontscale=10,
  PSfont=Helvetica,y0=2.5](10,0,-10){A Spherical Mirror}%
\end{pspicture}}

```

pst-sigsys: Signal Processing

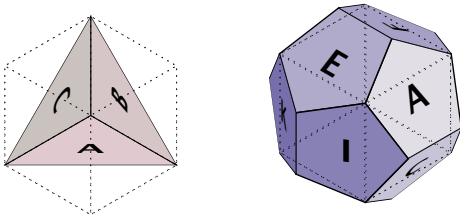


```

\usepackage{pstricks} \usepackage[pstadd]{pst-sigsys}
\begin{pspicture}(-2,-2)(5.5,2)
  \rput(-1.75,0){\rnode{xc}{x_c(t)}}
  \pscircloop[operation=times](0,0){otimes}\rput(0,1.25){\rnode{s}{s(t)}}
  \psblock[fillstyle=solid,fillcolor=purple!20](2.75,0){conv}{\parbox[c]{2\psunit}%
    {centering Conversion from impulse train to discrete-time sequence}}
  \rput(5.5,0){\rnode{x}{x[n]}} \psset{style=Arrow}
  \ncline[nodesepA=.15]{xc}{otimes} \ncline[nodesepA=.15]{s}{otimes}
  \ncline{otimes}{conv}\naput{x_s(t)}\ncline[nodesepB=.15]{conv}{x}
  \psframe[linecolor=purple,linestyle=dashed,style=Dash](-.75,-1.5)(4.5,1.5)
  \rput(1.875,1.75){C/D Converter}
\end{pspicture}

```

pst-platon: Platonic solids

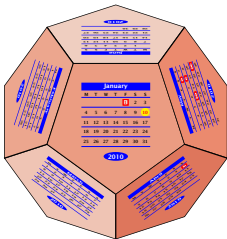


```

\usepackage{pst-platon}
\psTetrahedron\qqquad
\psDodecahedron

```

pst-calendar: Two or three dimensional calendars

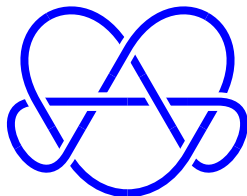


```

\usepackage{pst-calendar}
\begin{Example}[ltxps]{}
\psscalebox{0.13}{ %
  \psCalDodecaeder[Jahr=2010,style=march]}

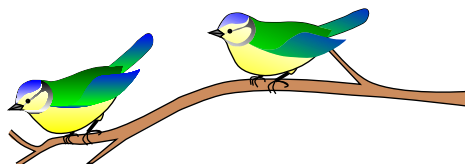
```

pst-knot: Knot lines



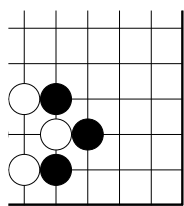
```
\usepackage{pst-knot}
\begin{pspicture}(-2,-2)(2,2)
\psKnot[linewidth=3pt,
linecolor=blue](0,0){7-7}
\end{pspicture}
```

pst-fun: “Funny objects”



```
\usepackage{pst-fun}
\begin{pspicture}(0,-1.3)(10,3)
\psBird[Branch]
\rput{-20}(4,1.8){\psBird}
\end{pspicture}
```

psgo: Game “Go”



D E F G H J

```
\usepackage{psgo}
\pscalebox{0.7}{%
\begin{psgopartialboard}[9]{{(4,1)(9,6)}
\stone{white}{c}{3} \stone{white}{e}{3}
\stone{white}{d}{2} \stone{white}{d}{4}
\stone{black}{f}{3} \stone{black}{e}{2}
\stone{black}{e}{4}
\end{psgopartialboard}}
```

PSTricks and PDF

Figure 2 on the facing page showed the different ways of generating PDF output from All \TeX source. With PSTricks, only the part in grey can be used to generate a PDF. Unless you are using the package `microtype` there will be no difference between a PDF generated in the PSTricks way with `latex` and one directly generated with `pdflatex`. When using a graphical user interface, e.g. Kile for Linux, \TeX Shop for MacOSX, or \TeX nicCenter for Windows, it is only one mouse click to generate the PDF output. The intermediate DVI and PostScript files are only temporaries, and can be deleted after generating the PDF.

The remaining sections describe several ways of using PSTricks-related code within a document whose final version will be compiled with `pdflatex`, which supports inclusion of images in PDF, PNG, and JPG formats, as well as the `microtype` package for optimized text formatting.

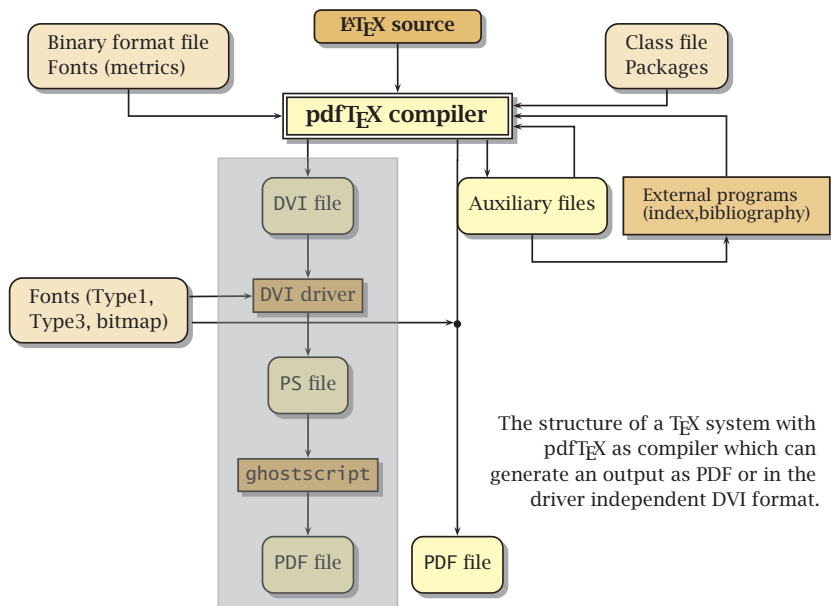


Figure 2: The shaded path shows how to generate PDF output when using PostScript code in the TeX source.

dvips and ps2pdf

If your PSTricks figure is created by a AllTeX file separate from your main document, say `fig.tex`, you can process it independently. First run `tex` or `latex` on `fig.tex` to create `fig.dvi`; then `dvips fig.dvi` to create `fig.ps`; and finally `ps2pdf fig.ps` (or another distiller program) to create `fig.pdf`. Then your main document can include `fig.pdf` like any PDF graphic.

pst2pdf

This is a Perl script to be used instead of the `pdflatex` command. This way of creating PDF output is the best choice when all graphics are needed as external images.

The script extracts all `pspicture` and `postscript` environments from the main text body and then runs these code snippets with the same preamble as the main document. The PDF output from each of these single documents is then cropped to get rid of the white space around the figure and also converted into EPS and (on Linux only) PNG formats.

After producing all PostScript-related code as a single image, saved in a default subdirectory `images/`, the script `pst2pdf` runs the source one last time with `pdflatex` and replaces all PostScript code with the formerly created image.

The script has several optional arguments which are described with their defaults at the beginning of the script.

pst-pdf and ps4pdf

This package from Rolf Niepraschk allows the cutting of the `pspicture` or `postscript` environments from the created DVI file into a new file `*-pics.ps`, which then is converted into a file `*-pics.pdf`. Every image will be on one page and the size of the image is taken from the `pspicture` coordinates or from the bounding box for a `postscript` environment. In a last `pdflatex` run the PDF images are inserted instead of the PostScript related code. There are four steps needed:

1. `latex FILE`
2. `dvips -Ppdf -o FILE-pics.ps FILE.dvi`
3. `ps2pdf -dAutoRotatePages=/None \`
`FILE-pics.ps FILE-pics.pdf`
4. `pdflatex FILE`

Alternatively, one can use the script `ps4pdf` to perform these steps. The script is part of any $\text{T}_{\text{E}}\text{X}$ distribution, and also available on CTAN. There are also some profiles for use from GUI programs, also available on CTAN (<http://mirror.ctan.org/graphics/pstricks/pst-support/>).

auto-pst-pdf

This package from Will Robertson works in the same way as `pst-pdf`, but it doesn't need a script or the four runs by the user. Everything is done in a single `pdflatex` run, and therefore you must allow execution of external programs from within `pdflatex`: the `shell-escape` option for $\text{T}_{\text{E}}\text{X}$ Live or `enable-write18` for $\text{M}\text{i}\text{K}\text{T}_{\text{E}}\text{X}$. Some GUI profiles are available from CTAN (<http://mirror.ctan.org/graphics/pstricks/pst-support/>).

pdftricks

This package from Radhakrishnan CV, Rajagopal CV, and Antoine Chambert-Loir was the first one to support PostScript related code with `pdflatex`. It works in a similar way as `pst-pdf`, but it needs additional code in the preamble to separate the PostScript part from the PDF part. More information available from CTAN, or by running `texdoc pdftricks`.

Conclusion

In this article we have introduced the current state of the PSTricks package, its new extensions with the accompanied examples in a form of colourful figures next to their full source code. The article has also described the use of PSTricks friends, such as `pst2pdf`, `pst-pdf`, `ps4pdf`, `auto-pst-pdf`, and `pdftricks`.

References

- [1] Goossens, Michel; Mittelbach, Frank; Rahtz, Sebastian; Roegel, Denis; and Voß, Herbert. *The L^AT_EX Graphics Companion*. Tools and Techniques for Computer Typesetting, 2nd edition. USA: Addison-Wesley Publishing Company, Boston, 2006. 976 pages. ISBN 978-0321508928.
- [2] Voß, Herbert. *PSTricks – Grafik für T_EX und L^AT_EX*. 5th edition. Germany: DANTE – Lehmanns, Heidelberg/Hamburg, 2008. 912 pages. ISBN 3865412805.

Summary: The current state of the PSTricks project

PSTricks is an abbreviation for PostScript Tricks and uses the enormous graphical capabilities of the *old* programming language PostScript. It is a so-called page code language (PCL) which is distributed since 1984 by Adobe Systems. PostScript is fully Turing compatible and a stack-oriented programming language, like Assembler, Forth, Hewlett Packard pocket calculators, et al. PDF is derived from PostScript, with some important extensions but without the possibility of mathematical calculations.

Keywords: PSTricks package, PostScript, PDF.

*Herbert Voß, herbert@dante.de
Wasgenstraße 21, Berlin
D-14129, Germany*