The `\texttt{pst-gr3d}` package
A PSTricks package for three dimensional grids

Denis GIROU*

Version 1.3
February 19, 2001
Documentation revised February 19, 2001

Abstract
This package allow to draw three dimensional grids using the macro \texttt{\textbackslash PstGridThreeD}. We can also specify how nodes of the grid must look like.

Contents

1 Introduction ........................................ 1
2 Usage .............................................. 2
   2.1 Parameters and hooks ........................... 2
3 Examples .......................................... 7

1 Introduction

`\texttt{pst-gr3d}` offer a main unique macro with few parameters to interact on it. But we can also use all the relevant PSTricks parameters to change the size, the characteristics of lines, etc.

The syntax is simply: \texttt{\textbackslash PstGridThreeD[optional\_parameters](X,Y,Z)}

We can define a macro \texttt{\textbackslash PstGridThreeD\textbackslash HookNode} to specify how the nodes at the interconnections must look like, and there are also some other hooks that can be used for special purposes.

The default viewpoint is \texttt{(1.2,-0.6,0.8)}, but this can of course be changed using the standard way.

The package try to compute approximatively the size of the object (the \texttt{pspicture} parameter, PSTricks speaking), but for three dimensional grids it is an impossible task to found it accurately in the general case. So, if the exact size is needed or if we change the viewpoint for the graphic, the size must be computed by hand, using the \texttt{\textbackslash psframebox[framesep=0]{\ldots}} construction to found the correct values by attempts and errors — fortunately, in practice few attempts are often enough...

*CNRS/IDRIS — Centre National de la Recherche Scientifique / Institut du Développement et des Ressources en Informatique Scientifique — Orsay — France — <Denis.Girou@idris.fr>.
2 Usage

2.1 Parameters and hooks

The three required parameters specify the lengths in the X, Y and Z directions, respectively:

\begin{verbatim}
\PstGridThreeD(3,1,1)\hfill
\PstGridThreeD(1,3,1)\hfill
\PstGridThreeD(1,1,3)
\end{verbatim}

Of course, we can use all the relevant generic PSTricks parameters, specially those applying to grids:

\begin{verbatim}
\PstGridThreeD[unit=1.5](1,1,1)\hfill
\PstGridThreeD[viewpoint=1.2 -1.5 0.4,griddots=7](1,3,2)\hfill
\PstGridThreeD[gridwidth=0.08,gridcolor=red](3,2,2)\hfill
\begin{pspicture}(-1.7,0)(0.8,3.6)
\PstGridThreeD[viewpoint=-0.4 -0.6 0.8,PstPicture=false](1,3,2)
\end{pspicture}
\end{verbatim}

We can draw one and two dimensional grids, using degenerated cases:

\begin{verbatim}
\PstGridThreeD(0,4,0)\hfill
\PstGridThreeD[linewidth=0.05](0,3,1)\hfill
\PstGridThreeD[griddots=5](3,1,0)
\end{verbatim}
To change the way the grids are drawn, we can also use nine specific parameters and five specific hooks:

**PstDebug (integer)**: to obtain some internal debugging informations — here, a framed box around the boundbox box used (the `pspicture` environment) could be drawn. It can take the values 0 (no debug) or 1. (*Default: 0 — no debugging informations*).

**PstPicture (boolean)**: to define or not a `pspicture` environment for the grid. We have to define this parameter to `false` mainly if we choose a viewpoint different than the default one — see examples later (*Default: true — which is not the case for basic PSTricks objects*).

**GridThreeDXUnit (integer)**: unit coefficient in the X direction (*Default: 1 — it must be an integer, not a real*).

**GridThreeDYUnit (integer)**: unit coefficient in the Y direction (*Default: 1 — it must be an integer, not a real*).

**GridThreeDZUnit (integer)**: unit coefficient in the Z direction (*Default: 1 — it must be an integer, not a real*).

To change the way the grids are drawn, we can also use LaTeX code:

```latex
\texttt{\begin{verbatim}
\PstGridThreeD[GridThreeDXUnit=2](1,1,1)\hfill
\PstGridThreeD[GridThreeDYUnit=3](1,1,1)\hfill
\PstGridThreeD[unit=0.5,GridThreeDZUnit=4](4,3,1)
\end{verbatim}}
```

**GridThreeDXPos (integer)**: position of the origin in the X direction (*Default: 0 — it must be an integer, not a real*).

**GridThreeDYPos (integer)**: position of the origin in the Y direction (*Default: 0 — it must be an integer, not a real*).

**GridThreeDZPos (integer)**: position of the origin in the Z direction (*Default: 0 — it must be an integer, not a real*).
These parameters are in fact mainly useful if we want to superpose grids, which can be done easily using the \pstGridThreeDHookEnd macro (see description below):

\begin{verbatim}
\% First grid
\def\pstGridThreeDHookEnd{%\}
\pstGridThreeD[\PstPicture=false,gridwidth=0.1, GridThreeDXPos=1](0,2,1)}\hfill
\pstGridThreeD(1,3,2)\hfill
\% Second grid
\def\pstGridThreeDHookEnd{%\}
\pstGridThreeD[\PstPicture=false,gridwidth=0.1, GridThreeDYPos=1](1,2,1)}\hfill
\pstGridThreeD(1,3,2)\hfill
\% Third grid
\def\pstGridThreeDHookEnd{%\}
\pstGridThreeD[\PstPicture=false,gridwidth=0.1, gridcolor=green, GridThreeDYPos=2, GridThreeDZPos=1](1,1,1)}\hfill
\pstGridThreeD(1,3,2)
\end{verbatim}

GridThreeDNodes (boolean) : to define or not the nodes at interconnection points of the grid. The nodes are named Gr3dNodeXYZ. We can use the Rx and Ry parameters to position the relevant material relatively to the nodes, specifying the distance in cartesian coordinates. The parameter angle used with Rx allow to use polar ones. (Default: false — no nodes defined).

\begin{verbatim}
\pstGridThreeD[GridThreeDNodes=\true](1,3,1)
\SpecialCoor
\rput*(Gr3dNode130){\footnotesize 130}
\rput*(Gr3dNode131){\footnotesize 131}
\end{verbatim}

\pstGridThreeDHookNode (macro) : this hook allow to define the form of the nodes. A predefined \pstGridThreeDNodeProcessor macro exist, which define a circle with a little white circle in it. We can also use the \iy counter to differentiate the nodes according to the Y faces — but note that we can’t do the same thing for the X or Z faces (Default: empty).
\PstGridThreeDHookEnd{macro} : this hook allow to execute a macro at the end of the grid drawing, before the \pspicture environment closing. This is specially interesting for instance to superpose grids, if we take care to define the \PstPicture parameter to false for them (Default: empty).
\PstGridThreeDHookXFace (macro) : this hook allow to execute a macro before
to draw the X faces (Default: empty).

\PstGridThreeDHookYFace (macro) : this hook allow to execute a macro before
to draw the Y faces (Default: empty).

\PstGridThreeDHookZFace (macro) : this hook allow to execute a macro before
to draw the Z faces (Default: empty).

In fact, these hooks are not very powerful, because we can’t control the order
of the faces drawing as we can dream... For instance, we can’t use this technic to
draw objects with only true visible lines. Take care also that for the Y faces, the
direction is negative in the horizontal direction, so the coordinates must take this
fact in account.

\def\PstGridThreeDHookXFace{%\ifnum\multidocount=1\psframe*[linecolor=cyan](3,2)\fi%\PstGridThreeD(1,3,2)\hfill%
\def\PstGridThreeDHookYFace{%\ifnum\multidocount=2\psframe*[linecolor=yellow](-3,0)(0,2)\fi%\PstGridThreeD(3,1,2)\hfill%
\def\PstGridThreeDHookZFace{%\ifnum\multidocount=2 \else\psframe*[linecolor=yellow](3,3)\fi%\PstGridThreeD(3,3,2)
3 Examples

We give here more advanced examples, most of them from technical drawings describing the architecture of a multiprocessors supercomputer.
\def\PstGridThreeDHookNode{% 
\PstGridThreeDNodeProcessor{Orange}}
\psset{unit=1.3}
\PstGridThreeD[GridThreeDNodes=true](1,2,2)
\SpecialCoor
\psset{arrows=->, arrowscale=2}
\ThreeDput[normal=0 0 -1](0,0,0){%
\ncloop[linecolor=red, arm=0.35, loopsize=0.6, angleA=-90, angleB=90]{Gr3dNode022}{Gr3dNode002}
\ncloop[linecolor=green, arm=0.7, nodesepA=0.18, nodesepB=0.12, loopsize=-0.5, angleA=180]{Gr3dNode002}{Gr3dNode102}}
\def\PstGridThreeDHookEnd{% 
\psset{PstPicture=false, gridwidth=0.1}
{\def\PstGridThreeDHookNode{% 
PstGridThreeDNodeProcessor{blue}}%
PstGridThreeD[gridcolor=blue, GridThreeDZPos=3](0,7,0)}%
{\def\PstGridThreeDHookNode{% 
PstGridThreeDNodeProcessor{red}}%
PstGridThreeD[gridcolor=red, GridThreeDXPos=1, GridThreeDZPos=1](0,3,1)}%
{\def\PstGridThreeDHookNode{% 
PstGridThreeDNodeProcessor{green}}%
PstGridThreeD[gridcolor=green, GridThreeDYPos=6](1,1,1)}}
PstGridThreeD[gridwidth=0.04, GridThreeDNodes=true](1,7,3)
\SpecialCoor
\rput([Rx=0.15, angle=140]Gr3dNode033){% 
\psline[linecolor=blue]{<-}(0.8;150)}
\rput([Rx=0.95, angle=140]Gr3dNode033){% 
\shortstack{1d grid \\footnotesize (X=8, Y=1, Z=1)}}
\rput([Rx=0.15, angle=-50]Gr3dNode121){% 
\psline[linecolor=red]{<-}(1.2;-50)}
\rput([Rx=1.5, angle=-55]Gr3dNode121){% 
\shortstack{2d grid \\footnotesize (X=4, Y=2, Z=1)}}
\rput([Rx=0.15, angle=-50]Gr3dNode121){% 
\psline[linecolor=red]{<-}(1.2;-50)}
\rput([Rx=0.2, angle=-100]Gr3dNode160){% 
\psline[linecolor=green]{<-}(0.8;-100)}
\rput([Rx=1.4, angle=-100]Gr3dNode160){% 
\shortstack{3d grid \\footnotesize (X=2, Y=2, Z=2)}}