The gridpapers package

Robert McNees* and Leo C. Stein†

v1.0.2 from 2021/03/27

All development happens at the repo: https://github.com/mcnees/LaTeX-Graph-Paper.

1 Introduction

Make your own quadrille, graph, hex, etc. paper! Uses the PGF/TikZ package for LaTeX, which should be part of any modern TeX installation. Lots of preset defaults to get started with ease, yet all colors and spacing are customizable.

There are example .tex files in the `examples` directory to help get you started with customization. Each tex file has an almost-empty body, with a `\usepackage` statement that you can customize. For example, engineer-pad.tex looks like this:

engineer-pad.tex

```
\documentclass{article}

\usepackage[pattern=majmin, colorset=engineer]{gridpapers}

\begin{document}
\thispagestyle{empty}
~
\end{document}
```

(The ~ in the body forces a non-empty body, or else latex wouldn't generate a PDF).

2 Usage

2.1 Options

Your graph paper is configured through a number of key/value options to the `\usepackage` command. Let's go through these options.

`pattern={⟨name⟩}`
Default: `std`

Which of the predefined patterns to use for the page or textarea background. The current list of pattern names is: `std`, `stdeight`, `majmin`, `dot`, `hex`, `hexup`, `tri`, `iso`, `lightcone`, `ruled`, `doubleruled`. We describe each of these patterns in Sec. 2.2. Patterns come with

*⟨rmcnees@luc.edu⟩; @mcnees
†⟨leo.stein@gmail.com⟩; @duetosymmetry

default page geometry (size and margins; see `geometry`), and default 'fullness' (whether they fill the page or not; see options `fullpage` and `textarea`).

`colorset={`⟨*name*⟩`}` Default: `std`
: Color presets. Valid color preset names are: `std`, `precocious`, `ghostly`, `brickred`, `engineer`, `plumpad`. A preset determines the `majorcolor`, `minorcolor`, and `bgcolor` all at once. But, you can start from a preset and then override some colors.

`majorcolor={`⟨*color*⟩`}`
: Override the preset "major" color. This can be a named color, or using the syntax from `xcolor` to mix colors together.

`minorcolor={`⟨*color*⟩`}`
: Override the preset "minor" color. As above.

`bgcolor={`⟨*color*⟩`}`
: Override the preset background color. As above.

`patternsize={`⟨*length*⟩`}`
: Override the preset pattern size. The meaning of this length argument is different for each pattern; see Sec. 2.2 for more.

`dotsize={`⟨*length*⟩`}` Default: `.7pt`
: Controls the size of the dots themselves for `pattern=dot`.

`fullpage`
: Make the pattern fill the whole page.

`textarea`
: Make the pattern fill only the text area of the document. At most one of the `fullpage` or `textarea` can be specified. If one is specified, it will override the default 'fullness' setting of the pattern.

`geometry={`⟨*geometry spec*⟩`}`
: Page geometry specification, using the syntax of the `geometry` package. This specification will override the pattern's default page geometry. However, if the `geometry` package was loaded before `gridpapers`, this option will be ignored.

2.2 Patterns

The current set of patterns:

`std`
: Quadrille, ten squares per inch. The `patternsize` option controls the side of a square. Default is `patternsize=0.1in`.

`stdeight`
: Quadrille, eight squares per inch. The `patternsize` option controls the side of a square. Default is `patternsize=0.125in`.

`majmin`
: Graph paper, eight squares per inch with a major grid every half-inch. The `patternsize` option controls the side of a small square (the larger squares are four times bigger). Default is `patternsize=0.125in`.

`dot`
: Grid of dots. The size of an individual dot is set by `dotsize`. The `patternsize` option controls the distance between dots. Default is `patternsize=0.1in`.

`hex`
: Grid of hexagons. The `patternsize` option controls the side length of a hexagon. Default is `patternsize=0.1666in`.

`hexup`
: Grid of hexagons, rotated 90 degrees from the default orientation. The `patternsize` option controls the side length of a hexagon. Default is `patternsize=0.1666in`.

`tri`
: Triangle grid. The `patternsize` option controls the side length of a triangle. Default is `patternsize=0.25in`.

`iso`
: Isometric grid. The `patternsize` option controls the side length of a triangle. Default is `patternsize=0.25in`.

`lightcone` A grid with light cones (45° lines) on a square grid. The `patternsize` option controls the side of a horizontal square (not the tipped squared). Default is `patternsize=0.25in`.

`ruled` Ruled page with bold lines. The `patternsize` option controls the vertical distance between lines. Default is `patternsize=0.2in`.

`doubleruled` Ruled page with bold lines alternating with light lines. The `patternsize` option controls the vertical distance between neighboring lines. Default is `patternsize=0.125in`.

3 Examples

3.1 Plenty of customization

Let's say you want to use the `tri` pattern, which by default fills the page. But you want it to fill just the textarea of an A4 page with 2cm margins, and you want the triangles to be .75cm long. Finally, you like the colors of the engineer set, but want a white background. Then you would write:

```
\usepackage[pattern=tri,
  patternsize=0.75cm,
  textarea,
  colorset=engineer,
  bgcolor=white,
  geometry={a4paper, margin=2cm}]{gridpapers}
```

3.2 Custom colors

Using named or blended custom colors is demonstrated in the example file `custom-colors.tex`:

custom-colors.tex

```
\documentclass{article}

\usepackage{xcolor}

% See the documentation of the xcolor package to learn about different
% color models for specifying colors
\definecolor{mydeepgreen}{rgb}{0.07, 0.56, 0.04}

% You can easily mix colors by using the ! syntax from xcolor.  Here
% we use it to mix 40% of our color with 60% white.
\usepackage[pattern=majmin,
  majorcolor=mydeepgreen,
  minorcolor={mydeepgreen!40}]{gridpapers}

\begin{document}
\thispagestyle{empty}
~
\end{document}
```

4 Implementation

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{gridpapers}
    [2021/03/27 v1.0.2 Graph paper backgrounds]

\RequirePackage{xkeyval}
\RequirePackage{kvoptions}
\RequirePackage{xcolor}
\RequirePackage{tikz}
\usetikzlibrary{patterns.meta,calc}
\RequirePackage{tikzpagenodes}
%% everypage has been superseded -- try to use the new builtin
%% approach, but fall back to everypage-1x if needed
%% This code is roughly taken from the new everypage code
\@ifundefined{AddToHook}{%
  \IfFileExists{everypage-1x.sty}{%
    %% If everypage is new enough to complain, avoid the complaints
    \RequirePackage{everypage-1x}
  }{\RequirePackage{everypage}}
}{%
  \newcommand*{\AddEverypageHook}[1]{%
  \AddToHook{shipout/background}{\put(1in,-1in){#1}}}
}
\RequirePackage{pagecolor}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Option parsing
%% Declare switches for processing the options.

\newif\ifGP@geometrypreviouslyloaded
\newif\ifGP@fullnessset
\newif\ifGP@fullpage
\newif\ifGP@textarea
\GP@geometrypreviouslyloadedfalse
\GP@fullnesssetfalse
\GP@fullpagefalse
\GP@textareafalse

\SetupKeyvalOptions{%
  family=GP,%
  prefix=GPOpt@%
}

\DeclareStringOption[std]{pattern}
\DeclareStringOption[std]{colorset}

\DeclareStringOption{majorcolor}
\DeclareStringOption{minorcolor}
\DeclareStringOption{bgcolor}

\DeclareStringOption{patternsize}
\DeclareStringOption[.7pt]{dotsize}

\DeclareVoidOption{fullpage}{\GP@fullpagetrue}
\DeclareVoidOption{textarea}{\GP@textareatrue}

\DeclareStringOption{geometry}

```

```
\ProcessKeyvalOptions*

%% Can only have one of fullpage or textarea
\ifGP@fullpage
  \ifGP@textarea
    \PackageError{gridpapers}{%
      Can not specify both fullpage and textarea, please remove one option}{}
    \fi
  \GP@fullnesssettrue
\fi

\ifGP@textarea
  \GP@fullnesssettrue
\fi

%% We keep track of this to know whether or not we would be overriding
%% a previously-set page geometry
\@ifpackageloaded{geometry}
  {\GP@geometrypreviouslyloadedtrue}
  {\GP@geometrypreviouslyloadedfalse%
    \PassOptionsToPackage{\GPOpt@geometry}{geometry}%
    \RequirePackage{geometry}%
  }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Actual package code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Some nice colors.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\definecolor{plum}{rgb}{0.36078, 0.20784, 0.4}
\definecolor{chameleon}{rgb}{0.30588, 0.60392, 0.023529}
\definecolor{cornflower}{rgb}{0.12549, 0.29020, 0.52941}
\definecolor{scarlet}{rgb}{0.8, 0, 0}
\definecolor{brick}{rgb}{0.64314, 0, 0}
\definecolor{sunrise}{rgb}{0.80784, 0.36078, 0}
\definecolor{rosiebg}{RGB}{250,247,232}
\definecolor{rosiegrid}{RGB}{186,137,113}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% The color to use for the null directions when drawing lightcones.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\colorlet{lightlines}{scarlet!30}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Pre-defined Color schemes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Here are some pre-defined color schemes for the paper background
%% and the major and minor grid lines.  These are switched by using
%% the option colorset=<name>.  The allowed values for colorset are in
%% the list below.
\define@choicekey*{GP}{colorset}[\val\nr]%
  %% Allowed values for colorset:
  {std,precocious,ghostly,brickred,engineer,plumpad}[std]{%
  \ifcase\nr\relax
    %% std
    \colorlet{minorcolor}{cornflower!30}
    \colorlet{majorcolor}{cornflower!50}
```

```
    \colorlet{bgcolor}{white}
  \or
    %% precocious
    \colorlet{minorcolor}{rosiegrid!50}
    \colorlet{majorcolor}{rosiegrid}
    \colorlet{bgcolor}{rosiebg}
  \or
    %% ghostly
    \colorlet{minorcolor}{gray!15}
    \colorlet{majorcolor}{gray!20}
    \colorlet{bgcolor}{white}
  \or
    %% brickred
    \colorlet{minorcolor}{brick!35}
    \colorlet{majorcolor}{brick!60}
    \colorlet{bgcolor}{scarlet!8}
  \or
    %% engineer
    \colorlet{minorcolor}{chameleon!50}
    \colorlet{majorcolor}{chameleon!80}
    \colorlet{bgcolor}{chameleon!10}
  \or
    %% plumpad
    \colorlet{minorcolor}{cornflower!40}
    \colorlet{majorcolor}{cornflower!70}
    \colorlet{bgcolor}{plum!10}
  \fi
}

%% Get the specified color set from the options
\def\@setkeyhelper#1#2{%
  \setkeys{GP}{#2=#1}
}
\expandafter\@setkeyhelper\expandafter{\GPOpt@colorset}{colorset}

%% If the user further specified majorcolor, minorcolor, and/or
%% bgcolor, we now override the selected colorset
\ifx\GPOpt@majorcolor\@empty
\else
  \colorlet{majorcolor}{\GPOpt@majorcolor}
\fi
\ifx\GPOpt@minorcolor\@empty
\else
  \colorlet{minorcolor}{\GPOpt@minorcolor}
\fi
\ifx\GPOpt@bgcolor\@empty
\else
  \colorlet{bgcolor}{\GPOpt@bgcolor}
\fi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% The size parameter -- different meanings for different patterns
%% Will be reset by pattern code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\GP@patternsize}{0.1in}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This section sets up a routine for filling a shape with
%% hexagons. Uses code from:
```

```
%% http://tex.stackexchange.com/questions/6019/drawing-hexagons/6128#6128
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% We have to delay this definition until after \GP@patternsize is
%% redefined (by the pattern selection and/or user override)
\newcommand{\GP@declarehexpat}{
\tikzdeclarepattern{
  name=hexagons,
  type=uncolored,
  bounding box={(0,0) and (3*\GP@patternsize,0.866025*2*\GP@patternsize)},
  tile size={(3*\GP@patternsize,0.866025*2*\GP@patternsize)},
  parameters={\tikzhexrotate},
  tile transformation={rotate=\tikzhexrotate},
  defaults={
    rotate/.store in=\tikzhexrotate,rotate=0,
  },
  code={
      \pgfsetlinewidth{0.6pt}
      \pgftransformshift{\pgfpoint{0mm}{0.866025*\GP@patternsize}}
      \pgfpathmoveto{\pgfpoint{0mm}{0mm}}
      \pgfpathlineto{\pgfpoint{0.5*\GP@patternsize}{0mm}}
      \pgfpathlineto{\pgfpoint{\GP@patternsize}{-0.866025*\GP@patternsize}}
      \pgfpathlineto{\pgfpoint{2*\GP@patternsize}{-0.866025*\GP@patternsize}}
      \pgfpathlineto{\pgfpoint{2.5*\GP@patternsize}{0mm}}
      \pgfpathlineto{\pgfpoint{3*\GP@patternsize}{0mm}}
      \pgfpathmoveto{\pgfpoint{0.5*\GP@patternsize}{0mm}}
      \pgfpathlineto{\pgfpoint{\GP@patternsize}{0.866025*\GP@patternsize}}
      \pgfpathlineto{\pgfpoint{2*\GP@patternsize}{0.866025*\GP@patternsize}}
      \pgfpathlineto{\pgfpoint{2.5*\GP@patternsize}{0mm}}
      \pgfusepath{stroke}
    }
  }
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This section sets up a routine for filling a shape with
%% triangles.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% We have to delay this definition until after \GP@patternsize is
%% redefined (by the pattern selection and/or user override)
\newcommand{\GP@declaretripat}{
\tikzdeclarepattern{
  name=triangles,
  type=uncolored,
  bounding box={(0,0) and (\GP@patternsize,2*0.866025*\GP@patternsize)},
  tile size={(\GP@patternsize,2*0.866025*\GP@patternsize)},
  parameters={\tikztrirotate},
  tile transformation={rotate=\tikztrirotate},
  defaults={
    rotate/.store in=\tikztrirotate,rotate=0,
    },
  code={
        \pgfsetlinewidth{0.6pt}
        \pgfpathmoveto{\pgfpoint{0mm}{0mm}}
        \pgfpathlineto{\pgfpoint{\GP@patternsize}{2*0.8660254*\GP@patternsize}}
        \pgfpathlineto{\pgfpoint{0mm}{2*0.8660254*\GP@patternsize}}
        \pgfpathmoveto{\pgfpoint{0mm}{0.8660254*\GP@patternsize}}
```

```
        \pgfpathlineto{\pgfpoint{\GP@patternsize}{0.8660254*\GP@patternsize}}
        \pgfpathmoveto{\pgfpoint{0mm}{2*0.8660254*\GP@patternsize}}
        \pgfpathlineto{\pgfpoint{\GP@patternsize}{0mm}}
        \pgfpathlineto{\pgfpoint{0mm}{0mm}}
        \pgfusepath{stroke}
    }
  }
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This section sets up a routine for filling the squares in a
%% grid with null lines.
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% TODO Still can't figure out the correct pattern shift!!
\newcommand{\GP@declarelightconepat}{
\pgfkeys{
  /pgf/pattern keys/myshift/.store in=\myshift,
  /pgf/pattern keys/myshift/.initial={(0,0)},
}
\tikzdeclarepattern{
  name=lightcones,
  type=uncolored,
  parameters={\myshift},
  bounding box={(0,0) and (\GP@patternsize,\GP@patternsize)},
  tile size={(\GP@patternsize, \GP@patternsize)},
  tile transformation={
    shift=\myshift,
  },
  defaults={
    myshift/.store in=\myshift,myshift={(0,0)},
  },
  code={
    %% TODO Make the dashing an option
    \tikzset{lightlines/.style={line width=0.4pt,dash=on 0.05cm off 0.05cm phase 0.025cm}}
    \draw [lightlines] (0,0) -- (\GP@patternsize,\GP@patternsize);
    \draw [lightlines] (0,\GP@patternsize) -- (\GP@patternsize,0);
  },
}
}
%% \pgfdeclarepatternformonly
%%   {lightcones}% name
%%   {\pgfpointorigin}% lower left
%%   {\pgfpoint{\GP@patternsize}{\GP@patternsize}}%  upper right
%%   {\pgfpoint{\GP@patternsize}{\GP@patternsize}}%  tile size
%%   {% shape description
%%     \pgfsetlinewidth{0.4pt}
%%     %% TODO Make an option
%%     %Comment out this line for solid lines on light cones, instead of dashes.
%%     \pgfsetdash{{0.05cm}{0.05cm}}{0.025cm}
%%     \pgfpathmoveto{\pgfpoint{0in}{0in}}
%%     \pgfpathlineto{\pgfpoint{\GP@patternsize}{\GP@patternsize}}
%%     \pgfpathmoveto{\pgfpoint{0in}{\GP@patternsize}}
%%     \pgfpathlineto{\pgfpoint{\GP@patternsize}{0in}}
%%     \pgfusepath{stroke}
%%   }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This section sets up a routine for filling a region with dots
%% Slightly modified version of code added by Leo
```

```
%% Stein (@duetosymmetry on Twitter).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% We have to delay this definition until after \GP@patternsize is
%% redefined (by the pattern selection and/or user override)
\newcommand{\GP@declaredotpat}{
\pgfdeclarepatternformonly
  {dotgrid}%% name
  {\pgfpoint{-0.5*\GP@patternsize}{-0.5*\GP@patternsize}}%% lower left
  {\pgfpoint{0.5*\GP@patternsize}{0.5*\GP@patternsize}}%%  upper right
  {\pgfpoint{\GP@patternsize}{\GP@patternsize}}%%  tile size
  {%% shape description
    \pgfpathcircle{\pgfqpoint{0pt}{0pt}}{\GP@dotsize}
    \pgfusepath{fill}
  }
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Begin pattern execution infrastructure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% This inner code will be set by the choicekey pattern=...
\newcommand{\GP@innerpatterncode}{}
%% This is the "outer" code to hook into every page
\newcommand{\GP@patterncode}{% No blank lines in this code!
\begin{tikzpicture}[remember picture, overlay]
%%
%% Change "thin" to "very thin" if the lines are too thick.
\tikzset{
  minorgrid/.style={minorcolor, thin},
  majorgrid/.style={majorcolor, thin},
}
\ifGP@fullpage%
\coordinate (a) at (current page.south west);
\coordinate (b) at (current page.north east);
\else%
\coordinate (a) at (current page text area.south west);
\coordinate (b) at (current page text area.north east);
\fi
%%
\GP@innerpatterncode%
%%
\end{tikzpicture}
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Begin pattern definition code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\define@boolkey{GP}{patterndefaultfullness}{}
\newcommand{\GP@patterndefaultgeometry}{}
\newcommand{\GP@patterndefaultsize}{}

%% Pattern-definer-helper
%% The interface is:
%% \GP@setpattern
%%   {<true for default fullpage, false for default textarea>}
%%   {<default geometry config>}
%%   {<default pattern size>}  %% NOTE, not tile length
%%   {<contents of inner pattern code>}
```

```
\newcommand{\GP@setpattern}[4]{%
\setkeys{GP}{patterndefaultfullness=#1}
\renewcommand{\GP@patterndefaultgeometry}{#2}
\renewcommand{\GP@patterndefaultsize}{#3}
\renewcommand{\GP@innerpatterncode}{#4}
}

\define@choicekey*{GP}{pattern}[\val\nr]%
  %% Allowed values for pattern:
  {std,stdeight,majmin,dot,hex,hexup,tri,iso,lightcone,ruled,doubleruled}{%
  \ifcase\nr\relax
    %% std
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Quadrille, ten squares per inch.
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, margin=0.2in}{0.1in}{%
%% Draw a grid with 10 squares per inch.
\draw[style=minorgrid, shift={(a)}] (0,0) grid [step=\GP@patternsize] (b);
%%
%% Draw a frame around the grid.
\draw[style=majorgrid] (a) rectangle (b);
    }
  \or
    %% stdeight
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Quadrille, eight squares per inch.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, margin=0.1875in}{0.125in}{%
%% Draw a grid with 10 squares per inch.
\draw[style=minorgrid, shift={(a)}] (0,0) grid [step=\GP@patternsize] (b);
%%
%% Draw a frame around the grid.
\draw[style=majorgrid] (a) rectangle (b);
    }
  \or
    %% majmin
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Graph paper, eight squares per inch with a major grid
%% every half-inch.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, margin=0.25in}{0.125in}{%
%% Draw a grid with 10 squares per inch.
\draw[style=minorgrid, shift={(a)}] (0,0) grid [step=\GP@patternsize] (b);
%%
\draw[style=majorgrid, shift={(a)}] (0,0) grid [step=4*\GP@patternsize]   (b);
%%
%% Draw a frame around the grid.
\draw[style=majorgrid] (a) rectangle (b);
    }
  \or
    %% dot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Dot grid
%% Slightly modified version of code added by Leo
%% Stein (@duetosymmetry).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{true}{}{0.1in}{%
    \fill [pattern=dotgrid,pattern color=minorcolor] (a) rectangle (b);
    }
```

```
  \or
    %% hex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Hex grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{true}{}{0.1666in}{%
    \fill [pattern=hexagons,pattern color=minorcolor] (a) rectangle (b);
    }
  \or
    %% hexup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Hex-up grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{true}{}{0.1666in}{%
    \fill [pattern={hexagons[rotate=90]},pattern color=minorcolor] (a) rectangle (b);
    }
  \or
    %% tri
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Triangle grid, adjust triangle size in the preamble
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{true}{}{0.25in}{%
      \fill [pattern=triangles,pattern color=minorcolor] (a) rectangle (b);
    }
  \or
    %% iso
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Isometric grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{true}{}{0.25in}{%
      \fill [pattern={triangles[rotate=90]}, pattern color=minorcolor] (a) rectangle (b);
    }
  \or
    %% lightcone
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% A grid with light cones.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, margin=.125in}{0.25in}{%
%% Draw a grid with 4 squares per inch.
\draw[style=minorgrid, shift={(a)}] (0,0) coordinate grid [step=\GP@patternsize] (b);
%%
%% Draw a border around the grid.
\draw[style=majorgrid, pattern={lightcones[myshift={(a)}]}, pattern color=lightlines] (a) rectang
    }
  \or
    %% ruled
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Ruled page with bold lines every 0.2in or 0.25in
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, body={8in,10.8in}}{0.2in}{%
%% Draw a ruled page with lines every 0.2in
\draw[style=majorgrid, shift={(a)}] (0,0) grid [ystep=\GP@patternsize, xstep=\paperwidth] (b);
%% Draw a frame around the grid.
\draw[style=majorgrid] (a) rectangle (b);
    }
  \or
    %% doubleruled
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Ruled page with bold lines every 0.25in and light lines
```

```
%% every 0.125 in.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    \GP@setpattern{false}{letterpaper, margin=.25in}{0.125in}{%
%% Draw a ruled pattern with thin lines every 0.125 in and bold lines every 0.25 in.
\draw[style=minorgrid, shift={(a)}] (0,0) grid [ystep=\GP@patternsize, xstep=\paperwidth] (b);
%%
\draw[style=majorgrid, shift={(a)}] (0,0) grid [ystep=2*\GP@patternsize, xstep=\paperwidth] (b);
%%
%% Draw a frame around the grid.
\draw[style=majorgrid] (a) rectangle (b);
    }
  \fi
}

%% Use the passed package option to set the above key
\expandafter\@setkeyhelper\expandafter{\GPOpt@pattern}{pattern}

%% Determine whether or not to (re)set fullpage vs textarea
\ifGP@fullnessset
%% Respect their choice
\else
  %% Reset the value of \GP@fullpage based on the pattern's default
  %% There's probably a more idiomatic way to do this but I can't
  %% figure it out
  \ifKV@GP@patterndefaultfullness
    \GP@fullpagetrue
  \else
    \GP@fullpagefalse
  \fi
\fi

%% Determine whether or not to fiddle with the page geometry
\ifGP@geometrypreviouslyloaded
%% Respect their previous choice
\PackageWarning{gridpapers}{'geometry' package was previously loaded, will not use pattern defaul
\else
  %% Use the pattern's defaults,
  \expandafter\geometry\expandafter{\GP@patterndefaultgeometry}
  %% And then override with any more specific settings passed by the user
  \expandafter\geometry\expandafter{\GPOpt@geometry}
\fi

%% Determine the correct pattern length
\ifx\GPOpt@patternsize\@empty
  % Use the pattern's preferred length
  \renewcommand{\GP@patternsize}{\GP@patterndefaultsize}
\else
  % Override with the user's choice
  \renewcommand{\GP@patternsize}{\GPOpt@patternsize}
\fi

%% Now that everything has been set up, we can finally define the
%% patterns with the correct lengths.
\GP@declarehexpat
\GP@declaretripat
\GP@declarelightconepat
\GP@declaredotpat

%% Set the background color.
```

```
\AtBeginDocument{\pagecolor{bgcolor}}
%% Actually hook it in!
\AddEverypageHook{%
\GP@patterncode%
}

\endinput
```

Change History