The pmdraw package*

Matthias Fresacher
matthias.ypg37@slmail.me

March 9, 2026

Abstract

The pmdraw package allows you to draw elements of the diagram monoids, commonly referred to as diagrams. The package provides a lot of flexibility to draw most diagrams and can be customised as needed.

Contents

*This document corresponds to pmdraw v2.4, 2026-03-09.

1 Introduction

1.1 Motivation

Diagram monoids and their closely related algebras are well studied objects with applications in theoretical physics and representation theory in mathematics. The elements of these monoids are called diagrams because they can be drawn graphically as a graph following certain conventions.

My PhD thesis examines properties of these diagram monoids and hence I needed to draw a large number of diagrams in a neat and consistent way. This thesis work formed the basis of this package with only customisability and the manual needed to be added to complete this package.

1.2 Acknowledgments

Most importantly, I thank James East, my PhD supervisor for allowing me to get a little sidetracked in my PhD research and develop and publish this package rather than work on my thesis like I should have.

This package has at its core the basic drawing macro of a diagram as used by James East and Nik Ruškuc in their 2022 publication *Classification of congruences of twisted partition monoids* in Advances in Mathematics. I thank them both for allowing me to build on their work.

I thank everybody that has helped me proof read, spot bugs, provide feedback, suggest improvements or has otherwise contributed to this package. Those that are happy to be named are (in chronological order):

- Nik Ruškuc
- Luka Carroll
- Yayi Zhu

This package heavily uses commands that can accept an arbitrary number of arguments as inputs. I thank `egreg` for their template upon which my commands are based (https://tex.stackexchange.com/a/72915).

As this was my first time using keys, I thank David Carlisle for their very helpful example (https://tex.stackexchange.com/a/542555).

As it turns out, passing my keys to the `\draw` command of `tikz` is a little more tricky than one might expect. I thank Andrew Stacey for their solution (https://tex.stackexchange.com/a/64237).

I thank Partha D. for their ready to use out of the box macro for adding grid lines to a `tikz` drawing (https://tex.stackexchange.com/a/467908).

I also thank everybody on https://tex.stackexchange.com who has answered a questions or solved a problem. There are too many to list you all. Thank you.

2 Troubleshooting

When something goes wrong, the first point of call should be this manual to ensure the command(s) being used is being used correctly. If that does not solve the problem, please check the list of bugs and known problems in Section 3.

A very common cause of problems is using an out of date version of this package. Please always use the latest version.

If this does not solve the problem, please get in touch to report the bug you found, see Section 3.1.

3 Bugs and known problems

3.1 Bugs

If you discover a bug, please be so kind and send me a minimum working example of the bug in action and I will work on fixing it.

I would like this package to be as perfect as it can be but I will need your help with this as there simply are too many keys to test every combination or use case.

The following are a list of currently known bugs that will be addressed:

1. When the radius of the arc for non-transversal edges is too large compared to the distance between the two nodes, the arcs overlap. The current solution is to manually specify a `level` or `height` that makes the arcs smaller.

2. Baseline adjustments are hard coded so when changing `scale`, `row sep` or text size, for example, the vertical position of diagrams is likely to be incorrect. The current solution is to manually adjust the `baseline` within the `tikz` key.

3. In a product diagram, if `row sep` is used in the bottom diagram, the spacing between diagrams is incorrect.

4. There are a number of options for equivalence relations and `E bricks` that do not work seamlessly and require manual adjustments to work. This is because they have not been fine tuned and assume a diagram instead of an equivalence relation is being drawn.

5. Edges are drawn as disjoint line segments. This means dotted or dashed line patters do not look even.

3.2 Incompatibilities

The following are a list of known problems or incompatibilities that are not scheduled to be resolved anytime soon:

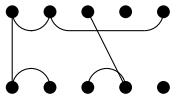
1. Whilst not strictly an incompatibility, when using `beamer`, frames have to be `fragile`.

4 Usage

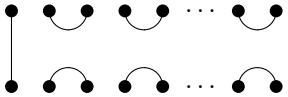
To use the package, one must understand that all diagrams are created from a basic building block called a brick.

4.1 Bricks

brick A `brick` is a diagram that contains no dots. For example,



is a brick whilst



<brick> is not a brick. The syntax for a brick contains four components and is as follows:

```
<brick> = [<options>]{<Uedges>}{<Ledges>}{<Tedges>}
```

brick/<options> where the `<options>` are described in Section 5.1.

<Uedges> is a list of upper non-transversal edges. An upper non-transversal edges has
Uedges/<options> three components `<Uedges> = [<options>]{<Svertex>}{<Tvertex>}` where the
Ledges/<options> `<options>` are described in Section 5.2.1, {<Svertex>} is the starting vertex
<Svertex> and {<Tvertex>} is the terminating vertex of the edge. Both {<Svertex>}
<Tvertex> and {<Tvertex>} are a x-coordinate and it is assumed that {<Svertex>} $<$ {<Tvertex>}.

list A `list` of upper non-transversal edges is simply each upper non-transversal edge written after the other. For example a list with one upper non-transversal edges is

```
{1}{2}
```

whilst one with three upper non-transversal edges may be

```
{1}{2}[level=1]{3}{4}{5}{6}
```

For ease of reading this will coded as

```
{1}{2}
[level=1]{3}{4}
{5}{6}
```

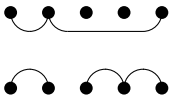
\pmdEmpty Should a list be empty, the command `\pmdEmpty` should be used.

<Ledges> is a list of lower non-transversal edges. A lower non-transversal edges has the same three components as an upper non-transversal edge, namely `<Ledges> = [<options>]{<Svertex>}{<Tvertex>}`.

<Tedges> is a list of transversal edges. A transversal edges has three components
Tedges/<options> `<Tedges> = [<options>]{<Tvertex>}{<Bvertex>}` where the `<options>` are de-
<Tvertex> scribed in Section 5.2.2, {<Tvertex>} is the top vertex and {<Bvertex>} is the bot-
<Bvertex> tom vertex of the edge. Both {<Tvertex>} and {<Bvertex>} are a x-coordinate.

An example `<brick>` and the resultant diagram is:

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
    {4}{5}
}{ % Transversal edges
    \pmdEmpty
}
```



`E brick` `<E brick>` An `E brick` is a modified brick for equivalence relations (see `\pmdEquiv`). In essence, it is just the top row of a brick. The syntax for an equivalence brick contains three components (removing the transversal edges `<Tedges>` from an ordinary brick) and is as follows:

```
<E brick> = [<options>]{<Uedges>}{<Ledges>}
```

where the `<options>` are the same as for `brick` (where they make sense) and are described in Section 5.1. Somewhat counter intuitively, the `Uedges` are drawn underneath the row of vertices and hence the `Ledges` are drawn above.

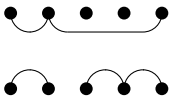
This is because the equivalence relation is by default the top row of a diagram and this has non-transversal edges drawn underneath. It also means that for most simple use cases, the edges of an `E brick` are entered in the first argument which makes sense and justifies the slightly confusing naming.

4.2 Draw commands

This package has the following draw commands.

`\pmdBrick` draws a brick and has the syntax `\pmdBrick<brick>`.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
    {4}{5}
}{ % Transversal edges
    \pmdEmpty
}
```

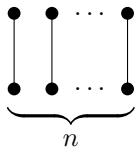


`\pmdDiagram` draws arbitrarily many bricks separated by dots and has the syntax

```
\pmdDiagram[<options>]{<bricks>}
```

`diagram/<options>` `<bricks>` where the `<options>` are described in Section 5.5 and `<bricks>` is a list of `<brick>`-s, with each brick enclosed in curly brackets.

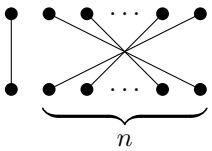
```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



It should be noted that `\pmdBrick` is simply a wrapper for a single brick that calls `\pmdDiagram` with no diagram options. This means that, for example, the `grid` key is not available for `\pmdBrick` and `\pmdDiagram` must be used instead to access the `grid` key.

For edges that go across multiple bricks, know that the coordinate positions of the vertices extends infinitely in both directions. If using default settings, the dots between bricks occupies one vertex position. To help with counting and identifying the correct coordinate position, the `grid` flag may also be used, see Section 5.5.1.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=3,
            brace={
                left=2,
                right=6,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{6}
            {3}{5}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{-1}
            {2}{-2}
        }
    }
}
```



`\pmdProduct` draws two diagrams on top of each other as is the case during multiplication or concatenation and has the syntax

```
\pmdProduct[<options>]{<Aedges>}{<Tbricks>}{<Bbricks>}
```

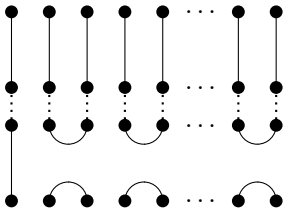
`diagram/<options>` where the `<options>` are described in Section 5.5.

`<Aedges>` is a list of edges that are added during the concatenation process. An added edge has two components, namely `<Aedges> = {<Svertex>}{<Tvertex>}`, where `{<Svertex>}` is the starting vertex and `{<Tvertex>}` is the terminating vertex of the collection of added edges. Both `{<Svertex>}` and `{<Tvertex>}` are a x-coordinate and it is assumed that `{<Svertex>}` $<$ `{<Tvertex>}`.

`<Tbricks>` is a list of bricks that correspond to the top diagram, with each brick enclosed in curly brackets.

`<Bbricks>` is a list of bricks that correspond to the bottom diagram, with each brick enclosed in curly brackets.

```
\pmdProduct{ % Added edges
    {1}{5}
    {7}{8}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {1}{2}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



`\pmdProductTriple` draws three diagrams on top of each other as is the case during multiplication or concatenation of three diagrams and has the syntax

```
\pmdProductTriple[<options>]{<ATedges>}{<ABedges>}{<Tbricks>}}{<Mbricks>}{<Bbricks>}
```

`diagram/<options>` where the `<options>` are described in Section 5.5.

`<ATedges>` is a list of edges that are added during the concatenation process. `<ATedges>`

`<ABedges>` is the list of edges between the top and middle diagrams, whilst `<ABedges>` is the list of edges between the middle and bottom diagrams. An added edge has two components, namely `<ATedges> =` {`<Svertex>`}{`<Tvertex>`} and `<ABedges> =` {`<Svertex>`}{`<Tvertex>`}, where {`<Svertex>`} is the starting vertex and {`<Tvertex>`} is the terminating vertex of the collection of added edges. Both {`<Svertex>`} and {`<Tvertex>`} are a x-coordinate and it is assumed that {`<Svertex>`} $<$ {`<Tvertex>`}.

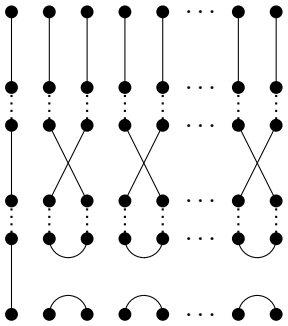
`<Tbricks>` is a list of bricks that correspond to the top diagram, with each brick enclosed in curly brackets.

`<Mbricks>` is a list of bricks that correspond to the top diagram, with each brick enclosed in curly brackets.

`<Bbricks>` is a list of bricks that correspond to the bottom diagram, with each brick enclosed in curly brackets.

```
\pmdProductTriple{ % Top added edges
    {1}{5}
    {7}{8}
}{ % Bottom added edges
    {1}{5}
    {7}{8}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }
}{ % Middle diagram
   % Continued on next page ...
```

```
   % ... continued from previous page (note repeated line)
}{ % Middle diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{3}
            {3}{2}
            {4}{5}
            {5}{4}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{2}
            {2}{1}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {1}{2}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



`\pmdEquiv` draws an equivalence relation determined by arbitrarily many equivalence bricks separated by dots. This can the thought of as just the top row of a diagram. It has the syntax

```
\pmdEquiv[<options>]{<E bricks>}
```

where the `<options>` are the same as for `diagram` (where they make sense) and `<E bricks>` are described in Section 5.5 and `<E bricks>` is a list of `<E brick>`-s, with each equivalence brick enclosed in curly brackets.

```
\pmdEquiv{
    { % E Brick 1
        [ % Options
            degree=6
        ]{ % Upper (underneath) non-transversal edges
            {1}{2}
            {5}{6}
        }{ % Lower (above) non-transversal edges
            {3}{4}
        }
    }{ % E Brick 2
        [ % Options
            degree=2
        ]{ % Upper (underneath) non-transversal edges
            \pmdEmpty
        }{ % Lower (above) non-transversal edges
            {1}{2}
        }
    }
}
```

5 Options

5.1 Brick options

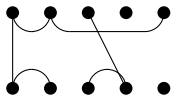
The following are all of the keys that are available for bricks.

5.1.1 Vertex options

`degree` specifies the degree, that is, the number of vertices in the top and bottom row of

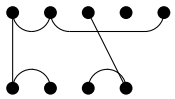
the diagram. Must be a non-negative integer and must be provided unless using `degree top` and `degree bottom`. If `degree=0` then no vertices are drawn.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



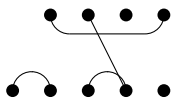
`degree top` `degree bottom` specifies the degree, that is, the number of vertices in the top (bottom) row of the diagram. Must be a non-negative integer integer and must be provided as a pair `degree top` and `degree bottom`. If `degree top=0` then no top vertices are drawn and if `degree bottom=0` then no bottom vertices are drawn

```
\pmdBrick[ % Options
    degree top=5,
    degree bottom=4
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`blank top` `blank bottom` specifies the number of blank vertices to the left of the top (bottom) row of the diagram. Default is zero.

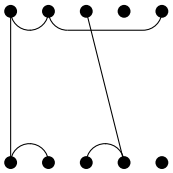
```
\pmdBrick[ % Options
    blank top=1,
    degree top=4,
    degree bottom=5
]{ % Upper non-transversal edges
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {3}{4}
}
```



It should also be noted that the `blank top` and `blank bottom` can also be used to create more space between bricks and the dots between bricks by setting a large number. Likewise if set to -1 and the dots is redefined to not draw, the space between bricks can be removed altogether. See Examples 13-15.

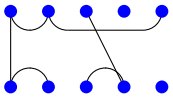
`row sep` specifies the vertical separation between rows of vertices. Default is two. If using edge options such as `levels`, `row sep` must be given before `levels`.

```
\pmdBrick[ % Options
    degree=5,
    row sep=4
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



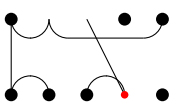
`vertices` passes through options to the `\draw` command of vertices. The default is no argument.

```
\pmdBrick[ % Options
    degree=5,
    vertices=blue
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`no vertex bottom 0` … `no vertex bottom 19` `no vertex top 0` … `no vertex top 19` specifies a vertex that is not drawn. Any of the twenty available keys can be used in any order. It can be used to draw single vertices that are different to the others. If more than twenty vertices need to be excluded, simply draw multiple bricks next to each other. The default is -1.

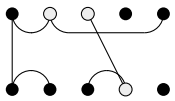
```
\pmdBrick[ % Options
    degree=5,
    no vertex top 1=2,
    no vertex top 2=3,
    no vertex bottom 1=4,
    decorate after={
        \fill[red] (4,0) circle (.1);
    }
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`ghost vertex bottom 0` … `ghost vertex bottom 19`
`ghost vertex top 0` … `ghost vertex top 19`

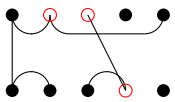
specifies a vertex that is drawn in a 'ghost' style. Any of the twenty available keys can be used in any order. If more than twenty vertices need to be ghosted, simply draw multiple bricks next to each other. The default is -1.

```
\pmdBrick[ % Options
    degree=5,
    ghost vertex top 1=2,
    ghost vertex top 2=3,
    ghost vertex bottom 1=4
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`ghosts` passes through options to the `\draw` command of ghost vertices. The default is no argument.

```
\pmdBrick[ % Options
    degree=5,
    ghosts=red,
    ghost vertex top 1=2,
    ghost vertex top 2=3,
    ghost vertex bottom 1=4
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```

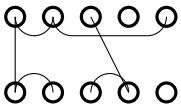


`vertices options` allows for the redefinition of the command that draws the vertices. It has two arguments, the x and y-position of the vertex. The default is

```
\fill[
    apply style/.expand once=\pmdraw@drawVertexOptionsDefault,
    apply style/.expand once=\pmdraw@drawVertexOptions
] (#1,#2) circle (.17);
```

An example use is

```
\pmdBrick[ % Options
    degree=5,
    vertices options={
        \draw[very thick] (#1,#2) circle (.25);
    }
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```

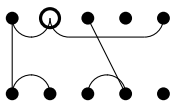


`ghosts options` allows for the redefinition of the command that draws the ghost vertices. It has two arguments, the x and y-position of the vertex. The default is

```
\fill[
    apply style/.expand once=\pmdraw@drawGhostOptionsDefault,
    apply style/.expand once=\pmdraw@drawGhostOptions
] (#1,#2) circle (.17);
```

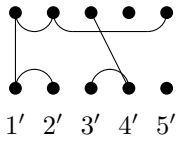
An example use is

```
\pmdBrick[ % Options
    degree=5,
    ghosts options={
        \draw[very thick] (#1,#2) circle (.25);
    },
    ghost vertex top 1=2
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`labels` `labels top` `labels bottom` flags that allow for the drawing of labels for the vertices. Use `labels` for labels on both the top and bottom or set individually. The internal values used are `1` for true and `0` for false. Hence to negate use `labels=0`. The default is no labels.

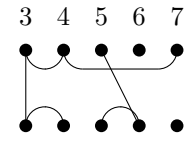
```
\pmdBrick[ % Options
    degree=5,
    labels bottom
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`labels start` sets the starting value of the labels for the vertices. The default is one.

`labels top start`
`labels bottom start`

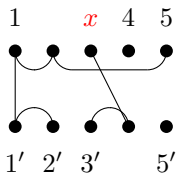
```
\pmdBrick[ % Options
    degree=5,
    labels,
    labels top start=3,
    labels bottom start=-2
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`no label bottom 0` ⋮ `no label bottom 19`
`no label top 0` ⋮ `no label top 19`

specifies a vertex for which no label is drawn. Any of the twenty available keys can be used in any order. It can be used to insert single unique labels that are different to the others. If more than twenty labels need to be excluded, simply draw multiple bricks next to each other. The default is -1.

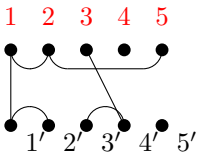
```
\pmdBrick[ % Options
    degree=5,
    labels,
    no label top 1=2,
    no label top 2=3,
    no label bottom 1=4,
    decorate after={
        \draw (3,2) node[above=6pt,red] {\(x\)};
    }
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`labels top draw`
`labels top node`
`labels bottom draw`
`labels bottom node`

passes through options to the `\draw` and `node` command of the labels of vertices. The default is no argument.

```
\pmdBrick[ % Options
    degree=5,
    labels,
    labels top node=red,
    labels bottom node={right=2pt}
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`labels top options`
`labels bottom options`

allows for the redefinition of the command that draws the labels of the vertices. They have three arguments, the x and y-position of the vertex and the label value. The default for the top is

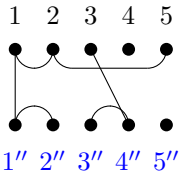
```
\draw[apply style/.expand once=\pmdraw@drawLabelTDrawOptions] (#1,#2)
    node[above=6pt,apply style/.expand once=\pmdraw@drawLabelTNodeOptions] {\(#3\)};
```

and for the bottom

```
\draw[apply style/.expand once=\pmdraw@drawLabelBDrawOptions] (#1,#2)
    node[below=6pt,apply style/.expand once=\pmdraw@drawLabelBNodeOptions] {\(#3'\)};
```

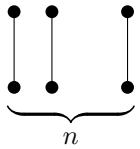
An example use is

```
\pmdBrick[ % Options
    degree=5,
    labels,
    labels bottom options={
        \draw (#1,#2) node[below=6pt,blue] {\(#3''\)};
    }
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



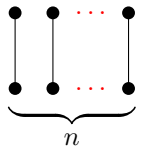
`no dots` The `no dots` flag removes the drawing of dots to the left of the current brick in diagrams. The internal values used are `1` for true and `0` for false. Hence to negate use `no dots=0`. Default is to draw dots except for the first brick.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1,
            no dots
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`dots draw` passes through options to the `\draw` and `node` command of the dots. The `dots node` default is no argument.

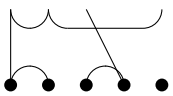
```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1,
            dots node=red
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`vertices top phantom`
`vertices bottom phantom`

These flags allow for all of the top or bottom vertices to be hidden using the `phantom` command. Simply wraps the vertex drawing macros with a `phantom` command. This hiding will also apply to the labels of that row (top or bottom) of vertices. To only hide a selected number of vertices, use seperate bricks. The default is false.

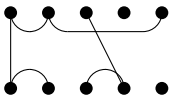
```
\pmdBrick[ % Options
    degree=5,
    vertices top phantom,
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`vertices top uncover`
`vertices bottom uncover`

allows for all of the top or bottom vertices to be uncovered in a `beamer` presentation[1] using the command `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the vertex drawing macros. This uncover will also apply to the labels of that row (top or bottom) of vertices. To only uncover a selected number of vertices, use seperate bricks. The default is `1-`.

```
\pmdBrick[ % Options
    degree=5,
    vertices top uncover={2,4-7,9-},
    vertices bottom uncover={3-}
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



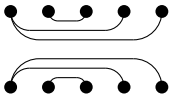
5.1.2 Edge options

These options should be read in conjunction with Section 5.2.

`levels`

specifies how many horizontal levels there are for non-transversal edges. Default is one.

```
\pmdBrick[ % Options
    degree=5,
    levels=3
]{ % Upper non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Lower non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Transversal edges
    \pmdEmpty
}
```

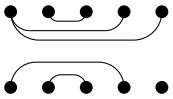


`levels top`
`levels bottom`

specifies how many horizontal levels there are for upper (lower) non-transversal edges. Default is one.

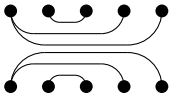
[1] Make sure to use a `fragile` frame.

```
\pmdBrick[ % Options
    degree=5,
    levels top=3,
    levels bottom=2
]{ % Upper non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Lower non-transversal edges
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Transversal edges
    \pmdEmpty
}
```



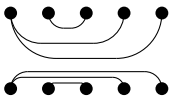
`levels sep` specifies the vertical separation between horizontal levels for non-transversal edges. Default is evenly spaced within the space available for the given number of `levels`.

```
\pmdBrick[ % Options
    degree=5,
    levels=3,
    levels sep=0.3
]{ % Upper non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Lower non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Transversal edges
    \pmdEmpty
}
```



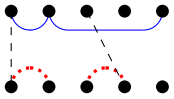
`levels sep top` `levels sep bottom` specifies the vertical separation between horizontal levels for upper (lower) non-transversal edges. Default is evenly spaced within the space available for the given number of `levels`.

```
\pmdBrick[ % Options
    degree=5,
    levels=3,
    levels sep top=0.4,
    levels sep bottom=0.15
]{ % Upper non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Lower non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Transversal edges
    \pmdEmpty
}
```



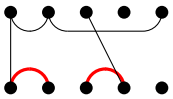
`edges` `edges non-transversal` `edges upper` `edges lower` `edges transversal` pass through options to the `\draw` command of their respective edge types. The default is no argument.

```
\pmdBrick[ % Options
    degree=5,
    edges upper=blue,
    edges lower={red, dotted, very thick},
    edges transversal=dashed
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



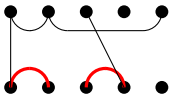
`edges first` flag ensures edges are drawn before vertices. This key is now redundant as since version 2.1 edges are drawn before vertices by default but it is retained for backward compatibility. The internal values used are `1` for true and `0` for false. Hence to negate use `edges first=0`. For the drawing order of a brick, see Section 5.1.6. The default is edges are drawn first.

```
\pmdBrick[ % Options
    degree=5,
    edges lower={red, very thick},
    edges first
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



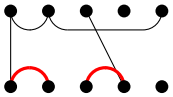
`edges last` flag ensures edges are drawn after vertices. The internal values used are `1` for true and `0` for false. Hence to negate use `edges last=0`. For the drawing order of a brick, see Section 5.1.6. The default is edges are drawn first.

```
\pmdBrick[ % Options
    degree=5,
    edges lower={red, very thick},
    edges last
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



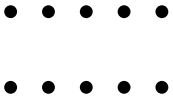
`transversals first` flag ensures transversal edges are drawn before non-transversal edges. To mix and match the drawing order of transversal and non-transversal edges, use two or more bricks next to each other with no dots and remember that edges can be drawn outside of a brick. The internal values used are `1` for true and `0` for false. Hence to negate use `transversals first=0`. For the drawing order of a brick, see Section 5.1.6. The default is non-transversal edges are drawn first.

```
\pmdBrick[ % Options
    degree=5,
    edges lower={red, very thick},
    transversals first
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`edges phantom` This flag allows for all edges to be hidden using the `phantom` command. Simply wraps the edge drawing macros with a `phantom` command. The internal values used are `1` for true and `0` for false. Hence to negate use `edges phantom=0`. The default is false.

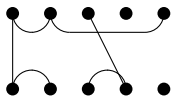
```
\pmdBrick[ % Options
    degree=5,
    edges phantom
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`edges uncover` allows for all edges to be uncovered in a `beamer` presentation[2] using the com-

mand `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the edge drawing macros. The default is `1-`.

```
\pmdBrick[ % Options
    degree=5,
    edges uncover={2,4-7,9-}
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```

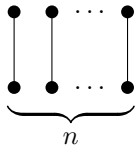


5.1.3 Brace

`brace` specifies instructions for the drawing of a brace with the brace keys. These are described in Section 5.3. It should be noted that only one brace can be drawn per brick. To draw more, draw as many bricks as braces (possibly without dots). The default is no brace is drawn.

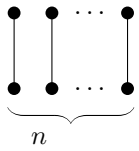
```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```

[2]Make sure to use a `fragile` frame.



`brace draw` passes through options to the `\draw` and `node` command of the brace. Must
`brace node` be given before the `brace` key. The default is no argument.

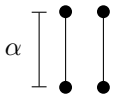
```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace draw=thin,
            brace node={pos=0.25},
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



5.1.4 Bracket

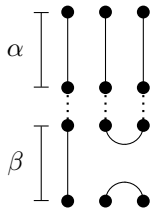
`bracket` specifies instructions for the drawing of a bracket with the bracket keys. These are described in Section 5.4. It should be noted that only one bracket can be drawn per brick. To draw more, draw as many bricks as brackets without dots and use the `shift` key to position as desired. The default is no bracket is drawn.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha}
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



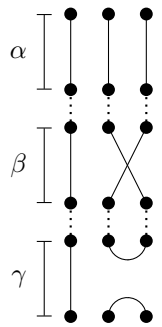
The `bracket` key also works for product diagrams as expected:

```
\pmdProduct{ % Added edges
    {1}{3}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\alpha}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\beta}
            }
        ]{ % Upper non-transversal edges
            {2}{3}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



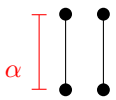

```
\pmdProductTriple{ % Top added edges
    {1}{3}
}{ % Bottom added edges
    {1}{3}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\alpha}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
        }
    }
}{ % Middle diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\beta}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{3}
            {3}{2}
        }
    }
}{ % Bottom diagram
   % Continued on next page ...
```

```
   % ... continued from previous page (note repeated line)
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\gamma}
            }
        ]{ % Upper non-transversal edges
            {2}{3}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`bracket draw` `bracket node` passes through options to the `\draw` and `node` command of the bracket. Must be given before the `bracket` key. The default is no argument.

```
\pmdBrick[ % Options
    degree=2,
    bracket draw=red,
    bracket node={pos=0.25},
    bracket={
        label={\alpha}
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```

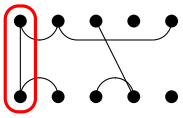


5.1.5 Decorations

`decorate before` `decorate after` provides a hook to draw additional `tikz` elements before and after the brick is

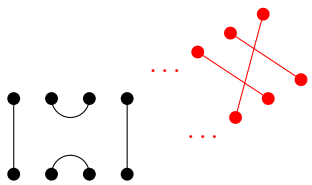
drawn. For the drawing order of a brick, see Section 5.1.6. The default is empty.

```
\pmdBrick[ % Options
    degree=5,
    decorate after={
        \draw[
            very thick,
            red,
            rounded corners=5pt
        ] (0.6, -0.4) rectangle (1.4, 2.4);
    }
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



These hooks can also be used for more sophisticated manipulation of diagrams be inserting not just `tikz` elements but arbitrary code before and after draw a brick.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=4
        ]{ % Upper non-transversal edges
            {2}{3}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            {1}{1}
            {4}{4}
        }
    }{ % Brick 2
        [ % Options
            degree=3,
            decorate before={\begin{scope}[red,shift={(1,1)},rotate=30]},
            decorate after={\end{scope}}
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{2}
            {2}{3}
            {3}{1}
        }
    }
}
```



5.1.6 Drawing order

For the drawing order of a diagram and product diagram, see Section 5.5.3.

From version 2.1 onwards: For clarity, this is the default drawing order for a brick:

1. `decorate before`
2. Upper non-transversal edges
3. Lower non-transversal edges
4. Transversal edges
5. Top row of vertices
6. Bottom row of vertices
7. `decorate after`

This order can be altered using the `edges first`, `edges last` and `transversals first` flags.

Version 2.0 and before: For clarity, this was the default drawing order for a brick:

1. `decorate before`
2. Top row of vertices
3. Bottom row of vertices
4. Upper non-transversal edges
5. Lower non-transversal edges
6. Transversal edges
7. `decorate after`

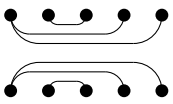
This order could be altered using the `edges first` and `transversals first` flags. The `edges last` key was only introduced in version 2.1.

5.2 Edge options

5.2.1 Non-transversal edge options

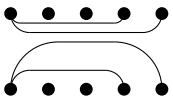
`level` specifies the horizontal level of a non-transversal edges. Default is one.

```
\pmdBrick[ % Options
    degree=5,
    levels=3
]{ % Upper non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Lower non-transversal edges
    [level=3]{1}{5}
    [level=2]{1}{4}
    [level=1]{2}{3}
}{ % Transversal edges
    \pmdEmpty
}
```



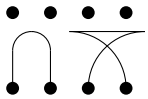
`height` specifies a manually set horizontal height of a non-transversal edges. Default is to use levels.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{5}
    [height=1.75]{1}{4}
}{ % Lower non-transversal edges
    [height=1.25]{1}{5}
    {1}{4}
}{ % Transversal edges
    \pmdEmpty
}
```



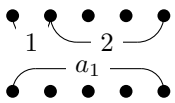
`radius` specifies the radius of the arcs of a non-transversal edge. It should not be larger than the edge's height relative to the row of vertices. This key can be used when edges appear wrongly drawn as the default arcs are too large. Default is to use the edge's height relative to the row of vertices.

```
\pmdBrick[ % Options
    degree=4
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    [height=1.5, radius=0.5]{1}{2}
    [height=1.5]{3}{4}
}{ % Transversal edges
    \pmdEmpty
}
```



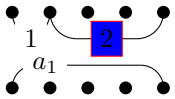
`label` adds a text label to the edge. This can be used for colourings where most commonly the colours are thought of as the natural numbers. Default is no label.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    [label=1, height=1.3]{1}{2}
    [label=2, height=1.3]{2}{5}
}{ % Lower non-transversal edges
    [label=$a_1$, height=0.6]{1}{5}
}{ % Transversal edges
    \pmdEmpty
}
```



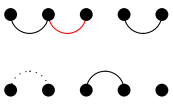
`label node` passes through options to the `node` command that draws the label. The default is `fill=white, pos=0.5`.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    [label=1, height=1.3]{1}{2}
    [label=2, label node={draw=red, fill=blue}, height=1.3]{2}{5}
}{ % Lower non-transversal edges
    [label=$a_1$, label node={pos=0.1}, height=0.6]{1}{5}
}{ % Transversal edges
    \pmdEmpty
}
```



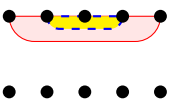
`edge draw` passes through options to the `\draw` command of the edge. The default is no argument.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    [edge draw=red]{2}{3}
    {4}{5}
}{ % Lower non-transversal edges
    [edge draw=dotted]{1}{2}
    {3}{4}
}{ % Transversal edges
    \pmdEmpty
}
```



`fill` draws an edges that is filled and passes through options to the `\draw` command of the filled edge. The default is no argument.

```
\pmdBrick[ % Options
    degree=5,
    levels=2
]{ % Upper non-transversal edges
    [level=2, fill={red, fill=red!10}]{1}{5}
    [fill={blue, dashed, thick, fill=yellow}]{2}{4}
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    \pmdEmpty
}
```



`options` allows for the redefinition of the command that draws the non-transversal edge. It has two arguments, the start and end x-position of the edge. The current default is now quite long and should just be looked at directly in source, but the main components of the command for an upper non-transversal are

```
\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#1,\pmdraw@rowSep) arc (180:270:\pmdraw{edgeHeight});% Draw left arch
\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#1+\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight})
        -- (#2-\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight});
        % Draw straight line
\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#2-\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight})
        arc (270:360:\pmdraw{edgeHeight});% Draw right arc
\renewcommand{\pmdraw@drawEdgeDrawOptions}{}% Reset draw options for edge
```

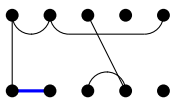
and for a lower non-transversal are

```
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#1,0) arc (180:90:\pmdraw{edgeHeight});% Draw left arch
```

```
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#1+\pmdraw{edgeHeight},\pmdraw{edgeHeight})
        -- (#2-\pmdraw{edgeHeight},\pmdraw{edgeHeight});% Draw straight line
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions,
    apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
    (#2-\pmdraw{edgeHeight},\pmdraw{edgeHeight})
        arc (90:0:\pmdraw{edgeHeight});% Draw right arc
\renewcommand{\pmdraw@drawEdgeDrawOptions}{}% Reset draw options for edge
```

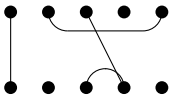
An example use is

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    [options={
        \draw[blue, very thick] (#1,0) -- (#2,0);
    }]{1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



`phantom` This flag allows for the edge to be hidden using the `phantom` command. Simply wraps the edge drawing macros with a `phantom` command. The internal values used are `1` for true and `0` for false. Hence to negate use `phantom=0`. The default is false.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    [phantom]{1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    [phantom]{1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```

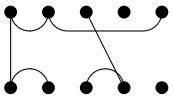


`uncover` allows for that edge to be uncovered in a `beamer` presentation[3] using the command `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the edge drawing macros. This uncover is nested inside the `edges uncover` and hence the limitations of nested uncovers are present here. For example, an individual edge will not uncover until the `edges`

[3] Make sure to use a `fragile` frame.

`uncover` has been uncovered. In such instances only use individual uncovers for each edge. The default is `1-`.

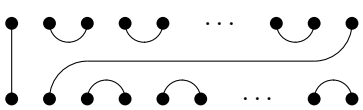
```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    [uncover={2,4-7,9-}]{1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    [uncover={3-}]{1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



5.2.2 Transversal edge options

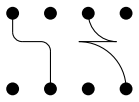
`height` draws transversal edges with a horizontal component and specifies the manually set horizontal height of the transversal edges. Default is to use non-horizontal transversal edges.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree top=5,
            degree bottom=6
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {3}{4}
            {5}{6}
        }{ % Transversal edges
            [height=1]{10}{2}
            {1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree top=3,
            blank bottom=1,
            degree bottom=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



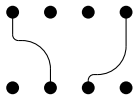
`radius` specifies the radius of the arcs of a transversal edge that uses the `height` key. It should not be larger than the minimum of the edge's height relative to both rows of vertices. This key can be used when edges appear wrongly drawn as the default arcs are too large. Default is to use the edge's height relative to the row of vertices with different radius for the left and right arcs if the relative heights are different.

```
\pmdBrick[ % Options
    degree=4
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    [height=1.25, radius=0.25]{1}{2}
    [height=1.25]{3}{4}
}
```



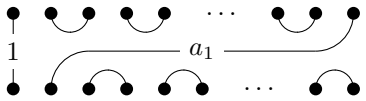
`radius left` `radius right` same as `radius` key except left and right radii can be specified independently. If one is given, then so must be the other. The value should not be larger than the edge's height relative to the respective row of vertices.

```
\pmdBrick[ % Options
    degree=4
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    [ % Options
        height=1.25,
        radius left=0.15,
        radius right=0.75
    ]{1}{2}
    [ % Options
        height=0.35,
        radius left=0.15,
        radius right=0.75
    ]{4}{3}
}
```



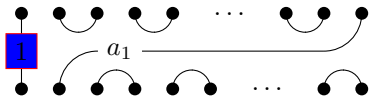
`label` adds a text label to the edge. This can be used for colourings where most commonly the colours are thought of as the natural numbers. Default is no label.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree top=5,
            degree bottom=6
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {3}{4}
            {5}{6}
        }{ % Transversal edges
            [height=1, label={$a_1$}]{10}{2}
            [label=1]{1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree top=3,
            blank bottom=1,
            degree bottom=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



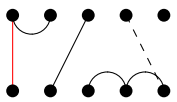
`label node` passes through options to the `node` command that draws the label. The default is `fill=white, pos=0.5`.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree top=5,
            degree bottom=6
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {3}{4}
            {5}{6}
        }{ % Transversal edges
            [height=1, label={$a_1$}, label node={pos=0.1}]{10}{2}
            [label=1, label node={draw=red, fill=blue}]{1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree top=3,
            blank bottom=1,
            degree bottom=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



`edge draw` passes through options to the `\draw` command of the edge. The default is no argument.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
}{ % Lower non-transversal edges
    {4}{5}
    {3}{4}
}{ % Transversal edges
    [edge draw=red]{1}{1}
}
```



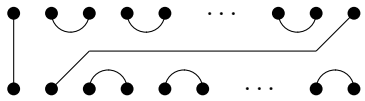
`options` allows for the redefinition of the command that draws the transversal edge. It has two arguments, the x-position of the edge in the top row of vertices and the x-position of the edge in the bottom row of vertices. The current default is now quite long and should just be looked at directly in source, but the main components of the command are

```
\ifnum\pmdraw@ifTedgeHorizontal=0% If drawing a straight line edge
```

```
        \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
            apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
            (#1,\pmdraw@rowSep) -- (#2,0);% Draw straight line
    \else% If transversal edge is drawn horizontally
        \ifnum#1>#2% If edge goes from top right to bottom left
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions] (#2,0) arc
                (180:90:\pmdraw{Tlevel});% Draw bottom arch
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
                (#2+\pmdraw{Tlevel},\pmdraw{Tlevel}) --
                (#1-\pmdraw@rowSep+\pmdraw{Tlevel}, \pmdraw{Tlevel});% Draw straight line
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
                (#1-\pmdraw@rowSep+\pmdraw{Tlevel},\pmdraw{Tlevel}) arc
                (270:360:{\pmdraw@rowSep-\pmdraw{Tlevel}});% Draw top arc
        \else% If edge goes from top left to bottom right
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
                (#2,0) arc (0:90:\pmdraw{Tlevel});% Draw bottom arch
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
                (#2-\pmdraw{Tlevel},\pmdraw{Tlevel}) --
                (#1+\pmdraw@rowSep-\pmdraw{Tlevel}, \pmdraw{Tlevel});% Draw straight line
            \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions,
                apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
                (#1+\pmdraw@rowSep-\pmdraw{Tlevel},\pmdraw{Tlevel}) arc
                (270:180:{\pmdraw@rowSep-\pmdraw{Tlevel}});% Draw top arc
        \fi%
        \renewcommand{\pmdraw@ifTedgeHorizontal}{0}% Reset flag to default
    \fi%
    \renewcommand{\pmdraw@drawEdgeDrawOptions}{}% Reset draw options for edge
```

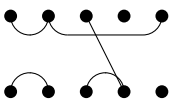
An example use is

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree top=5,
            degree bottom=6
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {3}{4}
            {5}{6}
        }{ % Transversal edges
            [options={
                \draw (#2,0) -- (#2+1,1); % Draw bottom line
                \draw (#2+1,1) -- (#1-1, 1); % Draw straight line
                \draw (#1-1,1) -- (#1,2); % Draw top line
            }]{10}{2}
            {1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree top=3,
            blank bottom=1,
            degree bottom=2
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



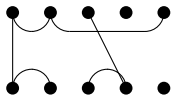
`phantom` This flag allows for the edge to be hidden using the `phantom` command. Simply wraps the edge drawing macros with a `phantom` command. The internal values used are `1` for true and `0` for false. Hence to negate use `phantom=0`. The default is false.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    [phantom]{1}{1}
    {3}{4}
}
```



`uncover` allows for that edge to be uncovered in a `beamer` presentation[4] using the command `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the edge drawing macros. This uncover is nested inside the `edges uncover` and hence the limitations of nested uncovers are present here. For example, an individual edge will not uncover until the `edges uncover` has been uncovered. In such instances only use individual uncovers for each edge. The default is `1-`.

```
\pmdBrick[ % Options
    degree=5
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    [uncover={2,4-7,9-}]{1}{1}
    {3}{4}
}
```



5.3 Brace

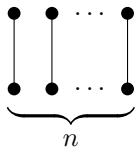
`brick/brace` As part of a diagram, it may be useful to draw a brace. It should be noted that only one brace can be drawn per brick. To draw more, draw as many bricks as braces (possibly without dots). The options of a brace are as follows:

`left` `right` specifies the x-position of the start/left and end/right of the brace. Both must be provided.

`label` specifies the label to be printed with the brace. Default is empty.

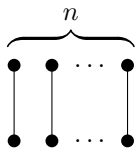
[4] Make sure to use a `fragile` frame.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`above` The `above` flag draws the brace above the diagram. The internal values used are `1` for true and `0` for false. Hence to negate use `above=0`. Default is below.

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n},
                above
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



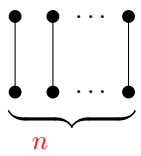
`options` allows for the redefinition of the command that draws the brace. It has three arguments, the x-position start and end of the brace, as well as the label of the brace. The current default is now quite long and should just be looked at directly in source, but the main components of the command are

```
\ifnum\pmdraw@ifProdDiag=1 % If drawing a product diagram
                           % or if placing brace above diagram
    \draw[
        very thick,
        decorate,
        decoration={calligraphic brace,amplitude=6pt},
        apply style/.expand once=\pmdraw@drawBraceDrawOptions
    ] (#1-0.17,\pmdraw@rowSep+0.5) -- (#2+0.17,\pmdraw@rowSep+0.5)
    node[
        pos=0.5,
        above=6pt,
        apply style/.expand once=\pmdraw@drawBraceNodeOptions
    ] {\(#3\)}; % Draw brace on top of diagram
```

```
\else % If not drawing a product diagram or if placing brace below diagram
    \draw[
        very thick,
        decorate,
        decoration={calligraphic brace,mirror,amplitude=6pt},
        apply style/.expand once=\pmdraw@drawBraceDrawOptions
    ] (#1-0.17,-0.5) -- (#2+0.17,-0.5)
    node[
        pos=0.5,
        below=6pt,
        apply style/.expand once=\pmdraw@drawBraceNodeOptions
    ] {\(#3\)}; % Draw brace on bottom of diagram
\fi%
```

It should be noted that the flag `\pmdraw@ifProdDiag` is both used for when drawing a product diagram but also when placing the brace above the diagram as this clearly must be the default for the top diagram in a product diagram. An example use is

```
\pmdDiagram{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n},
                options={
                    \draw[
                        red,
                        very thick,
                        decorate,
                        decoration={calligraphic brace,mirror,amplitude=6pt}
                    ] (#1-0.17,-0.5) --  (#2+0.17,-0.5)
                        node[pos=0.25,below=6pt]{\(#3\)};
                }
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```

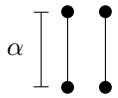


5.4 Bracket

`brick/bracket` As part of a brick, it may be useful to draw a bracket. It should be noted that only one bracket can be drawn per brick. To draw more, draw as many bricks as brackets without dots and use the `shift` key to position as desired. The options of a bracket are as follows:

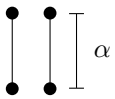
`label` specifies the label to be printed with the bracket. Default is empty.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha}
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



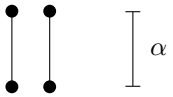
`left` `right` The `left`/`right` flag draws the bracket to the left or right of the brick respectively. In some complex diagrams (particularly if using the `right` flag), it may be that the default horizontal positioning of a bracket is not correct or otherwise requires manual adjustment. In this case use the `shift` key. Default is left.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha},
        right
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



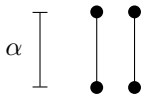
`shift` manually shifts the bracket in the horizontal direction. It should be noted that this shifts the default position (the left or right most vertex of the brick assuming the brick is not too complicated) of the bracket from which the separation as set by the `sep` key is then add to the left or right depending on the `left`/`right` flags. Default is 0.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha},
        shift=1.5,
        right
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



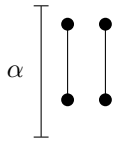
`sep` sets the separation between the bracket and the brick. Default is 0.7.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha},
        sep=1.5
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



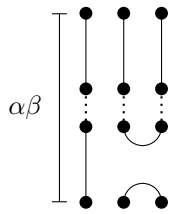
`up` `down` The `up`/`down` keys respectively raise and lower the starting point of the bracket. Default is 0.

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha},
        up=0.5,
        down=1
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



This can be useful when drawing products or triple products as seen in the following example:

```
\pmdProduct{ % Added edges
    {1}{3}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=3,
            bracket={
                label={\alpha \beta},
                down=3
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=3
        ]{ % Upper non-transversal edges
            {2}{3}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



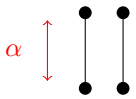
`options` allows for the redefinition of the command that draws the bracket. It has one argument, the label of the brace. The current default is now quite long and should just be looked at directly in source, but the main components of the command are

```
\begin{scope}[shift={(\pmdraw@BracketShift,0)}]% Shifts bracket horizontally
    \ifnum\pmdraw@ifBracketLeft=1% If placing bracket to the left of the brick
        \draw[
            |-|,
            apply style/.expand once=\pmdraw@drawBracketDrawOptions
        ] ({1-\pmdraw@BracketSep},0) -- ({1-\pmdraw@BracketSep},\pmdraw@rowSep)
        node[
            pos=0.5,
            left=3pt,
```

```
            apply style/.expand once=\pmdraw@drawBracketNodeOptions
        ] {\(#1\)};% Draw bracket on left of brick
    \else% If placing bracket to the right of the brick
        \draw[
            |-|,
            apply style/.expand once=\pmdraw@drawBracketDrawOptions
        ]
        ({\value{pmdraw@degreeB}+\pmdraw@BracketSep},0) --
            ({\value{pmdraw@degreeB}+\pmdraw@BracketSep},\pmdraw@rowSep)
        node[
            pos=0.5,
            right=3pt,
            apply style/.expand once=\pmdraw@drawBracketNodeOptions
        ] {\(#1\)};% Draw bracket on right of brick
    \fi%
\end{scope}%
```

An example use is

```
\pmdBrick[ % Options
    degree=2,
    bracket={
        label={\alpha},
        options={
            \draw[
                red,
                <->
            ] (0,0.2) --  (0,1.8)
                node[pos=0.5,left=6pt]{\(#1\)};
        }
    }
]{ % Upper non-transversal edges
    \pmdEmpty
}{ % Lower non-transversal edges
    \pmdEmpty
}{ % Transversal edges
    {1}{1}
    {2}{2}
}
```



5.5 Diagram options

5.5.1 For all (product) diagrams

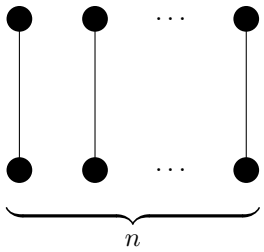
The following options can be used for both diagrams (`\pmdDiagram`) and product diagrams (`\pmdProduct`).

`row sep` same as `row sep` option in bricks but applied to all bricks within (product) diagram.

`tikz` passes through options to `tikz` environment. Specifically, the options in `\begin{tikzpicture}[<options>]`. The default is no argument. However there

are some default options passed to `tikz` automatically. These are an adjustment to the `baseline` depending on the current typesetting mode and a `grid` key if the grid is activated.

```
\pmdDiagram[ % Options
    tikz={scale=2}
]{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=4,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```

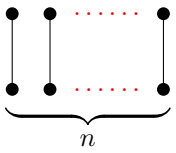


`dots options` allows for the redefinition of the command that draws dots between bricks. It has two arguments, the x and y-position of the dots. The default is

```
\draw[apply style/.expand once=\pmdraw@drawDotsDrawOptions]
    (#1,#2) node[apply style/.expand once=\pmdraw@drawDotsNodeOptions] {\(\dots\)};
```

An example use is

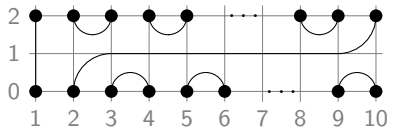
```
\pmdDiagram[ % Options
    dots options={
        \draw[red] (#1,#2) node {\(\dots\dots\)};
    }
]{
    { % Brick 1
        [ % Options
            degree=2,
            brace={
                left=1,
                right=5,
                label={n}
            }
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
        }
    }{ % Brick 2
        [ % Options
            degree=1,
            blank top=1,
            blank bottom=1
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {2}{2}
        }
    }
}
```



`grid` As it can get confusing where things are drawn and the different coordinate scopes between all the bricks and diagrams, the `grid` flag draws the base coordinate grid that the first drawn brick is based on. This should help with all coordinate related inquiries. The internal values used are `1` for true and `0` for false. Hence to negate use `grid=0`. Default is no grid.

It should be noted that the `grid` key is not available for `\pmdBrick` and `\pmdDiagram` must be used instead when needing a grid for a single brick only.

```
\pmdDiagram[ % Options
    grid
]{
    { % Brick 1
        [ % Options
            degree top=5,
            degree bottom=6,
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {3}{4}
            {5}{6}
        }{ % Transversal edges
            [height=1]{10}{2}
            {1}{1}
        }
    }{ % Brick 2
        [ % Options
            degree top=3,
            blank bottom=1,
            degree bottom=
        ]{ % Upper non-transversal edges
            {1}{2}
        }{ % Lower non-transversal edges
            {2}{3}
        }{ % Transversal edges
            \pmdEmpty
        }
    }
}
```



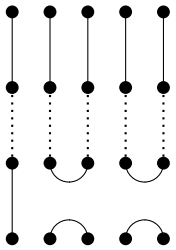
`decorate before` `decorate after` same as `decorate before` and `decorate after` in bricks but these are drawn before/after all bricks are drawn. For the drawing order of a diagram and product diagram, see Section 5.5.3.

5.5.2 For product diagrams only

The following options can only be used for product diagrams (`\pmdProduct`).

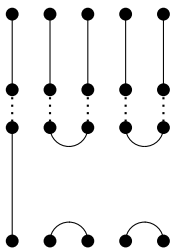
`diagram sep` specifies the vertical separation between the bottom row of vertices in top diagram and the top row of vertices in bottom diagram. Also controls the length of the added edges. Default is one.

```
\pmdProduct[ % Options
    diagram sep=2
]{ % Added edges
    {1}{5}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



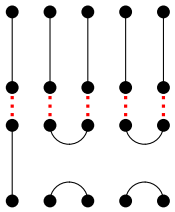
`vertex sep` increases the vertical separation between the bottom row of vertices in top diagram and the top row of vertices in bottom diagram without altering the length of the added edges. Useful for things when adjusting the `row sep` of the bottom diagram for example. Default is zero.

```
\pmdProduct[ % Options
    vertex sep=1
]{ % Added edges
    {1}{5}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5,
            row sep=3
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



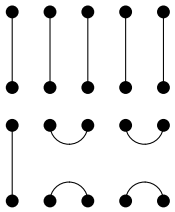
`edges added` passes through options to the `\draw` command of the added edges. The default is no argument.

```
\pmdProduct[ % Options
    edges added={red,very thick}
]{ % Added edges
    {1}{5}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`edges added phantom` This flag allows for the added edges to be hidden using the `phantom` command. Simply wraps the edge drawing macros with a `phantom` command. The internal values used are `1` for true and `0` for false. Hence to negate use `edges added phantom=0`. The default is false.

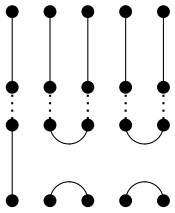
```
\pmdProduct[ % Options
    edges added phantom
]{ % Added edges
    {1}{5}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



`edges added uncover` allows for the added edges to be uncovered in a `beamer` presentation[5] using the command `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the added edge drawing macros. The default is `1-`.

[5] Make sure to use a `fragile` frame.

```
\pmdProduct[ % Options
    edges added uncover={2,4-7,9-}
]{ % Added edges
    {1}{5}
}{ % Top diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            \pmdEmpty
        }{ % Lower non-transversal edges
            \pmdEmpty
        }{ % Transversal edges
            {1}{1}
            {2}{2}
            {3}{3}
            {4}{4}
            {5}{5}
        }
    }
}{ % Bottom diagram
    { % Brick 1
        [ % Options
            degree=5
        ]{ % Upper non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Lower non-transversal edges
            {2}{3}
            {4}{5}
        }{ % Transversal edges
            {1}{1}
        }
    }
}
```



5.5.3 Drawing order

For the drawing order of a brick, see Section 5.1.6. For clarity, this is the default drawing order for a diagram:

1. `diagram/decorate before`
2. Brick loop:
   (a) `brick/decorate before`
   (b) Draws brick

(c) `brick/decorate after`

3. `diagram/decorate after`

And this is default drawing order for a product diagram:

1. `diagram/decorate before`
2. Draws added edges
3. Top diagram:
   (a) Brick loop:
      i. `brick/decorate before`
      ii. Draws brick
      iii. `brick/decorate after`
4. Bottom diagram:
   (a) Brick loop:
      i. `brick/decorate before`
      ii. Draws brick
      iii. `brick/decorate after`
5. `diagram/decorate after`

5.6 Global options

The following are all of the commands that change all diagrams after the command is given.

`\pmdSetDefault` sets the default style of all diagrams and has the syntax `\pmdSetDefault{<keys>}`, where the input keys are one of the following:

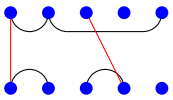
- `brick` – see Section 5.1,
- `NTedges` – non-transversal edges, see Section 5.2.1,
- `Tedges` – transversal edges, see Section 5.2.2,
- `brace` – see Section 5.3,
- `bracket` – see Section 5.4,
- `diagram` – see Section 5.5.

Only the keys (both families, for example brick, and individual keys, for example degree) that are to be changed need to be given. Unspecified keys will retain their existing default values.

The follow is an example use:

```
\pmdSetDefault{% Set default keys
    brick={% Brick keys
        degree=5,
        vertices=blue
    },
    Tedges={% Transversal edge keys
        edge draw=red
    }
}

\pmdBrick[ % Options
    % degree=5% Not needed anymore
]{ % Upper non-transversal edges
    {1}{2}
    {2}{5}
}{ % Lower non-transversal edges
    {1}{2}
    {3}{4}
}{ % Transversal edges
    {1}{1}
    {3}{4}
}
```



By default, any keys that are provided in a specific instance of a diagram completely override the default value of those keys for that specific instance of a diagram. However, the following keys work differently in that the default value is loaded and then the specific instance of that key in a diagram is appended to the list of options given:

- `brick` keys:
  - `vertices`,
  - `edges`,
  - `edges non-transversal`,
  - `edges upper`,
  - `edges lower`,
  - `edges transversal`.
- `diagram` keys:
  - `edges added`.

This means that, for example, if the default `brick/edges` key is set to `red` and for a specific diagram the `brick/edges` key is set to `thick`, the resulting edge will be `{red, thick}`. In short, this works in the default way that `tikz` handles multiple options, including repeated options, given to a `\draw` command. Options are appended in the following order:

1. Global default value,
2. Diagram value,
3. Brick value,

4. Edge value (if key is for an edge).

The following keys cannot be changed using the `\pmdSetDefault` command:

- `brick` keys:
  - `vertices options`,
  - `ghost options`,
  - `labels bottom options`,
  - `labels top options`,
  - `no dots`,
  - `brace`.
- `NTedges` keys:
  - `options`.
- `Tedges` keys:
  - `height`,
  - `options`.
- `brace` keys:
  - `label`,
  - `options`.
- `bracket` keys:
  - `label`,
  - `options`.
- `diagram` keys:
  - `dots options`.

Instead their default commands should be changed as a work around. For example, the key `labels bottom options` can be set to a different default value by finding the corresponding default command in the style file, `\pmdraw@drawLabelBottomDefault` for this example, and redefining that command to the new default value sought.

It should be noted that for some of these keys the above mentioned work around does not work. An example of this is for the `height` key as this is defined more complexly than just via a single command.

Due to the use of the `xkeyval` package which only has the ability to save one value for a key, it is not possible to have more than one default option defined. A substantial rewrite of the package would be needed to have an arbitrary number of predefined styles that can quickly be used.

`\pmdSetToOriginalDefault` sets the style of all diagrams to the default original style of the package. In effect it undoes any and all changes made with `\pmdSetDefault`.

6 Future features/work

I have attempted to make most use cases easy to use with dedicated functionality that does not require significant LaTeX knowledge whilst at the same time provide more advanced interfaces to allow for hopefully almost complete customisability for those who put in the effort.

Having said that, please do let me know if there are features or improvements you will like to see added.

The following are some of the aims for the future for this package:

1. Fix bugs.
2. Add warning and error messages to log.
3. Allow all brick keys to work as diagram keys that apply to all bricks in a diagram. This will require setting up a flag for each key so that the diagram key sets the brick key but the brick key is not reset after drawing the first brick. This process also will need to be able to handle when a single brick has a custom special different key value and that this is reset to the digram key value and not the global default key value.
4. Allow appropriate diagram keys to work as brick keys. Notably the `grid` key.
5. The creation of user profiles that allows for fast and efficient setting of default key values that are different to the package defaults. Hence within a diagram, there can be a number of different diagram styles that can be accessed with one command. That is, in the options for a diagram, have this profile key.
6. Allow for adjusting the horizontal spacing of vertices. Need to think about making that a simple scale factor so that integer interface for edge positions is the same but then hooks will have very messy coordinates or keep it as absolute coordinates with messy edges. Alternatively, do both, allowing the user to choose which option to do and somehow distinguish between the two in the background.
7. Create `pmdMirror` command that mirrors all upper non-transversals into lower non-transversals.
8. Create `invert` key for bricks, diagrams and product diagrams that inverts everything. Probably easiest done with a `scope` environment around everything and `yscale=-1`.
9. For product diagram, have interface where you input diagram instead of bricks so that you have access to diagram options.
10. Create a transversal edge key `identity` so that all edges in between the two values are straight vertical edges. Thus eliminating the need to write out each edge manually.
11. Create `shift` key for edges that shifts horizontal position.
12. Decouple the uncover of labels from `vertices top uncover`/`vertices bottom uncover`.

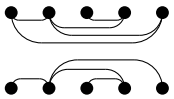
13. Decouple the phantom of labels from `vertices top phantom`/`vertices bottom phantom`.

14. For all of the keys in Section 5.5.2, create a top and bottom version to be used in `pmdProductTriple`.

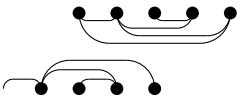
7 Examples

This section is simply a collection of example diagrams. The code for them can be found in the `examples` folder. Perhaps you will find them useful as a starting point for your diagrams or to see what this package is capable of.

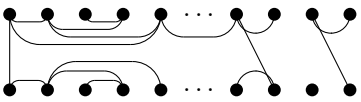
Example 1



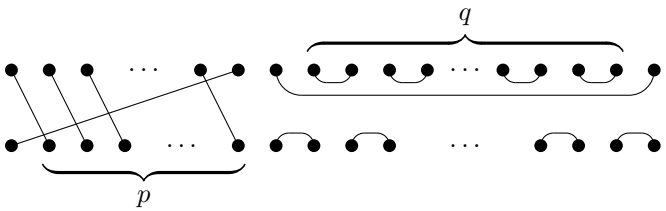
Example 2



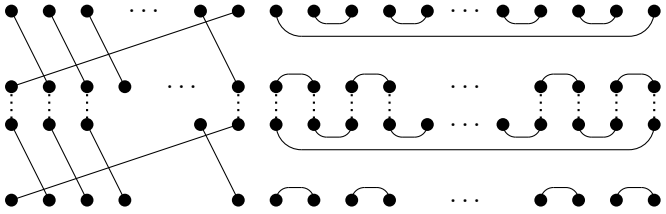
Example 3



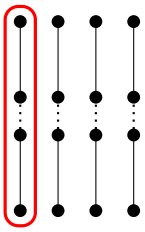
Example 4



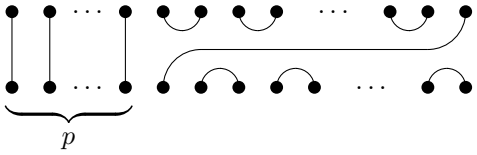
Example 5



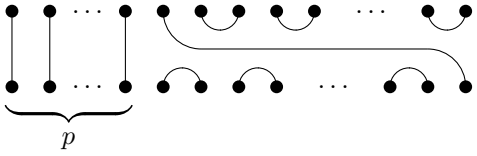
Example 6



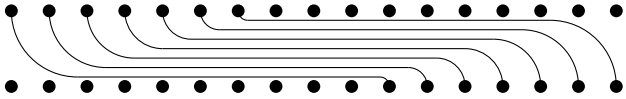
Example 7



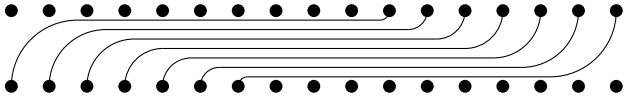
Example 8



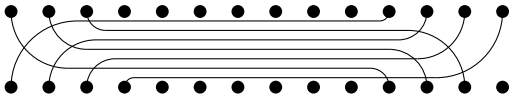
Example 9



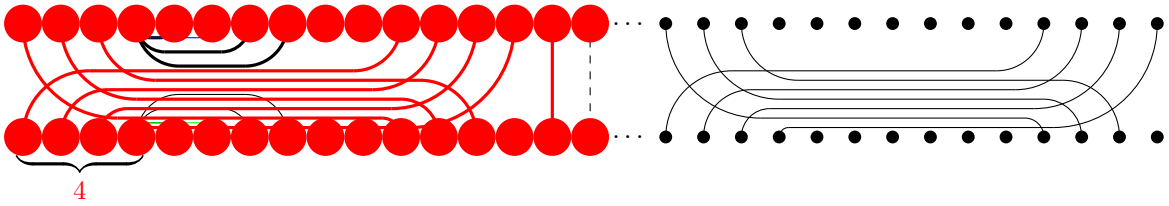
Example 10



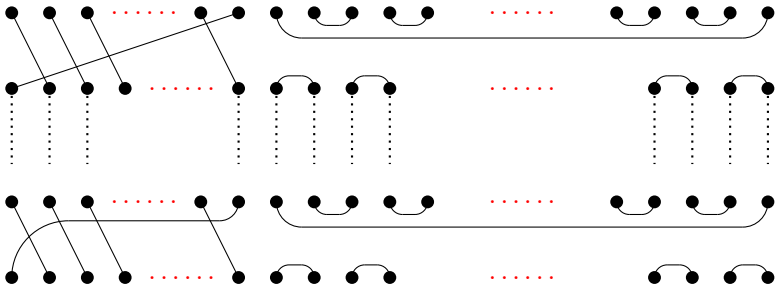
Example 11



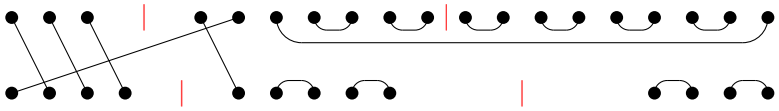
Example 12



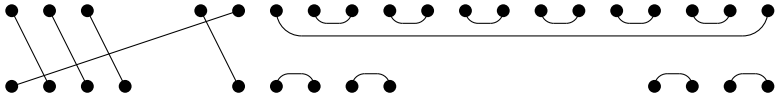
Example 13



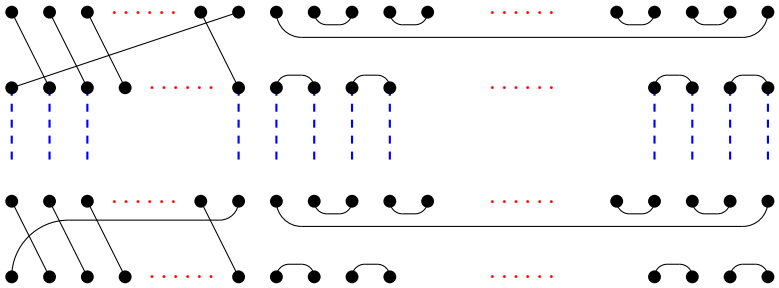
Example 14



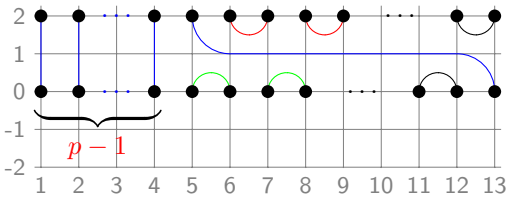
Example 15



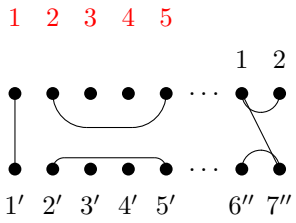
Example 16



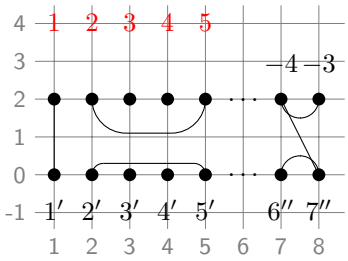
Example 17



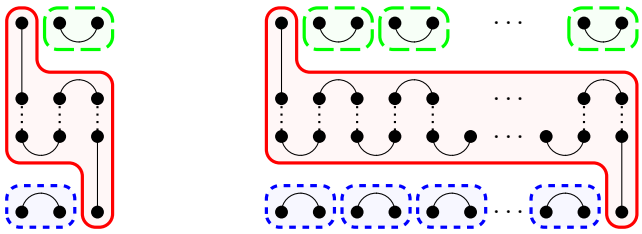
Example 18



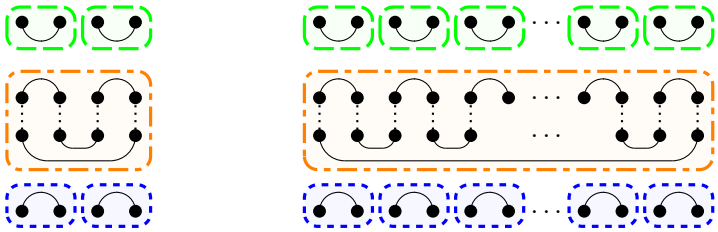
Example 19



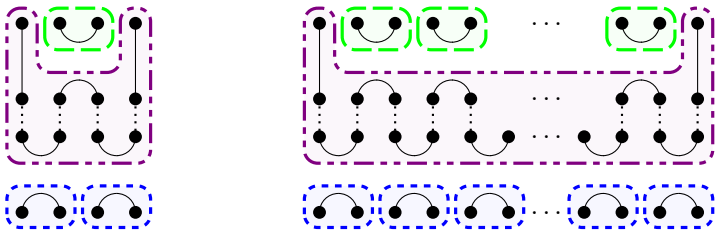
Example 20



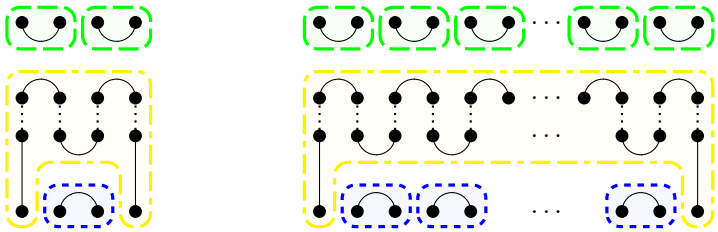
Example 21



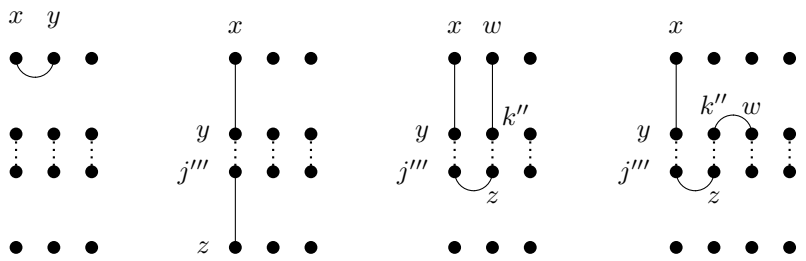
Example 22



Example 23



Example 24



Example 25