

The drawmatrix package

Elmar Peise
peise@aices.rwth-aachen.de

2015/11/26 v1.1.0

Abstract

drawmatrix provides macros to visually represent matrices. Various options allow to change the visualizations, e.g., drawing rectangular, triangular, or banded matrices.

Contents

1	Introduction	1
2	Drawing Matrices	1
2.1	Size	2
2.2	Shape	2
2.2.1	Triangular and Trapezoidal Matrices	2
2.2.2	Banded Matrices	3
2.2.3	Diagonal Matrices	3
2.3	Colors and Style	3
2.4	The Bounding Box	4
2.5	Position of the Label and Baseline	4
3	Changing Defaults	5
4	Externalization	5
5	Implementation	5
5.1	Package: TikZ	5
5.2	If for externalization	6
5.3	Key Declarations and Defaults	6
5.4	User Macros	8

1 Introduction

In many situations, visual representations of matrices facilitate the understanding of linear algebra properties, relations, and operations enormously. This package provides simple tools to bring such representations to L^AT_EX. For instance,

$$\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline X \\ \hline \end{array} + \begin{array}{|c|} \hline X \\ \hline \end{array} \begin{array}{|c|} \hline B \\ \hline \end{array} = \begin{array}{|c|} \hline C \\ \hline \end{array}$$

is typeset as follows:

```

 $\$$ 
\drawmatrix[upper]A \;
\drawmatrix[width=.5]X +
\drawmatrix[width=.5]X \;
\drawmatrix[upper, size=.5, bbox height=1]B =
\drawmatrix[width=.5]C
 $\$$ 

```

2 Drawing Matrices

`\drawmatrix` `\drawmatrix[<options>]{<label>}` draws a matrix labeled *<label>*: `\drawmatrix A` produces $\begin{array}{|c|} \hline A \\ \hline \end{array}$. The *<options>*, which modify various aspects of drawn matrix through PGF's key-value system, are introduced in the following sections.

By default, the matrix is centered around its label, which is aligned with the surrounding text. The label is typeset in the surrounding mode and style.

```

 $\$$ \drawmatrix A$:  $\begin{array}{|c|} \hline A \\ \hline \end{array}$ 

{\bf \drawmatrix A}:  $\begin{array}{|c|} \hline \mathbf{A} \\ \hline \end{array}$ 

{\large \drawmatrix A}:  $\begin{array}{|c|} \hline A \\ \hline \end{array}$ 

```


In equations, parentheses (spanned with `\left` and `\right`), subscripts, and superscripts naturally extend to the drawn shape: $\left(\begin{array}{|c|} \hline A \\ \hline \end{array} + \begin{array}{|c|} \hline B \\ \hline \end{array}^{-1} \right) \begin{array}{|c|} \hline C \\ \hline \end{array}$.

Used in matrix products such as $\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline B \\ \hline \end{array}$, a little space (`\;`) helps to yield a more natural result: $\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline B \\ \hline \end{array}$.

2.1 Size

By default, matrices are 1×1 large in terms of TikZ units. The width and height of a matrix are set through, respectively, `width= $\langle expr \rangle$` and `height= $\langle expr \rangle$` . A width or height of 0 are useful to represent vectors:

`\drawmatrix[width=0]A: A`



`size= $\langle expr \rangle$` sets both the width and height to $\langle expr \rangle$, resulting in a square matrix.


2.2 Shape

By default matrices are rectangular.


2.2.1 Triangular and Trapezoidal Matrices

Lower and upper triangular matrices are obtained by, respectively, setting the keys `lower` and `upper`. Hereby, non-square matrices become trapezoidal.

`\drawmatrix[lower]L:`




`\drawmatrix[upper, width=1.5]U:`




2.2.2 Banded Matrices

Matrices are drawn as banded through the key `banded`. The band width, i.e., the horizontal/vertical extent from the diagonal, is specified through `bandwidth= $\langle expr \rangle$` (default: 0.3);

`\drawmatrix[banded]B:`




`\drawmatrix[bandwidth=.5]B:`




Banding for the lower and upper part of the matrices can be specified separately through `lower banded` and `upper banded`. Separate bandwidths are set through `lower bandwidth= $\langle expr \rangle$` and `upper bandwidth= $\langle expr \rangle$` :


`\drawmatrix[lower banded]B:`




`\drawmatrix[lower bandwidth=.5, upper bandwidth=.2]B:`




Banding on rectangular matrices applies to the smaller of the two dimensions:

```
\drawmatrix[banded, width=.8]B: 
```


```
\drawmatrix[upper banded, height=.7]B: 
```

banded can be combined with **lower** or **upper** to draw the intersection of both shapes.

```
\drawmatrix[banded, lower]L: 
```


2.2.3 Diagonal Matrices


diag **diag** is a shorthand for **banded** with **bandwidth=0**:

```
\drawmatrix[diag]D: 
```

2.3 Colors and Style


By default, matrices are drawn in gray and filled white. The TikZ keys **draw**=*<color>* and **fill**=*<color>* change these colors. In fact, all keys not recognized by this package are passed to the TikZ **\filldraw** command drawing the matrix.

```
\drawmatrix[fill=yellow, draw=blue]A: 
```

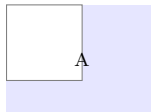
```
\drawmatrix[very thick, dashed]A: 
```

2.4 The Bounding Box

bbbox style All matrices are contained in a rectangular bounding box. To draw this bounding box (e.g., to visualize the 0 entries in the matrix), use **bbbox style**=*<style>*; this style is applied to the TikZ **\node** that is the bounding box.

```
\drawmatrix[lower, bbbox style={fill=blue!10}]L: 
```

bbbox height By default, the bounding box is just large enough to contain the matrix. Its size is changed through the keys **bbbox height**=*<expr>* and **bbbox width**=*<expr>* (or **bbbox size**=*<expr>* to set them both). The label of the matrix (and thus the alignment with respect to the surrounding text) are fixed at the center of the bounding box, while the matrix is positioned at its top-left corner.

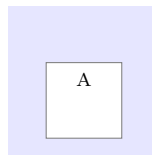
```
\drawmatrixset{bbbox style={fill=blue!10}}
\drawmatrix[bbbox width=2, bbbox height=1.5]A: 
```

The matrix can be positioned within its bounding box through `offset height= $\langle expr \rangle$` and `offset width= $\langle expr \rangle$` (or just `offset= $\langle expr \rangle$` to shift along the diagonal).

`offset`

```
\drawmatrixset{bbox style={fill=blue!10}}
```

```
\drawmatrix[bbox size=2, offset width=.5, offset height=.75]A:
```



2.5 Position of the Label and Baseline

By default, the label's `mid` is positioned at the bounding box's `center` and its `base` is used as the whole drawing's baseline. This is controlled by the keys `label anchor= $\langle anchor \rangle$` , `label pos= $\langle position \rangle$` , and `baseline= $\langle position \rangle$` . Here, $\langle position \rangle$ has to be an anchor of one of the following nodes: `bbox` (the bounding box), `matrix` (the matrix itself), or `label` (the label).

`label anchor`

`label pos`

`baseline`

```
\drawmatrixset{bbox height=1, height=.5, bbox style={fill=blue!10}}
```

```
\drawmatrix[label pos=bbox.south, label anchor=south]A:
```

```
\drawmatrix[label pos=matrix.north west]A: A
```

```
\drawmatrix[baseline=label.north]A:
```

```
\drawmatrix[baseline=bbox.south]A:
```



3 Changing Defaults

`\drawmatrixset` Specifying $\langle options \rangle$ with `\drawmatrixset{ $\langle options \rangle$ }` applies them to all following uses of `\drawmatrix` within the current scope.

```
\drawmatrixset{height=.5, lower}
```

```
$\drawmatrix A \; \drawmatrix B$:
```



`every bbox`
`every drawmatrix`
`every label`

Furthermore, *TikZ* keys for the entire picture, the bounding box, the matrix itself and the label can be set through the styles `every bbox`, `every drawmatrix`, and `every label`.

```
\drawmatrixset{every drawmatrix/.append style={rounded corners=5pt}}
```

```
$\drawmatrix A \; \drawmatrix[lower]B$:
```



4 Externalization

`\drawmatrix` behaves as any other TikZ picture, therefore when externalization is enabled, all matrix visualizations are also externalized. However, since there are usually many `\drawmatrix` pictures, each of which is very small and fast to produce, their externalization would mean a tremendous overhead. To avoid this overhead without explicitly dis- and re-enabling externalization throughout the document, `externalize=false` disables externalization for all `\drawmatrix` pictures:

```
\drawmatrixset{externalize=false}
```

5 Implementation

This section describes the implementation details of the `drawmatrix` package.

5.1 Package: TikZ

The `tikz` package is used for drawing.

```
1 \RequirePackage{tikz}
```

5.2 If for externalization

TeX `if` representing whether to explicitly disable TikZ externalization.

```
\ifdrawmatrix@externalize
```

```
2 \newif\ifdrawmatrix@externalize
```

5.3 Key Declarations and Defaults

We rely on PGF keys as much as we can.

```
3 \pgfkeys{
```

```
    Everything happens in the path /drawmatrix.
```

```
4     drawmatrix/.is family,
```

```
5     drawmatrix/.cd,
```

picture `picture` is the style for the `\tikzpicture` in which the matrix is drawn. **baseline** sets the baseline of the picture to a named coordinate of the matrix (default: base of the label).

```
6     picture/.style={},
```

```
7     baseline/.style={picture/.append style={baseline=(drawmatrix #1)}},
```

```
8     baseline=label.base,
```

bbox `bbox` is the style of the bounding box, to which `bbox` style appends keys.

bbox style `bbox style` is the style of the bounding box, to which `bbox style` appends keys.

```
9     bbox/.style={},
```

```
10    bbox style/.style={bbox/.append style={#1}},
```

<code>bbox height</code>	<code>bbox height</code> and <code>bbox width</code> don't have default values. <code>bbox size</code> sets them
<code>bbox width</code>	both to the same value.
<code>bbox size</code>	<pre> 11 bbox height/.initial, 12 bbox width/.initial, 13 bbox size/.style={bbox height=#1, bbox width=#1}, </pre>
<code>offset height</code>	<code>offset height</code> and <code>offset width</code> are 0 by default. <code>offset</code> sets them both to
<code>offset width</code>	the same value.
<code>offset</code>	<pre> 14 offset height/.initial=0, 15 offset width/.initial=0, 16 offset/.style={offset height=#1, offset width=#1}, </pre>
<code>height</code>	<code>width</code> and <code>height</code> are 1 (TikZ unit) by default. <code>size</code> sets them both to the same
<code>width</code>	value.
<code>size</code>	<pre> 17 height/.initial=1, 18 width/.initial=1, 19 size/.style={height=#1, width=#1}, </pre>
<code>lower bandwidth</code>	The <code>lower bandwidth</code> and <code>upper bandwidth</code> don't have default values.
<code>upper bandwidth</code>	<code>bandwidth</code> sets them both to the same value.
<code>bandwidth</code>	<pre> 20 lower bandwidth/.initial, 21 upper bandwidth/.initial, 22 bandwidth/.style={lower bandwidth=#1, upper bandwidth=#1}, </pre>
<code>lower banded</code>	<code>lower banded</code> and <code>upper banded</code> are shortcuts to set the corresponding band-
<code>upper banded</code>	widths to the default value of 0.3 (TikZ units). <code>banded</code> sets them both.
<code>banded</code>	<pre> 23 lower banded/.style={lower bandwidth=.3}, 24 upper banded/.style={upper bandwidth=.3}, 25 banded/.style={lower banded, upper banded}, </pre>
<code>lower</code>	<code>lower</code> and <code>upper</code> are implemented by setting the opposite bandwidth to 0. <code>diag</code>
<code>upper</code>	sets them both.
<code>diag</code>	<pre> 26 lower/.style={upper bandwidth=0}, 27 upper/.style={lower bandwidth=0}, 28 diag/.style={lower, upper}, </pre>
<code>externalize</code>	<code>externalize</code> sets a \TeX if (default: <code>true</code> = behave as all pictures).
	<pre> 29 externalize/.is if=drawmatrix@externalize, 30 externalize=true, </pre>
<code>label</code>	<code>label</code> is the style for the label. <code>label pos</code> sets the label at a named coordinate of
<code>label pos</code>	the matrix (default: center of the bounding box). <code>label anchor</code> sets the label's
<code>label anchor</code>	<code>anchor</code> (default: in the middle).
	<pre> 31 label/.style={}, 32 label pos/.style={label/.append style={at=(drawmatrix #1)}}, 33 label pos=bbox.center, 34 label anchor/.style={label/.append style={anchor=#1}}, 35 label anchor=mid, </pre>

Unknown keys are collected in `/drawmatrix/drawmatrix`.

```

36   drawmatrix/.style={},
37   .unknown/.code={%
38       \let\@currname\pgfkeyscurrentname%
39       \let\@currval\pgfkeyscurrentvalue%
40       \ifx#1\pgfkeysnovalue\pgfkeysalso{
41           drawmatrix/.append style/.expand once={\@currname}
42       }\else\pgfkeysalso{
43           drawmatrix/.append style/.expand twice={%
44               \expandafter\@currname\expandafter=\@currval%
45           }
46       }\fi%
47   },

```

`every picture` The default style for matrices: `every picture` applies to all `\tikzpictures`
`every bbox` the matrices are drawn in, `every bbox` applies to all bounding boxes,
`every drawmatrix` `every drawmatrix` applies to the matrices themselves, and `every label` applies
`every label` to the labels.

```

48   every picture/.style={},
49   every bbox/.style={
50       name=drawmatrix bbox,
51       inner sep=0,
52   },
53   every drawmatrix/.style={
54       fill=white,
55       draw=gray,
56   },
57   every label/.style={
58       name=drawmatrix label,
59       outer sep=0,
60       inner sep=0,
61   }
62 }

```

5.4 User Macros

`\drawmatrixset` as a simple shortcut like `\tikzset`.

`\drawmatrixset`

```

63 \newcommand\drawmatrixset[1]{\pgfqkeys{/drawmatrix}{#1}}

```

Here we go, the main thing: `\drawmatrix`. First, check if we are in math mode and apply the options.

`\drawmatrix`

```

64 \newcommand\drawmatrix[2][]{\ifmode
65     \ifmmode\def\@labeltext{##2}\else\def\@labeltext{#2}\fi%
66     \drawmatrixset{#1}%

```


Disable externalization if `externalize=false`.

```

67 \ifdrawmatrix@externalize\else%
68 \ifx\tikz@library@external@loaded\undefined\else%
69 \tikzset{external/export=false}%
70 \fi%
71 \fi%
```

Extract the sizes from the PGF keys and parse width, height, the minimum dimension and zero for comparison purposes.

```

72 \pgfkeys{/drawmatrix}{
73   height/.get=\@height,
74   width/.get=\@width,
75   lower bandwidth/.get=\@lowerbandwidth,
76   upper bandwidth/.get=\@upperbandwidth,
77   offset height/.get=\@offsetheight,
78   offset width/.get=\@offsetwidth,
79   bbox height/.get=\@bboxheight,
80   bbox width/.get=\@bboxwidth,
81 }%
82 \pgfmathsetmacro\@height{\@height + 0.0}%
83 \pgfmathsetmacro\@width{\@width + 0.0}%
84 \pgfmathsetmacro\@minsize{min(\@width, \@height)}%
85 \pgfmathsetmacro\@zero{0.0}%
```

Prepare the band widths: First, if the matrix is not banded, the bandwidth is set to the smaller matrix dimension. Then, the band width is limited by this smaller dimension.

```

86 \expandafter\ifx\@lowerbandwidth\pgfkeysnovalue%
87 \edef\@lowerbandwidth{\@minsize}%
88 \fi%
89 \expandafter\ifx\@upperbandwidth\pgfkeysnovalue%
90 \edef\@upperbandwidth{\@minsize}%
91 \fi%
92 \pgfmathsetmacro\@lowerbandwidth{min(\@minsize, \@lowerbandwidth)}%
93 \pgfmathsetmacro\@upperbandwidth{min(\@minsize, \@upperbandwidth)}%
```

Set the default bounding box size.

```

94 \expandafter\ifx\@bboxheight\pgfkeysnovalue%
95 \pgfmathsetmacro\@bboxheight{\@height + \@offsetheight}%
96 \fi%
97 \expandafter\ifx\@bboxwidth\pgfkeysnovalue%
98 \pgfmathsetmacro\@bboxwidth{\@width + \@offsetwidth}%
99 \fi%
```

Begin with (drawing) the path for the bounding box.

```

100 \begin{tikzpicture}[/drawmatrix/every picture, /drawmatrix/picture]
101   \node[/drawmatrix/every bbox, /drawmatrix/bbox,
102     minimum height=\@bboxheight cm,
103     minimum width=\@bboxwidth cm] {};
```

Whether needed or not, declare all matrix corners.

```

104      \path (drawmatrix bbox.north west)
105          ++(\@offsetwidth, -\@offsetheight)
106          ++(.5\pgflinewidth, -.5\pgflinewidth)
107          coordinate (drawmatrix north west)
108          ++(\@width, 0)
109          +(-\@minsize + \@upperbandwidth, 0)
110          coordinate (drawmatrix north)
111          +(0, -\@minsize + \@upperbandwidth)
112          coordinate (drawmatrix east)
113          ++(0, -\@height)
114          coordinate (drawmatrix south east)
115          ++(-\@width, 0)
116          +(\@minsize - \@lowerbandwidth, 0)
117          coordinate (drawmatrix south)
118          +(0, \@minsize - \@lowerbandwidth)
119          coordinate (drawmatrix west);

```

Now, draw only what is needed of the matrix. Otherwise path modifications (e.g., such as rounded corners) might not work.

```

120      \filldraw[/drawmatrix/every drawmatrix, /drawmatrix/drawmatrix]
121          (drawmatrix north west)
122          \ifx\@upperbandwidth\@zero
123              \ifx\@width\@minsize\else -- (drawmatrix north) \fi
124              \ifx\@height\@minsize\else -- (drawmatrix east) \fi
125          \else
126              -- (drawmatrix north)
127              \ifx\@upperbandwidth\@minsize\else -- (drawmatrix east) \fi
128          \fi
129          -- (drawmatrix south east)
130          \ifx\@lowerbandwidth\@zero
131              \ifx\@width\@minsize\else -- (drawmatrix south) \fi
132              \ifx\@height\@minsize\else -- (drawmatrix west) \fi
133          \else
134              -- (drawmatrix south)
135              \ifx\@lowerbandwidth\@minsize\else -- (drawmatrix west) \fi
136          \fi
137          -- cycle;

```

Add an invisible node the size of the matrix.

```

138      \node[minimum height=\@height cm, minimum width=\@width cm,
139          anchor=north west] at (drawmatrix north west)
140          (drawmatrix matrix) {};

```

The label.

```

141      \node[/drawmatrix/every label, /drawmatrix/label] {\@labeltext};
142      \end{tikzpicture}%
143  }}

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B		<code>\every_label</code> 5 , 48	O
<code>\banded</code> 3 , 23	<code>\every_picture</code> 48	<code>\offset</code> 4 , 14	
<code>\bandwidth</code> 3 , 20	<code>\externalize</code> 5 , 29	<code>\offset_height</code> 4 , 14	
<code>\baseline</code> 4 , 6		<code>\offset_width</code> 4 , 14	
<code>\bbox</code> 9	H		
<code>\bbox_height</code> 4 , 11	<code>\height</code> 2 , 17	P	
<code>\bbox_size</code> 4 , 11		<code>\picture</code> 6	
<code>\bbox_style</code> 4 , 9	I		
<code>\bbox_width</code> 4 , 11	<code>\ifdrawmatrix@externalize</code>	S	
 2	<code>\size</code> 2 , 17	
D	L	U	
<code>\diag</code> 3 , 26	<code>\label</code> 31	<code>\upper</code> 2 , 26	
<code>\drawmatrix</code> 1 , 64	<code>\label_anchor</code> 4 , 31	<code>\upper_banded</code> 3 , 23	
<code>\drawmatrixset</code> 5 , 63 , 66	<code>\label_pos</code> 4 , 31	<code>\upper_bandwidth</code> 3 , 20	
E	<code>\lower</code> 2 , 26	W	
<code>\every_bbox</code> 5 , 48	<code>\lower_banded</code> 3 , 23	<code>\width</code> 2 , 17	
<code>\every_drawmatrix</code> 5 , 48	<code>\lower_bandwidth</code> 3 , 20		

Change History

v1.0.0	(label placement for vectors) . . . 1
General: Initial Version 1	v1.0.2
v1.0.1	General: Bugfix: Bbox had a
General: Bugfix: Collapsible bbox	linewidth/2 offset 1