

The `drawmatrix` package

Elmar Peise
`peise@aices.rwth-aachen.de`

2017/06/05 v1.5.0

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	2
2	Drawing Matrices	2
2.1	Size	3
2.2	Shape	3
2.2.1	Triangular and Trapezoidal Matrices	3
2.2.2	Banded Matrices	4
2.2.3	Diagonal Matrices	5
2.2.4	Super- and Subscripts	5
2.3	Colors and Style	5
2.4	The Bounding Box	6
2.5	Coordinate System Transformations	7
2.6	Position of the Label and Baseline	7
3	Changing Defaults	8
4	Externalization	9
5	Implementation	9
5.1	Package: <code>TikZ</code>	9
5.2	If for externalization	9
5.3	Key Declarations and Defaults	9
5.4	Hooks	12
5.5	Code Keys	12
5.6	User Macros	15
	Index	17

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,

$$\boxed{A} \boxed{X} + \boxed{X} \boxed{B} = \boxed{C}$$

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[options]{label}`

Draws a matrix labeled *label*. The optional *options*, which modify various aspects of drawn matrix through PGF's key-value system, are introduced in the following sections.

The label is typeset in the surrounding mode and style.

```
$$\drawmatrix A$
{\bfseries\drawmatrix A}
{\large\drawmatrix A}
```

$$\boxed{A} \quad \boxed{\mathbf{A}} \quad \boxed{\text{A}}$$

In equations, parentheses (spanned with `\left` and `\right`), subscripts, and superscripts naturally extend to the drawn shape:

```
$$\left(
\drawmatrix A_i +
\drawmatrix B^{-1}
\right)
\drawmatrix C$
```

$$\left(\boxed{A}_i + \boxed{B}^{-1} \right) \boxed{C}$$

Used in matrix products, a little space (`\;`) helps to yield a more natural result:

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

$$\begin{array}{|c|c|} \hline A & B \\ \hline A & B \\ \hline \end{array}$$

`/drawmatrix[label text=text]` (no default, initially *label*)
 Stores the label text. It overwrites `\drawmatrix`^{→P. 2}'s *label* argument.



2.1 Size

`/drawmatrix/height=dimension` (no default, initially 1)
`/drawmatrix/width=dimension` (no default, initially 1)

Width and height of the drawmatrix in TikZ's coordinate system **canvas**. May be given in units such as **em** or **cm**.



A width or height of 0 are useful to represent vectors:



`/drawmatrix/size=dimension` (style, no default)
 Shortcut for setting both `/drawmatrix/height` and `/drawmatrix/width` to the same *dimension*, resulting in a square matrix.

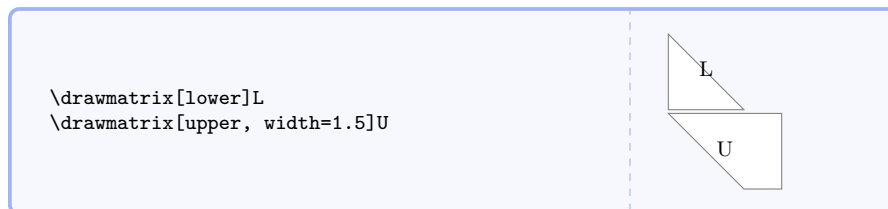
2.2 Shape

By default matrices are rectangular.

2.2.1 Triangular and Trapezoidal Matrices

`/drawmatrix/lower` (style, no value, initially unset)
`/drawmatrix/upper` (style, no value, initially unset)

Result in, respectively, lower- and upper-triangular matrices. Non-square matrices become trapezoidal.



2.2.2 Banded Matrices

`/drawmatrix/lower banded` (style, no value, initially unset)
`/drawmatrix/upper banded` (style, no value)

Draw matrices as banded with bandwidth 0.3.

```
\drawmatrix[lower banded]B
\drawmatrix[upper banded]B
```



`/drawmatrix/banded` (style, no value)

Shortcut for setting both `/drawmatrix/lower banded` and `/drawmatrix/upper banded`.

```
\drawmatrix[banded]B
```



`/drawmatrix/lower bandwidth=<dimension>` (no default, initially empty)
`/drawmatrix/upper bandwidth=<dimension>` (no default, initially empty)

The bandwidths, i.e., the horizontal/vertical extent from the diagonal.

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



`/drawmatrix/bandwidth=<dimension>` (style, no default)

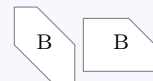
Shortcut for setting both `/drawmatrix/lower bandwidth` and `/drawmatrix/upper bandwidth`.

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



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

```
\drawmatrix[banded, width=.8]B
\drawmatrix[upper banded, height=.7]B
```



`/drawmatrix/banded` can be combined with `/drawmatrix/lower→P.3` or `/drawmatrix/upper→P.3` to draw the intersection of both shapes.

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



2.2.3 Diagonal Matrices

`/drawmatrix/diag` (style, no value)

Shortcut for `/drawmatrix/bandwidth→P.4=0`.

`\drawmatrix[diag]D`



2.2.4 Super- and Subscripts

`/drawmatrix/label base=text` (no default, initially empty)

Defines the label to be centered in the drawmatrix, and to which the actual `/drawmatrix/label text→P.3` is aligned. This feature is useful to, e.g., draw centered labels with exponents:

`$\drawmatrix[size=.5]{A^T}$`
`$\drawmatrix[size=.5, label base=A]{A^T}$`



`/drawmatrix/label base anchor=anchor` (no default, initially `base west`)

The anchor of the `/drawmatrix/label text→P.3` with respect to the `/drawmatrix/label base`.

`$\drawmatrix[size=.5, label base=A, label 2`
`{base anchor=base east}{^0_1A}$`



`/drawmatrix/exponent=text` (style, no default)

Shortcut to add an exponent to matrix without offsetting the label. It sets the `/drawmatrix/label base` to the current `/drawmatrix/label text→P.3` and adds the exponent `text` to `/drawmatrix/label text→P.3`.

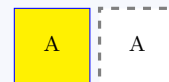
`$\drawmatrix[size=.5, exponent=T]A$`



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

All matrices are contained in a rectangular bounding box.

/drawmatrix/bbox (style, no default, initially empty)

Add options to the TikZ `\node` that is the bounding box.. This is useful to, e.g., to visualize the 0 entries in the matrix:

```
\drawmatrix[lower, bbox/.append >
{style={fill=blue!10}}]L
```



/drawmatrix/bbox style={⟨style⟩} (style, no default)

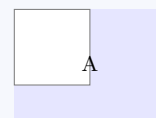
Shortcut for `/drawmatrix/bbox/.append style=⟨style⟩`.

/drawmatrix/bbox height=⟨dimension⟩ (no default, initially empty)

/drawmatrix/bbox width=⟨dimension⟩ (no default, initially empty)

Explicitly set the height and width of the bounding box. If unset, the bounding box is just large enough to contain the matrix. The label of the matrix (and thus the alignment with respect to the surrounding text) are by default fixed at the center¹ of the bounding box, while the matrix is positioned at its top-left corner.

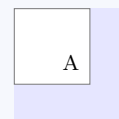
```
\drawmatrixset{bbox style={fill=blue!10}}
\drawmatrix[bbox width=2, bbox height=1.5]A
```



/drawmatrix/bbox size=⟨dimension⟩ (style, no default)

Shortcut for setting `/drawmatrix/bbox height` and `/drawmatrix/bbox width` to the same value.

```
\drawmatrixset{bbox style={fill=blue!10}}
\drawmatrix[bbox size=1.5]A
```



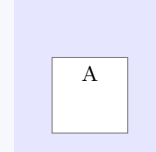
/drawmatrix/offset height=⟨dimension⟩ (no default, initially 0)

/drawmatrix/offset width=⟨dimension⟩ (no default, initially 0)

Sets the vertical and horizontal offset of the drawn matrix within its bounding box.

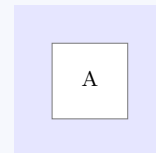
¹See `/drawmatrix/label anchor`→P. 7.

```
\drawmatrixset{bbox style={fill=blue!10}}
\drawmatrix[bbox size=2, offset width=.5, 2
  \offset height=.75]A
```



/drawmatrix/offset= $\langle dimension \rangle$ (style, no default)
 Shortcut for setting **/drawmatrix/offset height**^{→ P.6} and **/drawmatrix/bbox width**^{→ P.6} to the same value.

```
\drawmatrixset{bbox style={fill=blue!10}}
\drawmatrix[bbox size=2, offset=.5]A
```



2.5 Coordinate System Transformations

/drawmatrix/scale= $\langle factor \rangle$ (style, no default)
 Scales all dimensions passed to a matrix. Can be used repeatedly to multiply scales

```
\drawmatrix[scale=.6]A
\drawmatrix[scale=.6, width=.5]B
\drawmatrix[scale=.7, scale=.7]B
```



/drawmatrix/x= $\langle value \rangle$ (style, no default)
/drawmatrix/y= $\langle value \rangle$ (style, no default)
 Define the coordinate system for all unit-less dimensions.

```
\drawmatrix[x=.6cm, y=.4cm]A
\drawmatrix[x=.6cm, y=.4cm, width=1cm]B
```



2.6 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.

/drawmatrix/label anchor= $\langle anchor \rangle$ (style, no default, initially mid)
 Set the anchor of label's TikZ **\node**.

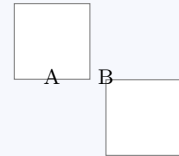
```
\drawmatrix[label anchor=north]A
```



`/drawmatrix/label pos=<position>` (style, no default, initially `bbox.center`)

Define the position of the label's TikZ `\node` within the picture. The following nodes and their anchors are available: `bbox` (the bounding box) and the `matrix` (matrix itself).

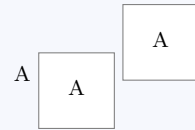
```
\drawmatrix[label pos=bbox.south]A
\drawmatrix[label pos=matrix.north west]B
```



`/drawmatrix/baseline=<position>` (style, no default, initially `label.base`)

Specify how the picture is vertically aligned with the surrounding text's baseline. Options are the same anchors as for `/drawmatrix/label pos` and anchors of `label` (the label).

```
A
\drawmatrix[baseline=label.north]A
\drawmatrix[baseline=bbox.south]A
```



3 Changing Defaults

`\drawmatrixset{<options>}`

Applies options to to all following uses of `\drawmatrix`^{P. 2} within the current scope.

```
\drawmatrixset{height=.5, lower}
$\drawmatrix A \; \drawmatrix B$
```



<code>/drawmatrix/every picture</code>	(style, no value)
<code>/drawmatrix/every bbox</code>	(style, no value)
<code>/drawmatrix/every drawmatrix</code>	(style, no value)
<code>/drawmatrix/every label</code>	(style, no value)
<code>/drawmatrix/every label</code>	(style, no value)

Settings for all `drawmatrix` pictures, bounding boxes, matrices, and labels. Options should be added not with `/.style=` but with `/.append style=` to avoid messing with internals.

```
\drawmatrixset{every drawmatrix/.append >
{style={rounded corners=5pt}}}
$\drawmatrix A \; \drawmatrix[lower]B$
```



4 Externalization

`\drawmatrix→P.2` behaves as any other TikZ picture, therefore when externalization is enabled, all matrix visualizations are also externalized. However, since there are usually many `\drawmatrix→P.2` pictures, each of which is very small and fast to produce, their externalization would mean a tremendous overhead.

`/drawmatrix/externalize=true|false` (no default, initially `true`)
Explicitly disables externalization for all `\drawmatrix→P.2` pictures. It does not enable externalization.

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.

```
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 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   path/.style={},
8   baseline/.style={picture/.append style={baseline=(drawmatrix #1)}},
9   scale/.style={path/.append style={scale=#1}},
10  x/.style={path/.append style={x=#1}},
11  y/.style={path/.append style={y=#1}},
12  baseline=label.base,
  bbox is the style of the bounding box, to which bbox style appends keys.
13  bbox/.style={},
14  bbox style/.style={bbox/.append style={#1}},
```

`bbox height` and `bbox width` don't have default values. `bbox size` sets them both to the same value.

```
15     bbox height/.initial,
16     bbox width/.initial,
17     bbox size/.style={bbox height=#1, bbox width=#1},
```

`offset height` and `offset width` are 0 by default. `offset` sets them both to the same value.

```
18     offset height/.initial=0,
19     offset width/.initial=0,
20     offset/.style={offset height=#1, offset width=#1},
```

`width` and `height` are 1 (TikZ unit) by default. `size` sets them both to the same value.

```
21     height/.initial=1,
22     width/.initial=1,
23     size/.style={height=#1, width=#1},
```

The lower bandwidth and upper bandwidth don't have default values. `bandwidth` sets them both to the same value.

```
24     lower bandwidth/.initial,
25     upper bandwidth/.initial,
26     bandwidth/.style={lower bandwidth=#1, upper bandwidth=#1},
```

`lower banded` and `upper banded` are shortcuts to set the corresponding bandwidths to the default value of 0.3 (TikZ units). `banded` sets them both.

```
27     lower banded/.style={lower bandwidth=.3},
28     upper banded/.style={upper bandwidth=.3},
29     banded/.style={lower banded, upper banded},
```

`lower` and `upper` are implemented by setting the opposite bandwidth to 0. `diag` sets them both.

```
30     lower/.style={upper bandwidth=0},
31     upper/.style={lower bandwidth=0},
32     diag/.style={bandwidth=0},
```

`label` is the style for the label with the text `label text`. `label pos` sets the label at a named coordinate of the matrix (default: center of the bounding box). `label anchor` sets the label's anchor (default: in the middle).

```
33     label text/.initial,
34     label/.style={},
35     label pos/.style={label/.append style={at=(drawmatrix #1)}},
36     label pos=bbox.center,
37     label anchor/.style={label/.append style={anchor=#1}},
38     label anchor=mid,
```

`label base` and `label base anchor` allow to offset labels with exponents.

```
39     label base/.initial,
40     label outer/.style={},
41     label base anchor/.style={label outer/.append style={
42         anchor=#1, at=(drawmatrix label.#1)
```

```

43   }},
44   label base anchor=base west,
    exponent is a shortcut to add an exponent to the label text without using the
    label base.
45   exponent/.style={
46       label base/.expanded=\pgfkeysvalueof{/drawmatrix/label text},
47       label text/.append=^{\#1}
48   },

```

Unknown keys are collected in /drawmatrix/drawmatrix.

```

49   drawmatrix/.style={},
50   .unknown/.code={%
51       \let\dm@currname\pgfkeyscurrentname%
52       \let\dm@currval\pgfkeyscurrentvalue%
53       \ifx#1\pgfkeysnovalue\pgfkeysalso{
54           drawmatrix/.append style/.expand once={\dm@currname}
55       }\else\pgfkeysalso{
56           drawmatrix/.append style/.expand twice={%
57               \expandafter\dm@currname\expandafter=\dm@currval%
58           }
59       }\fi%
60   },

```

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

```

61   every picture/.style={},
62   every bbox/.style={
63       name=drawmatrix bbox,
64       inner sep=0
65   },
66   every drawmatrix/.style={
67       fill=white,
68       draw=gray
69   },
70   every label/.style={
71       name=drawmatrix label,
72       outer sep=0,
73       inner sep=0
74   },
75   every node/.style={
76       name=drawmatrix matrix,
77       outer sep=0,
78       inner sep=0,
79       anchor=north west,
80       at=(drawmatrix north west)
81   },

```

`externalize` sets a `TeX` if (default: `true` = behave as all pictures).

```

82   externalize/.is if=drawmatrix@externalize,
83   externalize=true,

```

5.4 Hooks

Hooks for inserting code at various points of the render.

```

84   pre code/.code={},
85   post setup code/.code={},
86   pre coordinate code/.code={},
87   post coordinate code/.code={},
88   pre draw code/.code={},
89   post draw code/.code={},
90   pre label code/.code={},
91   post label code/.code={},
92   post code/.code={},

```

5.5 Code Keys

The following keys contain the code that construct the drawmatrix.

Prepare the label text and, if needed label outer text (for alignment). This needs to be outside the tikzpicture to properly detect math mode.

```

93   setup label code/.code={
94     \ifmmode\edef\dm@labeltext{\dm@labeltext}\fi%
95     \expandafter\ifx\dm@labelbase\pgfkeysnovalue\else%
96       \let\dm@labeltextouter\dm@labeltext%
97       \edef\dm@labelbase{%
98         \ifmmode$\dm@labelbase$\else\dm@labelbase\fi%
99       }%
100     \def\dm@labeltext{\phantom{\dm@labelbase}}%
101   \fi%
102 },

```

Disable externalization if `externalize=false`.

```

103   externalization code/.code={
104     \ifdrawmatrix@externalize\else%
105       \ifx\tikz@library@external@loaded\undefined\else%
106         \tikzset{external/export=false}%
107       \fi%
108     \fi%
109 },

```

Parse width, height, the minimum dimension and zero for comparison purposes.

```

110   setup sizes code/.code={
111     \path[/drawmatrix/path] (\dm@width, \dm@height);
112     \pgfgetlastxy\dm@width\dm@height
113     \path[/drawmatrix/path] (\dm@offsetwidth, \dm@offsetheight);
114     \pgfgetlastxy\dm@offsetwidth\dm@offsetheight
115     \pgfmathsetlengthmacro\dm@minsize{min(\dm@width, \dm@height)}
116     \pgfmathsetlengthmacro\dm@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.

```

117     \expandafter\ifx\dm@lowerbandwidth\pgfkeysnovalue
118       \def\dm@lowerbandwidth{\dm@minsize}
119     \else
120       \path[/drawmatrix/path] (\dm@lowerbandwidth, 0);
121       \pgfgetlastxy\dm@lowerbandwidth\dm@zero
122     \fi
123     \expandafter\ifx\dm@upperbandwidth\pgfkeysnovalue
124       \def\dm@upperbandwidth{\dm@minsize}
125     \else
126       \path[/drawmatrix/path] (0, \dm@upperbandwidth);
127       \pgfgetlastxy\dm@zero\dm@upperbandwidth
128     \fi
129     \pgfmathsetlengthmacro\dm@lowerbandwidth{
130       min(\dm@minsize, \dm@lowerbandwidth)
131   }
132     \pgfmathsetlengthmacro\dm@upperbandwidth{
133       min(\dm@minsize, \dm@upperbandwidth)
134   }

```

Set the default bounding box size.

```

135     \expandafter\ifx\dm@bboxheight\pgfkeysnovalue
136       \pgfmathsetlengthmacro\dm@bboxheight{
137         \dm@height + \dm@offsetheight
138       }
139     \else
140       \path[/drawmatrix/path] (0, \dm@bboxheight);
141       \pgfgetlastxy\dm@zero\dm@bboxheight
142     \fi
143     \expandafter\ifx\dm@bboxwidth\pgfkeysnovalue
144       \pgfmathsetlengthmacro\dm@bboxwidth{
145         \dm@width + \dm@offsetwidth
146       }
147     \else
148       \path[/drawmatrix/path] (\dm@bboxwidth, 0);
149       \pgfgetlastxy\dm@bboxwidth\dm@zero
150     \fi
151   },

```

Reset the bounding box and begin with (drawing) the path for the bounding box.

```

152   bbox code/.code={
153     \pgfresetboundingbox
154     \node[/drawmatrix/every bbox, /drawmatrix/bbox,
155       minimum height=\dm@bboxheight,
156       minimum width=\dm@bboxwidth] {};
157   },

```

Whether needed or not, declare all matrix corners.

```

158   coordinate code/.code={
159     \path (drawmatrix bbox.north west)
160       ++(\dm@offsetwidth, -\dm@offsetheight)

```

```

161         ++(.5\pgflinewidth, -.5\pgflinewidth)
162         coordinate (drawmatrix north west)
163         ++(\dm@width, 0)
164         +(-\dm@minsize + \dm@upperbandwidth, 0)
165         coordinate (drawmatrix north)
166         +(0, -\dm@minsize + \dm@upperbandwidth)
167         coordinate (drawmatrix east)
168         ++(0, -\dm@height)
169         coordinate (drawmatrix south east)
170         ++(-\dm@width, 0)
171         +(\dm@minsize - \dm@lowerbandwidth, 0)
172         coordinate (drawmatrix south)
173         +(0, \dm@minsize - \dm@lowerbandwidth)
174         coordinate (drawmatrix west);
175     },

```

Add an invisible node the size of the matrix.

```

176     node code/.code={
177         \node[/drawmatrix/every node,
178             minimum height=\dm@height,
179             minimum width=\dm@width] {};
180     },

```

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

```

181     draw code/.code={
182         \filldraw[/drawmatrix/every drawmatrix, /drawmatrix/drawmatrix]
183             (drawmatrix north west)
184             \ifx\dm@upperbandwidth\dm@zero
185                 \ifx\dm@width\dm@minsize\else -- (drawmatrix north) \fi
186                 \ifx\dm@height\dm@minsize\else -- (drawmatrix east) \fi
187             \else
188                 -- (drawmatrix north)
189                 \ifx\dm@upperbandwidth\dm@minsize\else
190                     -- (drawmatrix east)
191                 \fi
192             \fi
193             -- (drawmatrix south east)
194             \ifx\dm@lowerbandwidth\dm@zero
195                 \ifx\dm@width\dm@minsize\else -- (drawmatrix south) \fi
196                 \ifx\dm@height\dm@minsize\else -- (drawmatrix west) \fi
197             \else
198                 -- (drawmatrix south)
199                 \ifx\dm@lowerbandwidth\dm@minsize\else
200                     -- (drawmatrix west)
201                 \fi
202             \fi
203             -- cycle;
204     },

```

The label.

```

205     label code/.code={
206         \node[/drawmatrix/every label, /drawmatrix/label]
207             {\dm@labeltext};
208         \expandafter\ifx\dm@labelbase\pgfkeysnovalue\else
209             \node[/drawmatrix/every label, /drawmatrix/label,
210                 /drawmatrix/label outer] {\dm@labeltextouter};
211         \fi
212     }
213 }

```

5.6 User Macros

`\drawmatrixset`^{→P.8} as a simple shortcut like `\tikzset`.

```

214 \newcommand\drawmatrixset[1]{\pgfkeys{/drawmatrix}{#1}}

```

Here we go, the main thing: `\drawmatrix`^{→P.2}. First, apply the options and extract the sizes from the PGF keys.

```

215 \newcommand\drawmatrix[2] []{{%
216     \drawmatrixset{
217         label text={#2},
218         #1,
219         label text/.get=\dm@labeltext,
220         height/.get=\dm@height,
221         width/.get=\dm@width,
222         lower bandwidth/.get=\dm@lowerbandwidth,
223         upper bandwidth/.get=\dm@upperbandwidth,
224         offset height/.get=\dm@offsetheight,
225         offset width/.get=\dm@offsetwidth,
226         bbox height/.get=\dm@bboxheight,
227         bbox width/.get=\dm@bboxwidth,
228         label base/.get=\dm@labelbase,
229         pre code,
230         setup label code,
231         externalization code
232     }%
233     \begin{tikzpicture}[/drawmatrix/every picture, /drawmatrix/picture]
234         \drawmatrixset{
235             setup sizes code,
236             post setup code,
237             bbox code,
238             pre coordinate code,
239             coordinate code,
240             post coordinate code,
241             node code,
242             pre draw code,
243             draw code,
244             post draw code,
245             pre label code,
246             label code,
247             post label code

```

```
248     }
249   \end{tikzpicture}%
250   \drawmatrixset{post code}%
251 }}}
```


Index

- banded key, 4
- bandwidth key, 4
- baseline key, 8
- bbox key, 6
- bbox height key, 6
- bbox size key, 6
- bbox style key, 6
- bbox width key, 6

- diag key, 5
- \drawmatrix, 2
- \drawmatrixset, 8

- every bbox key, 8
- every drawmatrix key, 8
- every label key, 8
- every picture key, 8
- exponent key, 5
- externalize key, 9

- height key, 3

- Keys
 - /drawmatrix/
 - banded, 4
 - bandwidth, 4
 - baseline, 8
 - bbox, 6
 - bbox height, 6
 - bbox size, 6
 - bbox style, 6
 - bbox width, 6
 - diag, 5
 - every bbox, 8
 - every drawmatrix, 8
 - every label, 8
 - every picture, 8
 - exponent, 5
 - externalize, 9
 - height, 3
 - label anchor, 7
 - label base, 5
 - label base anchor, 5
 - label pos, 8
 - label text, 3
 - lower, 3
 - lower banded, 4
 - lower bandwidth, 4
 - lower offset, 7
 - lower offset height, 6
 - lower offset width, 6
 - lower scale, 7
 - lower size, 3
 - lower upper, 3
 - lower upper banded, 4
 - lower upper bandwidth, 4
 - lower width, 3
 - lower x, 7
 - lower y, 7
 - label anchor key, 7
 - label base key, 5
 - label base anchor key, 5
 - label pos key, 8
 - label text key, 3
 - lower key, 3
 - lower banded key, 4
 - lower bandwidth key, 4

 - offset key, 7
 - offset height key, 6
 - offset width key, 6

 - scale key, 7
 - size key, 3

 - upper key, 3
 - upper banded key, 4
 - upper bandwidth key, 4

 - width key, 3

 - x key, 7

 - y key, 7