The graphbox Package

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

nbeisert@itp.phys.ethz.ch

17 January 2018, v1.1

Abstract

graphbox is an extension for the LaTeX 2_ε package graphicx to facilitate the placement of graphics relative to the current position using additional optional arguments `[...]` of `\includegraphics`. For example, changing the vertical alignment is convenient for using graphics as elements of (mathematical) formulae. Options for shifting, smashing and hiding the graphics are mainly intended for designing presentations using, e.g., the beamer framework.

Contents

1 Introduction

Changing the placement of graphics using `\includegraphics` of the graphicx package is tedious. The command allows to scale and even rotate a figure by the optional arguments `[...]` of `\includegraphics`, namely `scale` (`xscale`, `yscale`) and `rotate`. However, it lacks options to change the vertical alignment, shift the graphics or modify the containing TeX box. In order to position the graphics according to one's needs, one would normally use the standard boxing tools of LaTeX such as `\makebox`, `\parbox`, `\raisebox` or the `minipage` and `picture` environments. While practically anything can be achieved by a combination of these tools, it has several disadvantages:

- use of complex commands for standard situations;
- several boxing commands need to be combined to achieve the desired effect; leads to several levels of nesting around the actual `\includegraphics` command;
- readability of the LaTeX construction is low (what is going on?);
- adjusting the placement is tedious;
- the higher-level boxing commands of LaTeX require to specify the box width; however, the actual dimensions of the graphics are typically not known (or they change as the graphics is modified); furthermore the same number needs to be used in several places;
- guessing the width of the required boxes may lead to extra space around the graphics or warning messages about an overfull box;
- ...

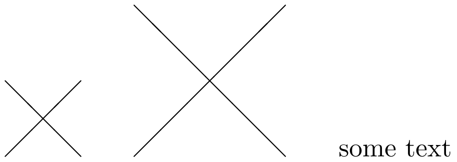
Therefore it would be desirable to specify the placement along with other options in the `[...]` argument of `\includegraphics`. This may be helpful in the following situations:

- presentations where several figure need to be placed across the frames without disturbing the flow of the text; similarly, in the composition of posters;
- when graphics are used as elements of (mathematical) formulae one might prefer a central vertical alignment over the default baseline alignment of `\includegraphics`.

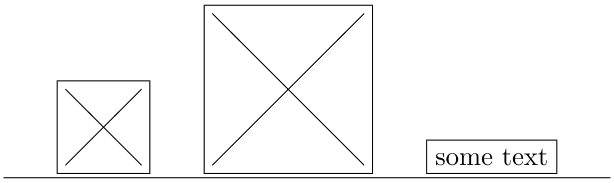
The following discusses these situations in more detail.

1.1 Vertical Alignment

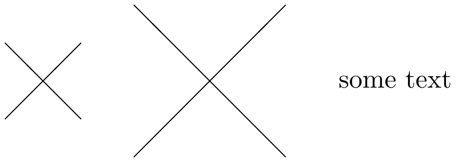
Suppose you want to show graphics files[1] of different height next to each other (potentially accompanied by some text). A plain inclusion via `\includegraphics` will produce the output:



This is because `\includegraphics` puts the graphics into a box whose baseline is the lower bound of the box:



Instead, a central alignment of the graphics is desirable in many situations:



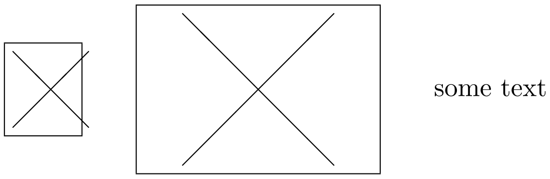
---

[1] a diagonal cross will serve as the sample figure in this documentation.

This can be achieved by code such as:

```
\parbox{1cm}{\includegraphics[width=1cm]{...}}
\qquad
\parbox{2cm}{\includegraphics[width=2cm]{...}}
\qquad
\mbox{some text}
```

However, if the width of the graphics is not as evident (e.g. when no width is specified or the graphics is scaled), the resulting output may easily look like:[2]



The present extension `graphbox` adds the option `align` to `\includegraphics` to simplify the declaration:

```
\includegraphics[align=c,width=1cm]{...}
\qquad
\includegraphics[align=c,width=2cm]{...}
\qquad
\mbox{some text}
```

1.2 Graphics Placement

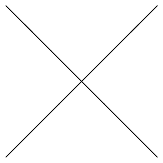
A related issue is the fine-tuning of graphics placement: In LaTeX this can be achieved by the commands `\raisebox` (vertically) and `\hspace` (horizontally). Furthermore, one might want to place the graphics into some free space relative to the present position. This is commonly needed in designing presentations or posters, where the flow of the text and the graphics is often fine-tuned. For example, one may ignore the vertical height of the graphics using `\smash`. Or one might place the graphics at the present horizontal position without allocating space by `\makebox[0pt]`, `\llap` or `\rlap`.

Each of these elementary operations can be handled well, but their combination can become very involved:

```
\smash{\raisebox{.1cm}{\makebox[0pt][l]{\hspace{1cm}{\includegraphics{...}}}}}
```

For example, this may be used to produce the output:

	A	B	C
D	1	2	3
E	4	5	6
F	7	8	9 □



Here, the graphics was put next to a table. The small box represents the anchor of the graphics. It occupies no space which allows to centre the table irrespectively of the size of the graphics. Unfortunately, the command is not easily readable. The `graphbox` extension abbreviates the declaration and make it much more accessible:

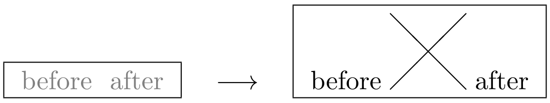
```
\includegraphics[smash=tr,hshift=1cm,vshift=.1cm]{...}
```

---

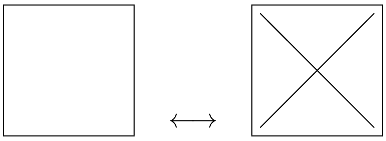
[2] Boxes were added to display the size of the bounding boxes.

1.3 Hiding Graphics

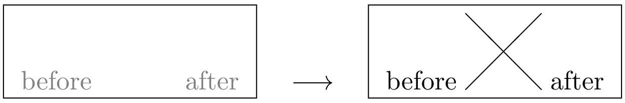
Sometimes one might wish to hide a graphics while reserving the space it would occupy.[3] For example, this feature is needed in presentations where the frames are successively uncovered. In this beamer class the uncovering of frames is conveniently achieved by the command `\pause`. Unfortunately, the effect of `\pause` on graphics is that they are either displayed or taken out of the text flow. This has the effect that the formatting can change drastically when graphics elements are uncovered:



The graphbox extension adds the option `hide` to `\includegraphics` to hide the corresponding graphics while reserving the space it would otherwise occupy:



Moreover, in conjunction with the beamer class, the definition of the `\includegraphics` command is altered such that it uses the same amount of space in all visibility modes (specified through the `\pause` mechanism and the `<...>` extension). Visibility is handled by automatically setting the `hide` option:



2 Usage

To use graphbox simply add the command

```
\usepackage{graphbox}
```

to the preamble of your LaTeX document. If not yet present, the package graphicx will automatically be loaded.

2.1 Extensions

The package mainly extends the functionality of the `\includegraphics` command of the graphicx package by allowing several additional optional arguments `[...]` as follows:

- `hide`[=*bool*]

  hides the graphics (unless *bool*=`false`) but reserves the space it would have occupied.

- `align`[=*valign*]

  adjusts the vertical alignment of the graphics where *valign* is one of the following:

  `b`: baseline aligned to bottom of graphics;
  default behaviour when `align` is not specified;

---

[3] In the graphicx package a similar effect achieved by the option `draft`. However, here the graphics are replaced by a box containing the file name.

  - `c`: centre of current line aligned to centre of graphics;[4]
    default behaviour when no parameter is given;
  - `t`: top of current line aligned to top of graphics;[5]
  - `l`: baseline aligned to bottom of graphics; same as `b`;
  - `m`: baseline aligned to centre of graphics;
  - `u`: baseline aligned to top of graphics.

- `vsmash`[=*vpos*]

  reduces the height of the bounding box to zero (equivalent to `\smash`); places the graphics according to *vpos* which can take the following values:

  - `b`: positions top of graphics below top of current line;
  - `c`: positions centre of graphics at centre of current line;
    default behaviour when no parameter is given;
  - `t`: positions bottom of graphics at baseline;
  - `l`: positions top of graphics at baseline;
  - `m`: positions centre of graphics at baseline;
  - `u`: positions bottom of graphics at baseline; same as `t`;
  - `n`: no vertical smashing.

  Note that the parameter *vpos* has the opposite effect of the parameter *valign* of `align`.

- `hsmash`[=*hpos*]

  reduces the width of the bounding box to zero (equivalent to `\makebox[0pt]`); places the graphics according to *hpos* which can take the following values

  - `r`: positions graphics right of current position;
  - `c`: positions centre of the graphics at current position; default behaviour when no parameter is given;
  - `l`: positions graphics left of current position;
  - `n`: no horizontal smashing; default behaviour when `hsmash` is not specified.

- `smash`[=*vpos hpos*]

  combination of `vsmash=`*vpos* and `hsmash=`*hpos*; default behaviour is `cc`.

- `vshift=`*len*

  shifts the graphics up by *len*.

- `hshift=`*len*

  shifts the graphics to the right by *len*; has no effect unless `hsmash` is activated.

- `tmargin=`*len*

  adds a top margin of height *len*; has no effect when `vsmash` is activated.

- `bmargin=`*len*

  adds a bottom margin of height *len* below the baseline; has no effect when `vsmash` is activated.

---

[4] The centre of the line appears to be (18/31)ex above the baseline which is at the centre of a capital letter for the default font.

[5] The top of the line is defined as (36/31)ex above the baseline which is the height of a capital letter for the default font.

- `vmargin=`*len*

  adds a bottom and top margin of height *len*; has no effect when `vsmash` is activated.

- `lmargin=`*len*

  adds a left margin of width *len*; has no effect when `hsmash` is activated.

- `rmargin=`*len*

  adds a right margin of width *len*; has no effect when `hsmash` is activated.

- `hmargin=`*len*

  adds a left and right margin of width *len*; has no effect when `hsmash` is activated.

- `margin=`*len*

  adds an overall margin of length *len* around the graphics; has no effect when `smash` is activated.

In specifying lengths in the above arguments, there are two additional dimensions `\gwidth` and `\gheight` representing the actual width and height of the included graphics. For example, one can add a horizontal margin of 10% of the size of the graphics by specifying `hmargin=0.1\gwidth`.

2.2 Package Options

The package provides one option:

- `nobeamer`

  do not override the overlay processing of the `beamer` class for `\includegraphics`; as usual, graphics will occupy no space when covered.

3 Information

3.1 Copyright

The Current Maintainer of this work is Niklas Beisert.

This work consists of the files `README.txt`, `graphbox.ins` and `graphbox.dtx` as well as the derived files `graphbox.sty`, `gboxsamp.tex` with `gboxsamp.mps` and `graphbox.pdf`.

3.2 Files and Installation

The package consists of the files

`README.txt`	readme file
`graphbox.ins`	installation file
`graphbox.dtx`	source file
`graphbox.sty`	package file
`gboxsamp.tex`	sample file
`gboxsamp.mps`	sample figure
`graphbox.pdf`	manual

The distribution consists of the files `README.txt`, `graphbox.ins` and `graphbox.dtx`.

- Run (pdf)LaTeX on `graphbox.dtx` to compile the manual `graphbox.pdf` (this file).
- Run LaTeX on `graphbox.ins` to create the package `graphbox.sty` and the sample consisting of `gboxsamp.tex` and `gboxsamp.mps`. Copy the file `graphbox.sty` to an appropriate directory of your LaTeX distribution, e.g. *texmf-root*`/tex/latex/graphbox`.

3.3 Interaction with CTAN Packages

The graphbox package extends the package graphicx. It also changes some functionality of the class beamer if present:

- Compatibility with the graphicx package has been tested with v1.0f (1999/02/16) and v1.0g (2014/04/25).
- The changes of functionality are described in section 1.3. Compatibility with the beamer class has been tested with v3.33 (2013/12/25).

3.4 Revision History

v1.1: 2018/01/17

- hooked deeper into `\includegraphics` chain
- manual rearranged
- minor internal changes

v1.0: 2014/08/31

- added further placement options
- manual and installation package added
- first version published on CTAN

v0.8: 2014/08/25

- basic functionality

A Sample File

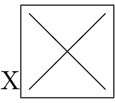
In this section we provide a LaTeX example how to use some of the graphbox extensions to graphicx. Preamble:

```
\documentclass[12pt]{article}

\usepackage[margin=2cm]{geometry}
\usepackage{graphicx}
\usepackage{graphbox}
```

The macro `\sample{`*options*`}` prints a block demonstrating the flow of text testing the new `\includegraphics` arguments *options*:

options: *options*

XXXXXX XXXXXX
XXXXXX XXXXXX
XXXXXX  XXXXXX
XXXXXX XXXXXX
XXXXXX XXXXXX

The box indicates the resulting bounding box of the `\includegraphics` command. It is defined as follows:

```
\newcommand{\sample}[1]{
\textbf{options:} \texttt{#1}\par\vspace{1ex}
\parbox{0.6\textwidth}{%
XXXXXX\phantom{\fbox{}}XXXXXX\\%
XXXXXX\phantom{\fbox{}}XXXXXX\\%
XXXXXX\fbox{\includegraphics[#1]{gboxsamp.mps}}XXXXXX\\%
XXXXXX\phantom{\fbox{}}XXXXXX\\%
XXXXXX\phantom{\fbox{}}XXXXXX%
}\vspace{1cm}\par}
```

In the document body we test the various new options for `\includegraphics`:

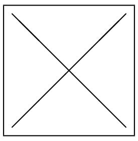
```
\begin{document}
\sample{}
\sample{hide}
\sample{hide=true}
\sample{hide=false}
\newpage
\sample{align=b}
\sample{align=c}
\sample{align=t}
\newpage
\sample{align=l}
\sample{align=m}
\sample{align=u}
\newpage
\sample{vshift=10pt}
\sample{vshift=-10pt}
\newpage
\sample{lmargin=10pt}
\sample{rmargin=10pt}
\sample{tmargin=10pt}
\sample{bmargin=10pt}
\newpage
\sample{lmargin=-10pt}
\sample{rmargin=-10pt}
\sample{tmargin=-10pt}
\sample{bmargin=-10pt}
\newpage
\sample{vsmash=b}
\sample{vsmash=c}
\sample{vsmash=t}
\newpage
\sample{vsmash=l}
\sample{vsmash=m}
\sample{vsmash=u}
\newpage
```

```
\sample{hsmash=l}
\sample{hsmash=c}
\sample{hsmash=r}
\newpage
\sample{smash=tl}
\sample{smash=cc}
\sample{smash=br}
\newpage
\sample{smash,vshift=10pt}
\sample{smash,hshift=10pt}
\sample{smash,hshift=-10pt,vshift=-10pt}
\end{document}
```

Sample Graphics. The following is a sample Metapost graphics file (`gboxsamp.mps`) for use in the sample LaTeX file:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -21 -21 21 21
%%EndComments
%%Page: 1 1
 0 0 0 setrgbcolor 0 1.5 dtransform truncate idtransform setlinewidth pop []
 0 setdash 1 setlinecap 1 setlinejoin 10 setmiterlimit
newpath -20 20 moveto 20 -20 lineto stroke
newpath -20 -20 moveto 20 20 lineto stroke
%%EOF
```

It contains a black cross:



The frame indicates the size of the graphics but it does not belong to it.

B Implementation

In this section we describe the package `graphbox.sty`.

Package Options. By default, the package overrides the overlay meachanism for includegraphics of the class beamer. The global option `nobeamer` disables the interaction.

```
\newif\ifGin@box@beamer\Gin@box@beamertrue
\DeclareOption{nobeamer}{\Gin@box@beamerfalse}
\ProcessOptions
```

Required Packages. The package loads graphicx if not yet present:

```
\RequirePackage{graphicx}
```

Internal Definitions. The package defines a box and several dimension registers:

```
\newsavebox{\Gin@box@box}
\newlength{\Gin@box@width}
\newlength{\Gin@box@height}
\newlength{\Gin@box@raise}
```

```
\newlength{\Gin@box@hspace}
\newlength{\Gin@box@bwidth}
\newlength{\Gin@box@bheight}
```

The following macros store the parameters for the present graphics:

```
\def\Gin@box@align{b}
\def\Gin@box@hsmash{n}
\def\Gin@box@hshift{0pt}
\def\Gin@box@vshift{0pt}
\def\Gin@box@lmargin{0pt}
\def\Gin@box@rmargin{0pt}
\def\Gin@box@tmargin{0pt}
\def\Gin@box@bmargin{0pt}
\newif\ifGin@box@hide\Gin@box@hidefalse
\newif\ifGin@box@vsmash\Gin@box@vsmashfalse
```

Additional Arguments. The optional arguments of `\includegraphics` are parsed by keyval which calls the appropriate handlers. We add handlers to introduce additional optional arguments:

```
\define@key{Gin}{hide}[true]{\lowercase{\Gin@boolkey{#1}}{box@hide}}
\define@key{Gin}{align}[c]{\def\Gin@box@align{#1}}
\define@key{Gin}{hsmash}[c]{\edef\Gin@box@hsmash{#1}}
\define@key{Gin}{vsmash}[c]{\if#1n\Gin@box@vsmashfalse\else%
                            \Gin@box@vsmashtrue\edef\Gin@box@align{%
                            \if#1bt\else\if#1tb\else%
                            \if#1lu\else\if#1ul\else#1\fi\fi\fi\fi\fi}}
\define@key{Gin}{smash}[cc]{\expandafter\KV@Gin@vsmash\@firstoftwo#1%
                          \expandafter\KV@Gin@hsmash\@secondoftwo#1}
\define@key{Gin}{hshift}{\def\Gin@box@hshift{#1}}
\define@key{Gin}{vshift}{\def\Gin@box@vshift{#1}}
\define@key{Gin}{lmargin}{\def\Gin@box@lmargin{#1}}
\define@key{Gin}{rmargin}{\def\Gin@box@rmargin{#1}}
\define@key{Gin}{tmargin}{\def\Gin@box@tmargin{#1}}
\define@key{Gin}{bmargin}{\def\Gin@box@bmargin{#1}}
\define@key{Gin}{hmargin}{\def\Gin@box@lmargin{#1}\def\Gin@box@rmargin{#1}}
\define@key{Gin}{vmargin}{\def\Gin@box@tmargin{#1}\def\Gin@box@bmargin{#1}}
\define@key{Gin}{margin}{\def\Gin@box@lmargin{#1}\def\Gin@box@rmargin{#1}%
                        \def\Gin@box@tmargin{#1}\def\Gin@box@bmargin{#1}}
```

The options are stored in internal registers for later processing. The arguments of `hsmash` and `vsmash` work in the opposite direction for conventional alignment of LaTeX boxes. Therefore the parameters `t` and `b` as well as `r` and `l` are interchanged.

Argument Processing. Next, we overwrite the function `\Gin@setfile` called by `\includegraphics` after the optional parameters have been processed by `\setkeys{Gin}{...}`:

```
\let\old@box@Gin@setfile\Gin@setfile
\def\Gin@setfile#1#2#3{%
```

We first save the graphics into the box `\Gin@box@box` and compute its width and height:

```
  \sbox{\Gin@box@box}{\old@box@Gin@setfile{#1}{#2}{#3}}%
  \settowidth{\Gin@box@width}{\usebox{\Gin@box@box}}%
  \settoheight{\Gin@box@height}{\usebox{\Gin@box@box}}%
```

Temporarily define the macros `\gwidth` and `\gheight` to represent the width and height of the graphics, respectively:

```
\def\gwidth{\Gin@box@width}%
\def\gheight{\Gin@box@height}%
```

Ignore top and bottom margin if vsmash is active:

```
\ifGin@box@vsmash%
 \def\Gin@box@tmargin{0pt}\def\Gin@box@bmargin{0pt}\fi%
```

Compute the vertical box dimensions:

```
\setlength{\Gin@box@raise}{\Gin@box@vshift}%
\setlength{\Gin@box@bheight}{\Gin@box@height}%
\addtolength{\Gin@box@bheight}{\Gin@box@tmargin}%
\addtolength{\Gin@box@bheight}{\Gin@box@bmargin}%
\if\Gin@box@align t%
 \addtolength{\Gin@box@raise}{-\Gin@box@height}%
 \addtolength{\Gin@box@raise}{1.161290323ex}\fi%
\if\Gin@box@align u%
 \addtolength{\Gin@box@raise}{-\Gin@box@height}\fi%
\if\Gin@box@align c%
 \addtolength{\Gin@box@raise}{-0.5\Gin@box@height}%
 \addtolength{\Gin@box@raise}{0.580645161ex}\fi%
\if\Gin@box@align m%
 \addtolength{\Gin@box@raise}{-0.5\Gin@box@height}\fi%
```

Compute the horizontal box dimensions:

```
\setlength{\Gin@box@hspace}{0pt}%
\if\Gin@box@hsmash n%
 \setlength{\Gin@box@bwidth}{\Gin@box@width}%
 \addtolength{\Gin@box@bwidth}{\Gin@box@lmargin}%
 \addtolength{\Gin@box@bwidth}{\Gin@box@rmargin}%
 \setlength{\Gin@box@hspace}{\Gin@box@lmargin}%
\else%
 \setlength{\Gin@box@bwidth}{0pt}%
 \setlength{\Gin@box@hspace}{\Gin@box@hshift}%
 \if\Gin@box@hsmash l%
  \addtolength{\Gin@box@hspace}{-\Gin@box@width}\fi%
 \if\Gin@box@hsmash c%
  \addtolength{\Gin@box@hspace}{-0.5\Gin@box@width}\fi%
\fi%
```

Define the vertical box:

```
\ifGin@box@vsmash\expandafter\smash\fi{%
 \raisebox{\Gin@box@raise}{%
  \parbox[b]{\Gin@box@bwidth}{%
   \rule[-\Gin@box@bmargin]{0pt}{\Gin@box@bheight}%
   \smash{%
```

Define the horizontal box:

```
    \makebox[0pt][l]{%
     \hspace*{\Gin@box@hspace}%
```

Print the graphics unless hidden:

```
     \ifGin@box@hide\else\usebox{\Gin@box@box}\fi%
    }%
    }%
   }%
  }%
 }%
}
```

Interaction with class beamer. For the class beamer we want to override the overlay mechanism to always reserve the space the graphics would occupy. Here we have to overwrite the command `\includegraphics`.

```
\@ifclassloaded{beamer}{\ifGin@box@beamer
```

Save the original definition of `\includegraphics` and begin the new one:

```
\let\old@box@includegraphics\includegraphics
\newcommand<>{\fibox@includegraphics}[2][]{\Gin@box@hidefalse%
```

Check whether overlay parameters `<...>` are specified. If none, use the counter of the `\pause` mechanism to determine visibility. Call `\alt<...>` to set the `hide` argument according to visibility:

```
 \beamer@ifempty{#3}%
  {\alt<\c@beamerpauses->{}{\Gin@box@hidetrue}}%
  {\alt#3{}{\Gin@box@hidetrue}}%
```

Pass on to old definition:

```
 \old@box@includegraphics[#1]{#2}}
```

beamer overwrites `\includegraphics` at the beginning of the document body. We have to overwrite it again discarding the changes introduced by beamer:

```
\AtBeginDocument{\let\includegraphics=\fibox@includegraphics}
\fi}{}
```