The empheq package*
Emphasizing equations in LaTeX 2ε

Morten Høgholm, Lars Madsen, and the LaTeX Project

2024/08/09

Abstract

The empheq package can best be described as a visual markup extension to amsmath. In short it offers a) a multi line equivalent of \boxed from amsmath and b) a way to produce arbitrary delimiters that span entire math displays.

Contents

I Basic user's guide **3**

1 Basic use of the package **3**
1.1 Using the empheq environment . . . 3
1.2 Markup instructions . . . 4
1.3 What won't work in the empheq environment . . . 7
1.4 Special delimiters . . . 7

2 Taking things a little further **8**
2.1 Loading the package . . . 8
2.1.1 Compatibility with the old version . . . 8
2.2 Package options . . . 9
2.2.1 Using multline . . . 9
2.2.2 The overload option . . . 11
2.3 A note on boxed displays . . . 13

3 Support for other packages **14**
3.1 Support for fancybox . . . 14
3.2 Support for ntheorem . . . 15

II Advanced user's guide **17**

4 Delimiters revisited **17**
4.1 Creating your own delimiters . . . 17
4.2 Fine-tuning of delimiters . . . 17
4.3 Scaling material yourself . . . 18

*This file has version number v2.17, last revised 2024/08/09.

5 A few short notes **19**
5.1 About `\eqref` 19
5.2 About changes to `\baselineskip` 19

6 Creating something new **19**
6.1 New `empheq`-like environments 20
6.2 Creating fancy boxes 21

7 Contact information **22**

Part I
Basic user's guide

There can be little doubt that the de facto standard for mathematical typesetting in LaTeX is the amsmath package. For the creation of empheq, a visual markup package for use in math, it made perfect sense to have amsmath as the backbone.

The main idea of empheq is to maintain the familiar syntax of the amsmath environments while still providing an easy way of specifying markup instructions. This manual is plastered with examples showing just how.

1 Basic use of the package

So what is it empheq does? Well, it allows you to produce displays like this:

$$X = Y \Rightarrow \left\{ \begin{array}{lll} A_1 = b_1 & c_1 = d_1 & e_1 = f_1 \\ A_2 = b_2 & c_2 = d_2 & e_2 = f_2 \\ A_3 = b_3 & c_3 = d_3 & e_3 = f_3 \end{array} \right. \quad \text{_A}^{\text{silly}}_{\text{tag_}} \tag{1a}$$

(1b)

In short empheq enables the user to put things on every side of the display without said user having to worry about what happens to the equation numbers. For example you can now have a display containing multiple lines and still get the effect of the \boxed command from amsmath.

1.1 Using the empheq environment

The package defines a single environment empheq and the usage is kind of straightforward:

```
\begin{empheq}[⟨markup instructions⟩]{⟨AMS env_name⟩}
   ⟨contents of AMS environment⟩
\end{empheq}
```

A first minimal example file would then be something like

```
\documentclass{minimal}
\usepackage{empheq}
\begin{document}
\begin{empheq}{align*}
  a&=b \tag{*}\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
\end{document}
```

This gives the following display:

$$\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned} \tag{*}$$

Environment	Usage	Environment	Usage
`equation`	`{equation}`	`equation*`	`{equation*}`
`align`	`{align}`	`align*`	`{align*}`
`gather`	`{gather}`	`gather*`	`{gather*}`
`flalign`	`{flalign}`	`flalign*`	`{flalign*}`
`alignat`	`{alignat=`⟨*cols*⟩`}`	`alignat*`	`{alignat*=`⟨*cols*⟩`}`
`multline`	`{multline}`	`multline*`	`{multline*}`

Table 1: The supported amsmath environments

Maybe not the most impressive example, but as you can see the contents of the environment are exactly the same as for the regular `align*` environment from amsmath. The rest of the $\mathcal{AMS}$ environments are chosen the same way by typing the name as the mandatory argument of `empheq` with the exception of the `alignat` environment. For this you have to specify the number of columns as shown below.

```
\begin{empheq}{alignat=2}
  a &= b &\quad c &= d \\
  \text{this} &= \text{that} &\quad \mathit{fish}&\neq fish
\end{empheq}
```

$$a = b \qquad c = d \tag{2}$$
$$\text{this} = \text{that} \quad \mathit{fish} \neq \mathit{fish} \tag{3}$$

To choose the starred version of `alignat`, simply type `{alignat*=2}` instead in the above example.

The supported $\mathcal{AMS}$ environments are listed in Table 1. Not supported is the standard LaTeX `eqnarray` environment as it is fundamentally deficient.[1]

1.2 Markup instructions

The optional argument of the `empheq` environment will take markup instructions consisting of a ⟨*key*⟩ = ⟨*value*⟩ list of assignments. There are currently five such keys (a sixth is added for naming consistency). They're shown in Table 2 on the next page.

left

The key left is for material put on the left side of the display. The material is typeset in math mode and centered vertically.

```
\begin{empheq}[left=L\Rightarrow]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

[1] See for instance https://texfaq.org/FAQ-eqnarray.

Key	Usage	Additional Info
box	`box=`⟨*box command*⟩	
innerbox	`innerbox=`⟨*box command*⟩	
left	`left=`⟨*math material*⟩	Use `\text{...}` if you need text material.
right	`right=`⟨*math material*⟩	Use `\text{...}` if you need text material.
outerbox	`outerbox=`⟨*box command*⟩	Alias for box.
marginbox	`marginbox=`⟨*box command*⟩	Contents must be horizontally centered; can only be used in `[fleqn]` mode.

Table 2: The six keys for the optional argument of the `empheq` environment

$$a = b \tag{4}$$
$$L \Rightarrow E = mc^2 + \int_a^a x\,dx \tag{5}$$

right

As there is a left key it hopefully comes as no surprise that there is a companion right key for typesetting material on the right side of the display.

```
\begin{empheq}[right=\Leftarrow R]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$a = b \tag{6}$$
$$E = mc^2 + \int_a^a x\,dx \Leftarrow R \tag{7}$$

box
outerbox

The key box specifies the kind of box you would like to put around the display. It can be any kind of box, as long as the contents of the box are situated on the baseline like in a `\fbox`.

```
\begin{empheq}[box=\fbox]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$\boxed{\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned}} \tag{8, 9}$$

The key outerbox is an alias for box and is added for naming consistency with the key described below.

innerbox

There is also an innerbox key. It is not very interesting unless you use one of the other keys.

```
\begin{empheq}[innerbox=\fbox,
               left=L\Rightarrow]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$L \Rightarrow \boxed{\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned}} \tag{10}$$

(11)

Feature request by Uwe Siart 2003/12/08

marginbox

The last key is the marginbox key. If you typeset your math in `[fleqn]` mode[2] you may want the math display and not the outer box to align at the left margin (or rather: the indentation). If you make sure the contents of the outer box are centered horizontally inside the box, marginbox will align it properly. You shouldn't set both box and marginbox at the same time, as this is surely not what you want and the package will silently use the last one in the list.

> **Warning**: keyval treats commas and equal signs as separator and assignment signs which in turn means that if you want to typeset either of them, they *must*—and believe me, it's really important—be enclosed in braces. In short:
>
> **Good** `[left={(A,B)=(1,0)}]`
>
> **Bad** `[left=(A,B)=(1,0)]`

Feature request by Uwe Siart 2004/01/13

`\empheqset{`⟨*markup instructions*⟩`}`

You can also set the keys globally[3] with the command `\empheqset`. This means that

```
\empheqset{marginbox=\psframebox}
```

will force a `box=\psframebox` in all occurrences of the `empheq` environment, but an explicitly given box or marginbox will override this setting. You can only use the keys listed in this section as arguments to `\empheqset`; not the $\mathcal{AMS}$ environments.

[2] Sorry, but I can't show you because this manual features centered math displays.

[3] Or rather: outside the scope of the `empheq` environment. The settings will still obey scoping rules.

1.3 What won't work in the `empheq` environment

`\intertext{`⟨*text*⟩`}`
`\displaybreak[`⟨*num*⟩`]`

Now we've seen some of the things that work in the `empheq` environment but we also have to take note of what won't work. As this package provides a way to box multi line math displays it can come as no surprise that using either `\intertext`[4] or `\displaybreak` inside the `empheq` environment makes no sense. Should you however happen to try them anyway, you'll experience that `\intertext` issues an error message and `\displaybreak` issues a warning. The reason only `\intertext` gives an error message is that you'll get output very different from what you expect, and that is not the case (so much at least) with `\displaybreak`. But of course, you'll never see either message because you read the manual!

Very long lines

The standard $\mathcal{AMS}$ environments will move the equation number if a line gets too close to it. Due to implementation, this feature is *not* carried correctly to the `empheq` environments, so keep equation lines short then applying `empheq` to them.

1.4 Special delimiters

As you've seen a few pages back, it's possible to add material on both sides of the math display. When doing so you often need a delimiter that scales to fit the entire display, so it comes as no surprise that empheq provides such delimiters.

`\empheql`⟨*delim_name*⟩ `\empheqbigl`⟨*delim_name*⟩
`\empheqr`⟨*delim_name*⟩ `\empheqbigr`⟨*delim_name*⟩

For example one might need a large brace like the one in the `cases` environment:

```
\begin{empheq}[left=\empheqlbrace, right=\empheqrbrace]{align}
  E&=mc^2 \\
  Y&= \sum_{n=1}^\infty \frac{1}{n^2}
\end{empheq}
```

$$\left.\begin{aligned} E &= mc^2 && (12)\\ Y &= \sum_{n=1}^{\infty} \frac{1}{n^2} && (13) \end{aligned}\right\}$$

The naming scheme is `\empheq`⟨*delimiter name*⟩ and `\empheqbig`⟨*delimiter name*⟩. Thus `\empheqrbrace` produces an auto-scaling right brace with same size as the math display while `\empheqbigrbrace` produces an even bigger version that spans the inner box plus the math display inside it. For a complete list of supported delimiters see Table 3 on the following page.

[4] And thus also `\shortintertext` (mathtools).

Delimiter			
Original	Normal	Bigger	Symbol
\lbrace	\empheqlbrace	\empheqbiglbrace	{
\rbrace	\empheqrbrace	\empheqbigrbrace	}
\lbrack	\empheqlbrack	\empheqbiglbrack	[
\rbrack	\empheqrbrack	\empheqbigrbrack	]
\langle	\empheqlangle	\empheqbiglangle	⟨
\rangle	\empheqrangle	\empheqbigrangle	⟩
\lparen	\empheqlparen	\empheqbiglparen	(
\rparen	\empheqrparen	\empheqbigrparen	)
\lvert	\empheqlvert	\empheqbiglvert	|
\rvert	\empheqrvert	\empheqbigrvert	|
\lVert	\empheqlVert	\empheqbiglVert	‖
\rVert	\empheqrVert	\empheqbigrVert	‖
\lfloor	\empheqlfloor	\empheqbiglfloor	⌊
\rfloor	\empheqrfloor	\empheqbigrfloor	⌋
\lceil	\empheqlceil	\empheqbiglceil	⌈
\rceil	\empheqrceil	\empheqbigrceil	⌉

Table 3: The supported auto-scaling delimiters in empheq

2 Taking things a little further

So far we have covered the basic functions of empheq: the markup instructions and the predefined auto-scaling delimiters. There is more to empheq however, and we'll cover that in this section.

2.1 Loading the package

The package has two main requirements: One is the mathtools[5] package which provides empheq with a lot of necessary tools for doing its thing. The other one is amsmath of course, and to make the loading procedure as easy as possible, you can simply substitute

```
\usepackage[leqno,fleqn,intlimits]{amsmath}
```

with

```
\usepackage[leqno,fleqn,intlimits]{empheq}
```

empheq makes sure that the amsmath options are passed on and loaded by amsmath. The same goes for the options provided by mathtools.

2.1.1 Compatibility with the old version

The current version (v2.17) of empheq is incompatible with versions prior to and including 0.7e. If you have documents produced with versions prior to 0.7e, then

[5] By the same author and is distributed with empheq. See its documentation for more information.

you have to load the package empheq07 instead. empheq07 now exists as a separate package with its own documentation, but I strongly recommend switching to the new version because it is so much better.

2.2 Package options

In addition to the amsmath options, empheq itself provides a string of options listed in Table 4.

Option	Short description
`overload`	Lets you use the empheq visual markup extensions transparently in an existing document.
`overload2`	A wild version of `overload`. Use with care.
`ntheorem`	A support module for users of ntheorem and its `thmmarks` and `thref` options.
`newmultline`	With v2.10 of empheq the syntax for `multline` and `multlined` has been improved immensely.
`oldmultline`	Support for the somewhat strange syntax for `multline` and `multlined` in v2.00.

Table 4: Supported options in the empheq package

`ntheorem`

The ntheorem package is supported by means of the `ntheorem` option. In order for this to work properly, an amsmath bug[6] is fixed. For more information on this option see §3.2 on page 15.

2.2.1 Using `multline`

`newmultline`
`oldmultline`

Those familiar with v2.00 of this package will recognize the somewhat weird syntax for using `multline` and `multlined`. The good news is that since v2.10 there is now a much improved syntax available. The option `newmultline` (default) selects this better interface, while the old interface is provided for compatibility reasons by the `oldmultline` option.

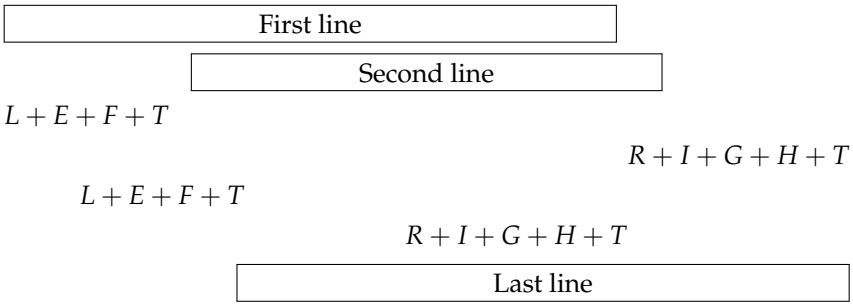
`\shoveleft[`⟨*dimen*⟩`]{`⟨*arg*⟩`}`
`\shoveright[`⟨*dimen*⟩`]{`⟨*arg*⟩`}`

With the new interface you also get an extended syntax for `\shoveleft` and `\shoveright` as shown in the example below.

```
\begin{empheq}{multline}
```

[6] See the LaTeX Bugs Database https://www.latex-project.org/cgi-bin/ltxbugs2html under AMS-LaTeX problem report 3624.

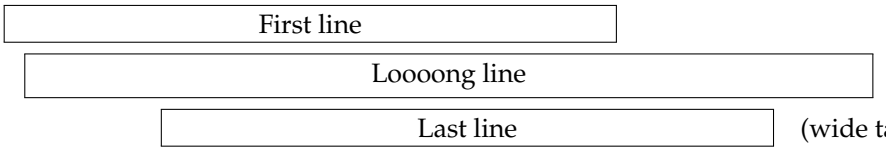
```
  \framebox[.65\columnwidth]{First line} \\
  \framebox[.5\columnwidth]{Second line} \\
  \shoveleft{L+E+F+T}          \\
  \shoveright{R+I+G+H+T}          \\
  \shoveleft[1cm]{L+E+F+T}          \\
  \shoveright[\widthof{$R+I+G+H+T$}]{R+I+G+H+T}          \\
  \framebox[.65\columnwidth]{Last line}
\end{empheq}
```



(14)

There are however a few differences in the output between the original amsmath version of `multline` and the one empheq provides. In amsmath a centered line in `multline` is centered on the page without taking into account the `\multlinegap`, `\multlinetaggap`, and the tag width. Thus amsmath can sometimes give you horrible output without giving you any warning as shown below.

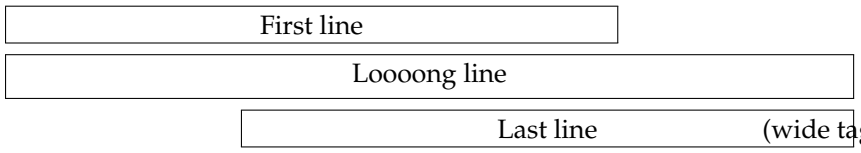
```
\begin{multline}
  \framebox[.65\columnwidth]{First line} \\
  \framebox[.9\columnwidth]{Loooong line} \\
  \framebox[.65\columnwidth]{Last line} \tag{wide tag}
\end{multline}
```



(wide tag)

In empheq these parameters are taken into account, so the same input will produce this in empheq:

```
\begin{empheq}{multline}
  \framebox[.65\columnwidth]{First line} \\
  \framebox[.9\columnwidth]{Loooong line} \\
  \framebox[.65\columnwidth]{Last line} \tag{wide tag}
\end{empheq}
```



(wide tag)

This results in an

```
Overfull \hbox (30.03783pt too wide) in paragraph ...
```

message in the log file, indicating a visual problem. I think this behavior is more sensible than the one the original `multline` environment provides.

`\begin{MTmultlined}[`⟨*pos*⟩`][`⟨*width*⟩`]` ⟨*contents*⟩ `\end{MTmultlined}`

When you choose the `oldmultline` option you may still want to use the much improved `multlined` environment defined in mathtools but unfortunately there will be a name clash. Instead you can access it under the name `MTmultlined`.

2.2.2 The `overload` option

`overload`
`overload2`

Feature request by Lars Madsen 2004/03/25

The `overload` option redefines the original $\mathcal{AMS}$ environments so that they take an optional argument.

`\begin{`⟨$\mathcal{AMS}$ *env_name*⟩`}[`⟨*markup instructions*⟩`]`
⟨*contents of* $\mathcal{AMS}$ *environment*⟩
`\end{`⟨$\mathcal{AMS}$ *env_name*⟩`}`

For example

```
\begin{gather}[box=\widefbox]
  a = b
\end{gather}
```

is then actually short for

```
\begin{empheq}[box=\widefbox]{gather}
  a = b
\end{empheq}
```

All the $\mathcal{AMS}$ environments are supported by this option except for the pseudo environment `\[ ... \]` (it's not really an environment), [7] because a) you don't really need markup instructions for a one line display, and b) would you really like a syntax like

```
\[[box=\fbox] a=b \]
```

where it's difficult to see whether or not there is a typo? Choosing `overload` is meant for people who don't want to change their entire document into empheq syntax. I have no problems with that; just be careful when you fiddle with `\empheqset` as it will affect *all* math displays!

There is of course a catch (well, two actually): These redefined environments don't run as fast as the regular ones (about three times as slow), but in this day and age I seriously doubt you'll be able to tell the difference unless you have a vast number of equations. The other catch is that you cannot use `\intertext`

[7] With the option `oldmultline` the `multline` and `multline*` environments aren't supported either because their syntax in empheq then differs from their syntax in amsmath.

and `\displaybreak` as described earlier. If you find yourself wanting to use one of these features in say, an `align` environment, you have to use the original `align` environment. Luckily it is available if you call it like this:

```
\begin{AmS⟨AMS env_name⟩}
  ⟨contents of AMS environment⟩
\end{AmS⟨AMS env_name⟩}
```

For example the original `align` environment could be selected with

```
\begin{AmSalign}
  ...
\end{AmSalign}
```

These original versions with prefix `AmS` exist for all the environments.

The option `overload2` activates the overloading feature for the pseudo environment `\[ ... \]`, although I doubt you'll find it useful. Beware that this definition is fragile unless you have ε-TEX as LATEX engine.[8] Not surprisingly `overload2` activates `overload`.

Before you get all excited about these options, you should take note of some aspects of centered math displays in amsmath. In certain circumstances truly centering the display is not always the best solution as in this example:

Wide math display; not adjusted Wide Tag

Instead centering the display in the available space works pretty well:

Wide math display; adjusted Wide Tag

This is what amsmath does normally, but in the `gather` environment it does it on a per line basis. This means that we get results like

Wide math display; adjusted Wide Tag

Wide math display; not adjusted (i)

whether we like it or not. Inside the `empheq` environment we need to have the same adjustment for all lines else the boxing process will not work properly, so when activating the `overload` option the above two-line `gather` will instead look like this:

Wide math display; adjusted Wide Tag

Wide math display; adjusted i

I leave it to you to choose whether or not this is better (I think it is better).

[8] ε-TEX has been the default engine for LATEX in most major distributions since 2003.

2.3 A note on boxed displays

When browsing a 400+ pages textbook with at least twice the number of displayed formulae, some of them are surely more important than others. Thus the author of the book (in cooperation with the designer) should make sure that such formulae are easily found again i.e., they should be easily distinguishable from the rest of the pack. One way of doing this is putting the formula into a box which is something we've seen empheq being capable of. There are however a few things to keep in mind:

- Don't overdo it. If you do it on half of them there's no point in doing it at all. I don't see much reason for applying this technique to more than 10% of the formulae.
- Choose the type of box carefully. You want to draw attention to it so it might as well look good.

The latter point can be illustrated by defining a macro similar to `\fbox` only with a little more space to the right and left of the argument.

```
\newcommand*\widefbox[1]{\fbox{\hspace{1em}#1\hspace{1em}}}
```

If we replace the `\fbox` from the box example before, we get this display:

```
\begin{empheq}[box=\widefbox]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$\boxed{\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned}} \tag{15--16}$$

Compare it with

$$\boxed{\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned}} \tag{17--18}$$

and see which one you prefer.

Similarly one might be tempted to use colored boxes:

```
\definecolor{myblue}{rgb}{.8, .8, 1}
\newcommand*\mybluebox[1]{%
  \colorbox{myblue}{\hspace{1em}#1\hspace{1em}}}
```

We know the drill by now.

```
\begin{empheq}[box=\mybluebox]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$\colorbox{myblue}{$\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\,dx \end{aligned}$} \tag{19--20}$$

There is more on boxes later in this manual. If you're into the fancybox package then remember to read §3.1 on the next page.

3 Support for other packages

With the multitude of packages for LaTeX, it is not always easy to be a PWWO package ("Plays Well With Others"), but empheq tries really hard to do so. This section lists the packages where empheq has to provide workarounds and they can be divided into two categories.

Compatibility Some packages affect LaTeX's labelling mechanism and since empheq internally has to turn off labels and related code temporarily, hooks must be provided for these packages. Examples of such packages are hyperref and showkeys.

Enhancements Other packages provide useful features that for some reason may not work directly or optimally with empheq. In these cases the problematic commands are redefined so that they not only work with empheq, but also give the same if not better output. An example of this would be the `\shadowbox` command from fancybox (described below).

3.1 Support for fancybox

The fancybox package provides various boxes and you can use them with empheq as well. Here's `\ovalbox`:

```
\begin{empheq}[box=\ovalbox]{align}
    E&=mc^2 \\
    Y&= \sum_{n=1}^\infty \frac{1}{n^2}
\end{empheq}
```

$$E = mc^2 \tag{21}$$

$$Y = \sum_{n=1}^{\infty} \frac{1}{n^2} \tag{22}$$

`\shadowbox*{`⟨*arg*⟩`}`
`shadowcolor`

The only problem with using the fancybox boxes in conjunction with the `empheq` environment is `\shadowbox`, as this macro typesets the shadow on the baseline and for the equation numbers to be placed correctly, the box command must set its argument on the baseline. Thus the normal `\shadowbox` produces ⟨*arg*⟩, but we need this instead: ⟨*arg*⟩. Therefore empheq will detect if fancybox is loaded and in that case it'll enhance `\shadowbox` in two ways:

- It defines a starred version `\shadowbox*` which typesets its argument on the baseline.
- The color `shadowcolor` is introduced. The default color is black.

```
\definecolor{shadowcolor}{rgb}{0,.5,.5}
\setlength\shadowsize{2pt}
```

```
Line of text for \shadowbox*{testing of \Verb|f#&%\| verbatim} and
showing the shadow color.
```

Line of text for testing of f#&%\ verbatim and showing the shadow color.

The point is that if you want a \shadowbox around your math display then you must use the starred version:

```
\begin{empheq}[box=\shadowbox*]{align}
    E&=mc^2 \\
    Y&= \sum_{n=1}^\infty \frac{1}{n^2}
\end{empheq}
```

$$E = mc^2 \tag{23}$$

$$Y = \sum_{n=1}^{\infty} \frac{1}{n^2} \tag{24}$$

See §6.2 on page 21 if you want to make a fancy box yourself.

3.2 Support for ntheorem

Caveat: Due to an unfortunate interaction between empheq and ntheorem, users may want to add

```
\usetagform{default}
```

after loading ntheorem otherwise the tags may be placed wrong and any labels within the empheq environment may be lost. In the future ntheorem should be able to test for this and add \usetagform{default} automatically.

Users who use \usetagform to get another tag design should make sure to postpone this configuration until *after* loading ntheorem.

ntheorem

The ntheorem package is supported by the means of the ntheorem option of empheq. This loads a set of extra macros which fixes various compatibility problems between ntheorem and amsmath and furthermore introduces special (internal) macros for optimum positioning of end-of-theorem markers, while retaining a user friendly interface.

When you want to use the automatic end-of-theorem marker mechanism from ntheorem you sometimes run into problems as you would want the marker to be placed aligned at the bottom of the math display but still keeping the tags in their proper place. In [leqno] mode this is not that much of a problem as the tags and the marker are set on either side of the math display like in Figure 1 on the next page.

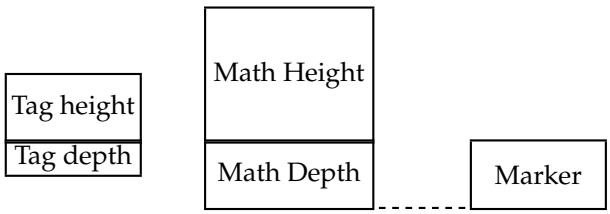


Figure 1: Marker placement in `[leqno]` mode

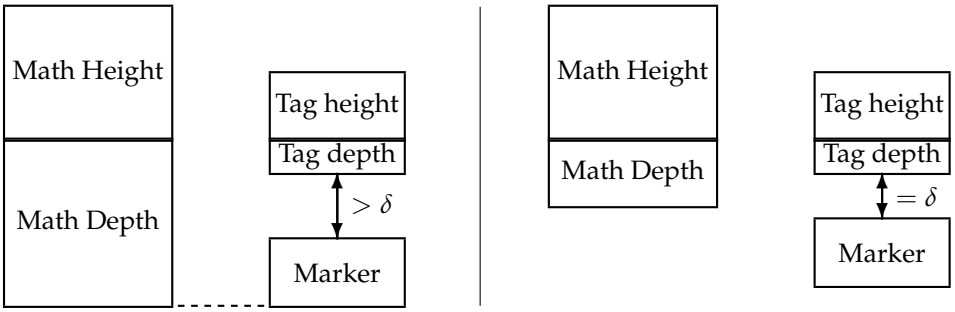


Figure 2: The two possible situations for end marks in `[reqno]` mode

`\mintagvsep`

Unfortunately things are not this easy in `[reqno]` mode. There are two possible situations as shown in Figure 2. If possible we want the bottom of the marker to be aligned with the bottom of the math display, but at the same time we want to ensure a minimum vertical separation δ between marker and tag. In empheq this is controlled by the length parameter `\mintagvsep` which by default is 5 pt.

The good news is that this is where empheq sets in. Basically all you need to do is to use the empheq environment to typeset your equations inside the theorem environments:

```
\begin{Theorem}
Some text at first and then a math display
\begin{empheq}{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
\end{Theorem}
```

Then the tag placement and the end-of-theorem marker will be set properly (after a couple of runs as usual).

To ensure the correct outcome remember to load the packages like this:

```
\usepackage[ntheorem]{empheq} % this loads amsmath as well
\usepackage[thmmarks,amsmath]{ntheorem}
```

Remember that if you use the `overload` option you can just use the regular math environments to get the desired result with the end mark. The exception is the `\[ ... \]` environment which will only work if you use the option `overload2`.

Part II
Advanced user's guide

The empheq package has more to offer than you have seen, but I found some of the functionality so dangerous it was best to hide it for a little while. The commands you have encountered so far in this manual all have one thing in common: They have only lower case letters in their names. Now it's time to reveal those that have *mixed-case* names, thus implying that you should take great care when using them.

4 Delimiters revisited

Let's go back to delimiters, shall we?

4.1 Creating your own delimiters

As a convenience for the user, this interface is extended in a more general way so that it is also possible to declare delimiters with the following two commands:

`\DeclareLeftDelimiter[`⟨*space adjustment*⟩`]{`⟨*delimiter*⟩`}`
`\DeclareRightDelimiter[`⟨*space adjustment*⟩`]{`⟨*delimiter*⟩`}`

While empheq provides auto-scaling versions of the most common delimiters, you may sometimes want some new ones. Say for instance you have loaded the stmaryrd package and you want to use the double bracket commands `\llbracket` and `\rrbracket` with empheq. Then you simply do this:

```
\DeclareLeftDelimiter{\llbracket}
\DeclareRightDelimiter{\rrbracket}
```

This defines the new delimiters `\empheqllbracket`, `\empheqbigllbracket`, `\empheqrrbracket` and `\empheqbigrrbracket`.

You can use `\big...` delimiters as well if you don't like the automatic scaling. There are however ways to fine-tune if you really want it.

4.2 Fine-tuning of delimiters

TEX provides two primitives to control the scaling of delimiters produced with `\left` and `\right`, namely the dimension `\delimitershortfall` (denoted by δ here) and the integer `\delimiterfactor` (f). The idea is that the sub-formula inside the `\left`-`\right` pair is to be vertically centered and given its height h_1 and its depth h_2 we want to produce a delimiter with total height h, where $h = 2\max(h_1, h_2)$. TEX's rules on this are that the minimum delimiter size $h_{\min}$ must meet the requirements

$$h_{\min} \geq h\frac{f}{1000} \quad \wedge \quad h_{\min} \geq h - \delta$$

LaTeX sets `\delimitershortfall` = 5.0pt and `\delimiterfactor` = 901, but in our case we will almost always want a delimiter that spans the entire `\left`-`\right` pair (the math display), thus a change of these settings is needed.

```
\EmphEqdelimitershortfall
\EmphEqdelimiterfactor
```

However it is a bad idea to just change these two settings without thinking of the effect it'll have on the rest of the mathematics in your document. Therefore empheq provides the parameters `\EmphEqdelimitershortfall` (default setting is 1.0pt) and `\EmphEqdelimiterfactor` (default is 950) to cater for this.

4.3 Scaling material yourself

The attentive reader may have noticed that I still haven't revealed how I managed to get the right size of that big exclamation mark in the first example of this manual, so I guess it's about time.

```
\EmphEqdisplayheight
\EmphEqdisplaydepth
```

The height of the math display plus the surrounding inner box is given by `\EmphEqdisplayheight` and the depth by `\EmphEqdisplaydepth`. These can be used when constructing something that should scale to fit the display and can't be done with "auto-scaling" commands like `\vrule` or `\left` or `\right`. There is a catch however: If you do any horizontal resizing of the material you want to scale, then you must specify the width *manually*:[9]

```
\begin{empheq}[
  box=\colorbox{myblue},
  right={\;
         \makebox[.9em]{%
          $\raisebox{-.5\totalheight+\fontdimen22\textfont2}
            {\resizebox{!}{%
              \EmphEqdisplayheight+\EmphEqdisplaydepth}{!}}$%
         }},
  left={X=Y\Rightarrow}
   ]{alignat=3}
  A_1&=b_1 & \qquad c_1&=d_1 & \qquad e_1&= f_1
       \tag*{\_A\raisebox{1ex}{silly}\raisebox{-1ex}{tag}\_}\\
  A_2&=b_2 & \qquad c_2&=d_2 & \qquad  e_2&= f_2 \\
  A_3&=b_3 & \qquad c_3&=d_3 & \qquad  e_3&= f_3
\end{empheq}
```

Here's another example. We want to be able to put a `\parbox` with some descriptive text top-aligned on the side of the display.

```
\begin{empheq}[
  left={\parbox[c][\EmphEqdisplayheight+\EmphEqdisplaydepth][t]
```

[9] The `\fontdimen22\textfont2` in the ⟨*length*⟩ argument of `\raisebox` is the *math axis* of the font. It is needed in order to get the vertical positioning of the oversized exclamation mark just right.

```
        {4.5cm}
        {You may find this kind of description useful.}\enspace}]
    {align}
  a&=\int_0^1 x\,dx +\frac{foo + bar}{baz}\\
  E&= mc^2
\end{empheq}
```

You may find this kind of description useful.

$$a = \int_0^1 x\,dx + \frac{foo + bar}{baz} \tag{25}$$

$$E = mc^2 \tag{26}$$

5 A few short notes

5.1 About \eqref

Internally empheq separates the displayed math from the corresponding equation numbers such that we can add special delimiters or boxes. This is done by nullifying (or rather conveniently redefining) an internal command called \maketag@@@. Unfortunately, this component is also used by \eqref to typeset a reference to an equation number, so in earlier versions, \eqref would not work inside an empheq environment.

In the current version this should now be working, but if you ever need to mess with \eqref it may be handy to know what is being done. In essence we do something similar to this:

```
\let\empheqeqrefbase\textup
 ... % next go inside empheq env
\renewcommand\eqref[1]{\empheqeqrefbase{%
     \let\maketag@@@\EQsavedmaketag%
     \tagform@{\ref{##1}}}}
```

Thus if you need to alter things inside \eqref and need that to work within empheq as well, you may get away with redefining \empheqeqrefbase.

5.2 About changes to \baselineskip

Users should never mess with \baselineskip directly, it is not the correct manner to get double spacing. Have a look at say the setspace package or similar, or play with \baselinestretch followed by \normalsize to initiate.

Explanation: We use \parbox inside to place the display box and the box containing the eqn numbers. To do its stuff \parbox resets some internal settings, in this case, \baselineskip is reset to \normalbaselineskip. So if you don't remember to reset that one as well...

6 Creating something new

You can create your own environments and boxes to go with empheq, but there are certain things that must be fulfilled to get it to work properly.

6.1 New empheq-like environments

```
EmphEqMainEnv
\EmphEqMainEnv
\endEmphEqMainEnv
```

Although the real work in empheq is done by an environment (EmphEqMainEnv), you cannot use it like that yourself when you want to define an empheq-like environment. Due to technical reasons (see [4]), you have to use a slightly different syntax:

```
\newenvironment {⟨env_name⟩} [⟨num⟩] [⟨default⟩]
  {⟨other commands⟩ \EmphEqMainEnv}
  {\endEmphEqMainEnv}
```

In the above environment the ⟨*other commands*⟩ part must contain certain things before it'll work.

```
EmphEqEnv
EmphEqOpt
```

Before your homemade environment will work you must set at least one key. The key family EmphEqOpt is for all the markup instructions like left and box, while EmphEqEnv controls the type of $\mathcal{AMS}$ environment. You must set the EmphEqEnv family before \EmhpEqMainEnv else you'll get an error message.

One example could be something like this:

```
\newenvironment{important}[2][]{%
  \setkeys{EmphEqEnv}{#2}%
  \setkeys{EmphEqOpt}{box=\mybluebox,#1}%
  \EmphEqMainEnv}%
  {\endEmphEqMainEnv}
```

Thus

```
\begin{important}{gather}
  a = b + c +d \\ e = f
\end{important}
```

produces

$$a = b + c + d \tag{27}$$
$$e = f \tag{28}$$

while

```
\begin{important}[left={A=B \Rightarrow \empheqlbrace}]{alignat=2}
  a &= b &\quad c &= d \\
  \text{this} &= \text{that} &\quad \mathit{fish}&\neq fish
\end{important}
```

produces

$$A = B \Rightarrow \begin{cases} a = b & c = d \\ \text{this} = \text{that} & \mathit{fish} \neq \mathit{fish} \end{cases} \tag{29}$$
$$\tag{30}$$

6.2 Creating fancy boxes

As a final example I will show you how to create complicated displays involving (vertically) asymmetrical boxes like `\shadowbox*`. In order to get the correct output, the contents of the box must be placed on the baseline as in `\fbox`. In this example we want to put a set of equations into a bright yellow box and then add another box with some explanatory text at the top of the yellow box making them overlap.

First we define the colors used and allocate the boxes. We could probably use scratch boxes, but this is safer.

```
\definecolor{shadecolor}{cmyk}{0,0,0.41,0}
\definecolor{light-blue}{cmyk}{0.25,0,0,0}
\newsavebox{\mysaveboxM} % M for math
\newsavebox{\mysaveboxT} % T for text
```

Save the display body in `\mysaveboxM` and the text argument in `\mysaveboxT`.

```
\newcommand*\Garybox[2][Example]{%
  \sbox{\mysaveboxM}{#2}%
  \sbox{\mysaveboxT}{\fcolorbox{black}{light-blue}{#1}}%
```

Then typeset the math display in a `\parbox` where we control the height and save it in `\mysaveboxM`.

```
  \sbox{\mysaveboxM}{%
    \parbox[b][\ht\mysaveboxM+.5\ht\mysaveboxT+.5\dp\mysaveboxT][b]{%
      \wd\mysaveboxM}{#2}%
  }%
```

We put it into the colored box with the desired width.

```
  \sbox{\mysaveboxM}{%
    \fcolorbox{black}{shadecolor}{%
      \makebox[\linewidth-10em]{\usebox{\mysaveboxM}}%
    }%
  }%
```

Then finally we get to the real typesetting. We just insert the math display and then make a box of zero width right next to it. In that box we lift the text argument and center it at the top of the display.

```
  \usebox{\mysaveboxM}%
  \makebox[0pt][r]{%
    \makebox[\wd\mysaveboxM][c]{%
      \raisebox{\ht\mysaveboxM-0.5\ht\mysaveboxT
                +0.5\dp\mysaveboxT-0.5\fboxrule}{\usebox{\mysaveboxT}}%
    }%
  }%
}
```

Let's see what it looks like.

```
\begin{empheq}[box=\Garybox]{align}
  \sum \mathbf{F} &= \mathbf{0} \\
  \sum F_{x} \,\mathbf{i} + \sum F_{y} \,\mathbf{j}
```

```
    + \sum F_{z}  \,\mathbf{k} &= \mathbf{0} \\
  \sum F_{x} &= 0 \\ \sum F_{y} &= 0 \\ \sum F_{z} &= 0
\end{empheq}
```

Example

$$\sum \mathbf{F} = \mathbf{0} \tag{31a}$$
$$\sum F_x\,\mathbf{i} + \sum F_y\,\mathbf{j} + \sum F_z\,\mathbf{k} = \mathbf{0} \tag{31b}$$
$$\sum F_x = 0 \tag{31c}$$
$$\sum F_y = 0 \tag{31d}$$
$$\sum F_z = 0 \tag{31e}$$

If we use the optional argument of `\Garybox` we have to enclose the entire argument of box in braces because we're in `empheq`'s optional argument (as described in [1, page 167]):

```
\begin{empheq}[box={\Garybox[Same old example again]}]{align}
  a&=b\\
  E&=mc^2 + \int_a^a x\, dx
\end{empheq}
```

Same old example again

$$a = b \tag{31f}$$
$$E = mc^2 + \int_a^a x\,dx \tag{31g}$$

7 Contact information

Should you have any feature request, suggestions, or bug reports then feel free to open an issue at (it also covers empheq)

https://github.com/latex3/mathtools

Contributors

- In November 2002 Lars Madsen (daleif@imf.au.dk) asked for some features that wasn't readily available with amsmath or any other package. This was the start of empheq.
- Gary Gray (gray@engr.psu.edu) gave me some bug reports on the old version (v0.7) which convinced me to rewrite the package completely. The `\Garybox` example is inspired by a wish from Gary.
- Uwe Siart (uwe.siart@tum.de) has been a thorough beta-tester on v1.00 of the package.
- Bernard Alfonsi (alfonsi@math.u-psud.fr) reported a problem with the `ntheorem` option. It turned out to be a bug in the ntheorem package itself but it's fixed in the option.

- I have received reports on weird behavior from empheq when used with the color package on some dvi-viewers, which is caused by the lack of color support in those dvi-viewers. Both Andrew B. Collier (colliera@ukzn.ac.za) and André M. de Roos (aroos@science.uva.nl) have notified me of this. The solution is to either upgrade your dvi-viewer if possible or convert the document in question to PDF or PS.

Thank you all.

References

[1] Leslie Lamport, *LaTeX User's Guide and Reference Manual*, 2nd edition, Addison-Wesley, Reading, Massachusetts, 1994.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin, *The LaTeX Companion*, Addison-Wesley, Reading, Massachusetts, 1994.

[3] Donald Ervin Knuth, *The TeXbook*, Addison-Wesley, Reading, Massachusetts, 1986.

[4] American Mathematical Society and Michael Downes, *Technical notes on the amsmath package* Version 2.0, 1999/10/29. (Available from CTAN as file `technote.tex`.)

[5] Frank Mittelbach, Rainer Schöpf, Michael Downes, and David M. Jones, *The amsmath package* Version 2.13, 2000/07/18. (Available from CTAN as file `amsmath.dtx`.)

[6] Timothy Van Zandt, *The fancybox package* Version 1.3, 2000/09/19. (Available from CTAN as file `fancybox.sty`.)

[7] Sebastian Rahtz, *Hypertext marks in LaTeX* Version v6.74m, 2003/11/30. (Available from https://www.tug.org/applications/hyperref as file `hyperref.dtx`.)

[8] David Carlisle, *The keyval Package*, Version 1.13, 1999/03/16. (Available from CTAN as file `keyval.dtx`.)

[9] Kresten Krab Thorup, Frank Jensen, and Chris Rowley. *The calc package*, Version 4.1b, 1998/07/07. (Available from CTAN as file `calc.dtx`.)

[10] Wolfgang May and Andreas Schlechte, *The ntheorem package* Version 1.203, 2002/01/07. (Available from CTAN as file `ntheorem.dtx`.)

[11] David Carlisle. *The showkeys package*, v3.12, 1997/06/12. (Available from CTAN as file `showkeys.dtx`.)