# The physics2 package

## Zhang Tingxuan

2023/04/02 Version 0.2.2\*

## Abstract

This is the document for physics2 package, which defines commands for typesetting math formulae faster and more simply. physics2 is a modularized package, each module provides its own function.

This document describes the physics2 package in more detail. But if you are a user of the legacy physics package, you can click here to see the documentation for physics users before you start. If you never used physics package before, just read *this* documentation.

## **Contents**

1	Intr	oduction	2
	1.1	The purpose of this package	2
	1.2	Packages required	2
	1.3	Loading the physics2 package	2
	1.4	Loading a module of physics2	2
2	Modules of physics2		
	2.1	The automatically loaded common module	3
	2.2	The ab module — automatic braces	4
	2.3	The ab.braket module — Dirac bra-ket notation	5
	2.4	The braket module — Dirac bra-ket notation	7
	2.5	The diagmat module — simple diagonal matrices	8
	2.6	The double product operator	9
	2.7	The xmat module — matrices with formatted entries	10

<sup>\*</sup>https://www.github.com/AlphaZTX/physics2

3	The	"legacy" modules	11
	3.1	The ab.legacy module	12
	3.2	The bm-um.legacy module	12
	3.3	The nabla.legacy module	13
	3.4	The op.legacy module	13
	3.5	The qtext.legacy module	14

## 1 Introduction

## 1.1 The purpose of this package

This package aims to provide a bundle of commands for typesetting math faster in different modules. The commands provided by physics2 and its different modules are designed to be short and easy to memorize.

#### 1.2 Packages required

The physics2 package itself only requires the keyval package, which is part of the latex-graphics bundle. Almost every LATEX distribution will include this bundle.

Different modules of physics2 might require different packages. It will be explained in the following sections that which module requires which package.

The physics2 package requires  $\LaTeX$  2 $_{\mathcal{E}}$  kernel released after 2020/10. Please make sure that your  $\LaTeX$  distribution is not too old.

## 1.3 Loading the physics2 package

Just like loading any package, write

```
\usepackage{physics2}
```

in the preamble to load the physics2 package. In this version, physics2 doesn't provide a package option.

However, physics2 itself only provides very few functions. Actually, it just provides a method to load modules. You need to load different modules of physics2 to have different kinds of functions applied to your document.

#### 1.4 Loading a module of physics2

You can load a module of physics2 only *after* you write \usepackage{physics2} in the preamble. Load a physics2 module like this:

```
\userbysicsmodule(\langle module \rangle)
```

The usage of \usephysicsmodule is similar to \usepackage, so you can load several modules in one line. For example,

```
\usephysicsmodule{ab,ab.braket}
```

This line loads the ab and ab.braket modules.

You can also load *one* module with options. The options of a physics2 module can be a comma-separated key-value list. For example,

```
\usephysicsmodule[tightbraces=true]{ab}
\usephysicsmodule{ab.braket,doubleprod}
```

These two lines load the ab module with option tightbraces = true and load ab.braket and doubleprod modules.



The common module will be loaded automatically when you load the physics2 package and *only* the common module will be loaded automatically. Any other module should be loaded manually by writing \usephysicsmodule  $\{\langle module \rangle\}$  after you loaded physics2 in the preamble.

The following sections introduce all the user-level modules of physics2. View back to the table of contents to see the names of user-level modules.

## 2 Modules of physics2

## 2.1 The automatically loaded **common** module

The common module provides the following commands:

\delopen and \delclose, followed by a math delimiter. They can be regarded as abbreviations of "open delimiter" and "close delimiter". If you had heard of the mleftright package. You can regard \delopen and \delclose as a simpler version of \mleft and \mright. For example,

\biggg and \Biggg, followed by a math delimiter. They are even bigger than \Bigg. \biggg and \Biggg may be useful when you need to write something really tall in math mode, but most OpenType math font do not support \langle (or U+27E8) and \rangle (or U+27E9) in this large size. Take an example,

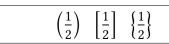


\bigggl, \bigggm, \bigggr, \Bigggl, \Bigggm and \Bigggr are also supported.

#### 2.2 The ab module — automatic braces

This module provides the command \ab. The \ab command, as a shorthand of "automatic braces", would specify the size of the following pair of delimiters. The delimiters after \ab should not be out of the range described by the following chart:

For example, it's illegal to write an "\ab(" without a ")"; it's also illegal to write \ab=foo=. Take some correct examples:



You can also write a command from  $\big$  to  $\Biggg$  between  $\ab$  and the first delimiter, which means to specify the size of delimiters manually. Also, you can write a star (\*) between  $\ab$  and the first delimiter, to prevent  $\ab$  from setting the size of delimiters. For example,





Always remember, do not put an \ab separately at the end of math mode like \$\ab\$, because \ab will try to absorb the following math shift character (\$) as its argument.

The ab module also provides  $\X$  ab commands, where X can be p, b, B, a, v and  $\forall$ . These commands take a normal argument but not an argument delimited with paired delimiters. For example,

These  $\X$ ab commands can take an optional star and an optional  $[\langle biggg \rangle]$  argument. Star stands for using the default sizes. For example,

The options of ab module tightbraces, a bool type key, whose default value is true, influences whether thin skips are reserved around the paired delimiters. It only works with the automatically sized delimiters.

#### 2.3 The ab.braket module — Dirac bra-ket notation

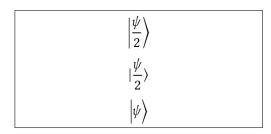
This module provides four commands — \bra, \ket, \braket and \ketbra. After these commands can be a star (\*) or a "biggg" command. These commands share similar syntaxes like \ab's syntax. But, the bra-ket commands from ab.braket module are completely different from \ab. Their internal structures are different.

The argument of \bra should be delimited with < and |, that is,

For example,

The argument of \ket should be delimited with | and >, that is,

For example,



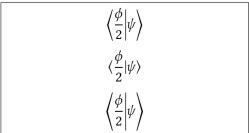


If you want to write ">" and "<" for relations in the argument of \bra and \ket, you can write \mathrel{>} and \mathrel{<} (although there is almost no such need).

The argument of \braket should be delimited with < and >, that is,

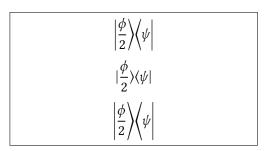
In the *(subformula)* argument, every "|" will be regarded as an extensible vertical bar. For example,

\def\0{\frac\phi2} \[\braket < \0 | \psi > \] \[\braket\* < \0 | \psi > \] \[\braket\Bigg< \0 | \psi > \]



The argument of \ketbra should be delimited with | and |. In the argument, > and < will be regarded as extensible > and <. That is,

For example,







If you want to write ">" and "<" for relations in the argument of \braket and \ketbra, you can write \> and \< (although there is almost no such need). It is quite different from \mathrel{>} or \mathrel{<} because in these commands' argument, > and < will be redefined.

Next, the braket module will be introduced. Please notice that braket is conflict with ab.braket, they cannot be used together.

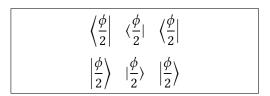
#### 2.4 The **braket** module — Dirac bra-ket notation

Please notice that this module is conflict with the ab.braket module. Don't use them together.

This module contains four commands — \bra, \ket, \braket and \ketbra. After these commands can be a star (\*) or a square-bracket-delimited size option, the size option can take the following values:

Star stands for "do not size the bra-ket automatically".

The argument(s) of these four commands are braced with { and }. \bra and \ket take one mandatory argument. For example,



The \braket command, in default, can take two arguments.

If you want \braket to take one or three arguments, you can write the number of arguments in the square bracket. If you need to specify the size of bra-ket simultaneously, you need to separate the number and the size with a comma:

The \ketbra command takes two mandatory arguments. It can also take an optional argument between the two mandatory arguments. The optional argument will be placed between  $\rangle$  and  $\langle$ :

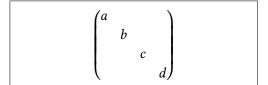
## 2.5 The diagmat module — simple diagonal matrices

This module provides \diagmat command:

```
\forall diagmat[empty = \langle empty entry \rangle] \{\langle diag \rangle\}
```

where  $\langle diag \rangle$  is the diagonal of the diagonal matrix. The entries should be separated by commas. The empty option is optional, with default value 0. For example,

\pdiagmat, \bdiagmat, \bdiagmat, \vdiagmat and \Vdiagmat are also available. Prefixes like p, b, B have the same meaning as the p, b, B in amsmath's pmatrix, bmatrix and Bmatrix. For example,



This module requires amsmath.

The options of diagmat module You can set the default value of \diagmat's empty entries in the module option like this:

\usephysicsmodule[empty={\cdot}]{diagmat}

## 2.6 The **doubleprod** module — tensors' double product operator

Take an example of this module:

\doublecross and \doubledot are regarded as binary operators by TFX.

The options of doubleprod module You can control the scale of "x" and "·" in \doublecross and \doubledot in module option. For example,

```
\usephysicsmodule[crossscale=0.75,dotscale=1.2]{doubleprod}
```

The default values of crossscale and dotscale are 0.8 and 1. You can also control the distances between the two "x"s and ":"s through the crossopenup and dotopenup options. For example,

```
\usephysicsmodule[crossopenup=.05,dotopenup=.25]{doubleprod}
```

The default values of crossopenup and dotopenup are 0.02 and 0.2. The value stands for the multiple of current font size. Moreover, you can change the symbols produced by \doublecross and \doubledot by setting crosssymbol and dotsymbol in module option.

## 2.7 The **xmat** module — matrices with formatted entries

The xmat module provides \xmat command for matrices with formatted entries:

```
\mbox{\cols shown} {\cols shown} {\cols shown}
```

If  $\langle rows \ shown \rangle$  and  $\langle cols \ shown \rangle$  are digits, the value of them must be less at least 2 than the value of amsmath's MaxMatrixCols counter. For example,

 $a_{11}$ 

 $a_{21}$ 

 $a_{12}$   $a_{13}$ 

 $a_{23}$ 

 $a_{22}$ 

\pxmat, \bxmat, \Bxmat, \vxmat and \Vxmat are also available. The meaning of p and so on is the same as the p in pmatrix of amsmath. For example,

If  $\langle rows shown \rangle$  and  $\langle cols shown \rangle$  contain non-digit characters, extra dots will be added. For example,

In this example we used the showleft and showtop options. The default value of them is the value of MaxMatrixCols minus 2. You can also set them in the module option like this:

```
\usephysicsmodule[showtop=3,showleft=3]{xmat}
```

Then every \xmat with non-digital  $\langle rows \ shown \rangle$  and  $\langle cols \ shown \rangle$  will have 2 topmost rows and 3 left-most columns shown. This will also influence "\xmat"s with digital  $\langle rows \ shown \rangle$  and  $\langle cols \ shown \rangle$  when  $\langle rows \ shown \rangle$  and  $\langle cols \ shown \rangle$  are larger than the values corresponding to showtop and showleft. For example,

However, when  $\langle rows \ shown \rangle$  and  $\langle cols \ shown \rangle$  are 1 greater than  $\langle showtop \rangle$  and  $\langle showleft \rangle$ , for example,  $\langle rows \ shown \rangle = 4$  and  $\langle cols \ shown \rangle = 4$  in last example's settings,  $\rangle$  was will still add the extra dots:

In such situations, we need to specify showtop and showleft manually. For example,

The \xmat command provides the format option, which allows users to use a new entry format. For example,

```
\[[2.7.7] \text{\showleft=2,showtop=2,} \text{\showleft=2,showtop=
```

In the value of format key, #1 stands for the common entry, or the first mandatory  $\langle entry \rangle$  argument of  $\rangle = 2$  stands for the row index and #3 stands for the column index.

This module requires amsmath.

The options of xmat module Only showtop and showleft can be used as module options. format should be only used in the optional argument of the \xmat command.

## 3 The "legacy" modules

The legacy modules have similar names like  $\langle module \rangle$ . legacy. Most of them are designed to provide solutions to maintain documents written with the legacy physics package. It's not suggested to use them in a new document.

## 3.1 The ab.legacy module

This module provides the following commands:

```
\abs \norm \eval (\peval \beval) \order
```

They share the same syntax as  $\langle cmd \rangle^* [\langle biggg \rangle] \{\langle subformula \rangle\}$ . Star and  $\langle biggg \rangle$  are optional. Star stands for "use the default size". For example,

You can set the "order" symbol in this module through the order option like this:

```
\usephysicsmodule[order=0]{ab.legacy}
```

For further information of this module, see §2.1 of physics2-legacy.

#### 3.2 The bm-um.legacy module

If you are maintaining a document with plenty of "\bm"s or "\boldsymbol"s in it but want to use unicode-math package simultaneously, you could take a look at this module.

The \bm command from bm package uses \mathversion to support its function, but there are few OpenType math fonts who released with a bold version. The bm-um.legacy module provides a \bm command too, but this \bm can only take *one* math character or a series of math characters sharing the same category code as its argument. If the argument was Latin letters or Greek letters, \bm would switch to the bold italic glyphs corresponding to them (if there exists bold italic glyphs); else \bm would switch to the bold upright glyphs. For example,

## 3.3 The nabla.legacy module

This module provides some commands related to nabla  $(\nabla)$ . Notice that this module requires the fixdif package with file date 2023/01/31 at minimum.

This module defines \grad and \curl and redefines \div. For example,

The "÷" symbol was redefined as \divsymbol.

## 3.4 The op.legacy module

This module provides a series of commands for log-like operators. They are

where \Re and \Im are redefined. The first four lines of commands yield what they look like in math mode. For example,

\PV yields " $\mathcal{P}$ " as an ordinary symbol and \pv yields "p.v.". For example,

\Re and \Im are redefined as "Re" and "Im".  $\Re$  and  $\Im$  are redefined as \Resymbol and \Imsymbol, in default.

This module *does not* require amsmath.

The options of op.legacy module ReIm, a bool key with default value true, determines whether to redefine \Re and \Im. If you want to reserve the definition of \Re and \Im, you can write like this:

\usephysicsmodule[ReIm=false]{op.legacy}

## 3.5 The qtext.legacy module

This module was written just to offer a method to maintain documents written with the legacy physics package. See §2.4 of physics2-legacy for more information.