physics2 manual for the legacy physics users

Zhang Tingxuan

2023/04/02 Version 0.2.1[*]

Abstract

This short document describes physics2 package for those who are used to the physics package. This document is only a simple reference manual for:

- Frequent users of the legacy physics package;
- Those who have to maintain a document written with physics;
- Users who failed to use unicode-math with physics.

It seems no reason for any other user to read *this* document instead of the package documentation of physics2, because this document cannot describe the package in detail.

In this document, the modules of physics2 will be introduced in the same order as the physics documentation.

Contents

1 Before you start **2**
1.1 Legacy problems with physics package 2
1.2 Loading physics2 2

2 List of commands **3**
2.1 Automatic bracing 3
2.2 Vector notation 5
2.3 Operators 6
2.4 Quick quad text 7
2.5 Derivatives 7
2.6 Dirac bra-ket notation . . 7
2.7 Matrix macros 11

[*] https://www.github.com/AlphaZTX/physics2

1 Before you start

1.1 Legacy problems with physics package

The physics package provides `\qty` command for automatic-sizing braces. The `\qty` command would cause conflict with the siunitx package, which provides a unified method to typeset numbers and units correctly.

Besides, after you loaded physics, when you type `\homework` you will get Maxwell equations and Schrödinger equation. The `\homework` command is "declared" in `physics.sty` but it was not described in the documentation. That is, if you have defined `\homework` before loading physics package, physics would overwrite the definition "silently".

The vector-notation part of physics uses amsmath's (more exactly, `amsbsy.sty`'s) `\boldsymbol` command to generate bold vectors. Commands for cross/dot product are defined with `\boldsymbol`. `\boldsymbol` uses `\mathversion`, a LaTeX 2_ε kernel command that works well with traditional TFM-based fonts but fails when using unicode-math.

In the definition of `\imat`, `\xmat`, `\dmat` and `\admat` commands from physics, there is a `\newtoks` command which allocates a token list register and two `\newcount` commands allocating two count registers. Every time you write a command like `\imat` in your document, then one token list register and two count registers will be wasted. What's even worse is that, if you wrote really too many matrix commands from physics (for example, 32767 `\imat`s in LuaLaTeX), there'd be no room for a new `\count`.

physics integrated all the functions in one file (`physics.sty`), that is, you cannot load one of the total seven parts of functions; you have to load the seven parts altogether, even included the extra `\homework` command we mentioned in the first paragraph.

Moreover, the code of `physics.sty` "abuses" the `g`-type arguments of xparse package. Therefore the syntax of physics package looks kind of weird. See here for more.

1.2 Loading physics2

The physics2 package includes different modules, among which every module focuses on one single function.

Write the following line in the preamble to load physics2:

```
\usepackage{physics2}
```

But this is not enough. physics2 contains different modules, among which, only the common module would be loaded automatically by the package. If you want to load other modules of physics2, write this after loading physics2 package:

```
\usephysicsmodule{⟨module list⟩}
```

For example, "`\usephysicsmodule{ab,doubleprod}`" loads the ab module and the doubleprod module.

You can also load a module with options:

```
\usephysicsmodule[⟨option list⟩]{⟨module⟩}
```

For example, "`\usephysicsmodule[legacy]{ab}`" loads ab with the option "`legacy`".

Attention, if you used any font package in your document, remember that physics2 requires to be loaded *after* font packages.

2 List of commands

2.1 Automatic bracing

As mentioned in §1.1, the `\qty` command from physics would cause conflicts with siunitx. The command for automatic braces in physics2 is `\ab`, a shorthand for automatic **b**races.

The `\ab` command requires the ab module, so don't forget to write `\usephysicsmodule{ab}` in the preamble after you loaded physics2. Always remember, *do not put an* `\ab` *separately in the end of a math formula.* Take some examples:

[2.1.1]
```
\[ \ab ( \frac12 )  \quad
   \ab [ \frac12 ]  \quad
   \ab\{ \frac12 \} \]
```

$$\left(\frac12\right) \quad \left[\frac12\right] \quad \left\{\frac12\right\}$$

`\ab` can modify a delimiter-braced subformula. But the delimiters should not be out of the range described by the following chart:

```
 (,   )
 [,   ]
\{,  \}    or    \lbrace,  \rbrace
 <,   >    or    \langle,  \rangle
 |,   |    or      \vert,  \vert
\|,  \|    or      \Vert,  \Vert
```

For example, `$\ab{foo}$` and `$\ab(foo]$` are illegal, but `$\ab\{foo\}$` and `$\ab(foo)$` are okay; `$\ab([)$` is okay but `$\ab(()$` is illegal.

Attention, if you want to delimit a subformula with "{" and "}", you can only write `\{`, `\}` or `\lbrace`, `\rbrace` around it. `{` and `}` are not supported in ab module.

Between `\ab` and the first delimiter can be a "biggg" command, that is, from `\big` to `\Bigg`. Actually, you can also write `\biggg` and `\Biggg` because physics2 defines these after you load it. For example,

[2.1.2]

```
\[ \ab\Big \| \frac12 \| \quad
   \ab\Bigg < \frac12 >  \quad
   \ab\Biggg| \frac12 |  \]
```

$$\Big\|\frac12\Big\| \quad \Bigg\langle\frac12\Bigg\rangle \quad \Bigg|\frac12\Bigg|$$

Between `\ab` and the first delimiter can also be a star (`*`), which means "use the default size of delimiters". But in this situation, you needn't use the `\ab` command at all.

The physics package provides commands like `\pqty`, `\bqty`. In the ab module of physics2, these commands have changed to `\pab`, `\bab`, etc. The following example shows all the `\`X`ab` commands in ab module:

[2.1.3]

```
\def\0{\frac12}
\[ \pab{\0} \quad \bab{\0}
            \quad \Bab{\0} \]
\[ \aab{\0} \quad \vab{\0}
            \quad \Vab{\0} \]
```

$$\left(\frac12\right) \quad \left[\frac12\right] \quad \left\{\frac12\right\}$$
$$\left\langle\frac12\right\rangle \quad \left|\frac12\right| \quad \left\|\frac12\right\|$$

`\`X`ab` can take an optional star and an optional `[`⟨*biggg*⟩`]` argument. For example,

[2.1.4]

```
\def\0{\frac12}
\[ \pab[Big]{\0} \quad \bab*{\0} \]
```

$$\Big(\frac12\Big) \quad [\frac12]$$

physics also provides the following commands:

```
\abs    \norm    \eval    \order      \comm    \acomm    \pb
```

These commands are not originally supported by physics2, but the first four commands can be used through the ab.legacy module of physics2:

```
\usephysicsmodule{ab.legacy}
```

For example,

[2.1.5]

```
\def\0{\frac12}
\[ \abs{\0} \quad \abs[big]{\0}
            \quad \abs*{\0}     \]
```

$$\left|\frac12\right| \quad \big|\frac12\big| \quad |\frac12|$$

Users of the legacy physics package should notice that the syntax of `\eval` has been changed. The ab.legacy module abandoned the `\eval(foo|`-like syntax. The new `\eval`'s syntax is just like other commands in this module. There are also two variants of `\eval` — `\peval` and `\beval`. For example,

[2.1.6]

```
\def\0{1+\frac12x}
\[ \eval{\0}_a^b      \quad
   \peval*{\0}_a^b    \quad
   \beval[big]{\0}_a^b   \]
```

$$1+\frac{1}{2}x\Bigg|_a^b \quad (1+\frac{1}{2}x|_a^b \quad \big[1+\frac{1}{2}x\big|_a^b$$

The `\comm`, `\acomm` and `\pb` (Poisson bracket) are not supported. But you can write like `\ab[foo,baz]` or `\bab{foo,baz}` instead.

By the way, you can set the "order" symbol in ab.legacy through the `order` option like this:

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

Then `$\order(N)$` yields $O(N)$.

2.2 Vector notation

Unfortunately, there is not a plan for physics2 to support this part of physics completely, but the rest of this section will show some methods to maintain the document written with physics.

The `\vb(*)`, `\va(*)` and `\vu(*)` are not supported in any module of physics2. But these commands can be defined by copying the following lines below and pasting them in the preamble:

```
\makeatletter
\newcommand\vb{\@ifstar\boldsymbol\mathbf}
\newcommand\va[1]{\@ifstar{\vec{#1}}{\vec{\mathrm{#1}}}}
\newcommand\vu[1]{%
  \@ifstar{\hat{\boldsymbol{#1}}}{\hat{\mathbf{#1}}}}
\makeatother
```

The `\boldsymbol` command requires the amsmath or bm package. If you prefer to use bm, you can also use the `\bm` command. What's more, if you tried the commands above, you might find that, the result of `\va` above is different from that of physics. This is beacuse, if you choose to present a vector in bold, it's alomost no need to put a `\vec` ($\vec{\ }$) sign above it.

However, the method above may not work well with unicode-math because there are so many OpenType math fonts without a bold version. When using unicode-math, it's recommended to use `\symbf` and `\symbfit` for a separate vector. For example, `$\symbf{0}$` yields $\mathbf{0}$, and `$\symbfit{A}$` yields $\boldsymbol{A}$.

The `\vdot` and `\cross` commands are not supported in any module of physics2. Actually, there is no need to use a bold "$\cdot$" or "$\times$" for the products of two vectors. Using `\cdot` and `\times` is enough.

The commands related to "∇" are supported through nabla.legacy module. These commands are `\grad`, `\div` and `\curl`. These commands should not be put in the end of a math formula either (just like `\ab`). Notice that the former `\div` command for a "$\div$" (if there exists one) is redefined as `\divsymbol`. For example,

[2.2.1]

```
% \usephysicsmodule{nabla.legacy}
\[ \grad F       \quad
   \grad(\frac G2)     \]
\[ \div\Bigg[X]   \quad
    \curl*\{\frac Y2\} \]
\[ 2 \divsymbol 1      \]
```

$$\nabla F \quad \nabla\left(\frac{G}{2}\right)$$

$$\nabla\cdot\Bigg[X\Bigg] \quad \nabla\times\{\frac{Y}{2}\}$$

$$2 \div 1$$

Actually, the nabla-related commands end with `\ab`. Thus, the subformula after these commands can be delimited with `()`, `[]` and `\{\}`.

The nabla.legacy requires the fixdif package at least version 2.0 (file date: 2023/01/31 or after 2023/01/31).

By the way, if you are used to write `\bm` for a vector but interested in unicode-math, the bm-um.legacy module would be a passable alternative to bm package. Notice that the `\bm` command from the bm-um.legacy module can only take *one* letter (or *one* digit) as its argument.

2.3 Operators

There's no plan for physics2 to support this part of physics completely. The syntax in this part of physics (such as `\tan[2](x)`) abuses xparse.

It's suggested to write like this if you used the ab module:

```
$ \sin^2 \ab( \frac{\alpha}{2} ) $
```

Rather than take the superscript as an optional argument of the command of log-like functions.

The physics package provides a bundle of commands for log-like functions that have not been defined in the LaTeX 2_ε kernel. Those log-like functions can be used with the op.legacy module; this module do not support the syntax of physics either. For example:

[2.3.1]
```
% \usephysicsmodule{op.legacy}
\[ \asin^2 x \quad \rank \{ A \} \]
```

$$\operatorname{asin}^2 x \quad \operatorname{rank}\{A\}$$

The `\Re` and `\Im` commands are redefined as operators "Re" and "Im", while $\Re$ and $\Im$ are reserved as `\Resymbol` and `\Imsymbol`. $\Re$ and $\Im$ are ordinary symbols but Re and Im are opetators.

2.4 Quick quad text

The qtext.legacy module provides the `\q`⟨*foo*⟩ commands for `\quad`-wrapped texts. These commands have the same syntax as physics. For example,

[2.4.1]
```
% \usephysicsmodule{qtext.legacy}
\[ A \qq {foo bar} B \]
\[ A \qq*{foo bar} B \]
\[ C \qcc D \qcc*  E \]
\[ F \qif G \qthen H \]
```

$$A \quad \text{foo bar} \quad B$$
$$A\text{foo bar} \quad B$$
$$C \quad \text{c.c} \quad D\text{c.c} \quad E$$
$$F \quad \text{if} \quad G \quad \text{then} \quad H$$

All the commands described in §2.4 of physics documentation are supported when using qtext.legacy module, but I don't recommend to use this module unless you are maintaining a document written with physics's `\q`⟨*foo*⟩ commands.

2.5 Derivatives

There is no plan for physics2 to support this part of physics. If you want to typeset the differential operators on a better sense, you can try the fixdif package; if you want an easy way to type derivatives, you can try the derivative package. These two packages can be used together. For example,

[2.5.1]
```
% \usepackage{fixdif,derivative}
\[ \pdv{f}{x,y,z} \d x \]
Math ($\d x$) v.s.\ Text (\d x)
```

$$\frac{\partial^3 f}{\partial x\,\partial y\,\partial z}\,\mathrm{d}x$$

Math ($\mathrm{d}x$) v.s. Text (x̣)

Here are the documentations of fixdif and derivative.

fixdif's commands behave better in superscripts and subscripts.

2.6 Dirac bra-ket notation

There are two solutions to Dirac bra-ket in physics2 — ab.braket and braket. These two modules are *not* compatible and neither of them supports physics's syntax completely. Click here to see the ab.braket module and here to see the braket module.

The ab.braket module 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,

`\bra <` ⟨*subformula*⟩ `|`

For example,

[2.6.1]

```
\[ \bra < \frac \phi 2 | \]
\[ \bra*< \frac \phi 2 | \]
\[ \bra\Big< \phi  |     \]
```

$$\left\langle\frac{\phi}{2}\right|$$

$$\langle\frac{\phi}{2}|$$

$$\Big\langle\phi\Big|$$

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

`\ket |` ⟨*subformula*⟩ `>`

For example,

[2.6.2]

```
\[ \ket | \frac \psi 2 > \]
\[ \ket*| \frac \psi 2 > \]
\[ \ket\Big| \psi >      \]
```

$$\left|\frac{\psi}{2}\right\rangle$$

$$|\frac{\psi}{2}\rangle$$

$$\Big|\psi\Big\rangle$$

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,

`\braket <` ⟨*subformula*⟩ `>`

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

[2.6.3]
```
\[ \braket< \phi >           \]
\[ \braket< \phi | \psi >     \]
\[ \braket< \phi | A | \psi > \]
```

$$\langle\phi\rangle$$
$$\langle\phi|\psi\rangle$$
$$\langle\phi|A|\psi\rangle$$

[2.6.4]
```
\def\0{\frac\phi2}
\[ \braket     < \0 | \psi >  \]
\[ \braket*    < \0 | \psi >  \]
\[ \braket\Bigg< \0 | \psi >  \]
```

$$\left\langle\frac{\phi}{2}\middle|\psi\right\rangle$$
$$\langle\frac{\phi}{2}|\psi\rangle$$
$$\Bigg\langle\frac{\phi}{2}\Bigg|\psi\Bigg\rangle$$

The argument of `\ketbra` should be delimited with `|` and `|`. In the argument, `>` and `<` will be regarded as extensible $\rangle$ and $\langle$. that is,

`\ketbra |` $\langle\textit{subformula}_1\rangle$ `>` $\langle\textit{optional}\rangle$ `<` $\langle\textit{subformula}_2\rangle$ `|`

For example,

[2.6.5]
```
\def\0{\frac\phi2}
\[ \ketbra     | \0 >< \psi | \]
\[ \ketbra*    | \0 >< \psi | \]
\[ \ketbra\Bigg| \0 >< \psi | \]
```

$$\left|\frac{\phi}{2}\right\rangle\left\langle\psi\right|$$
$$|\frac{\phi}{2}\rangle\langle\psi|$$
$$\Bigg|\frac{\phi}{2}\Bigg\rangle\Bigg\langle\psi\Bigg|$$

[2.6.6]
```
\def\0{\frac\phi2}
\[ \ketbra| \0 >_x^y < \psi | \]
```

$$\left|\frac{\phi}{2}\right\rangle_x^y\left\langle\psi\right|$$

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.

The braket module 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:

`big`, `Big`, `bigg`, `Bigg`, `biggg` or `Biggg`.

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,

[2.6.7]

```
\def\0{\frac\phi2}
\[ \bra {\0} \quad \bra* {\0}
             \quad \bra[Big] {\0} \]
\[ \ket {\0} \quad \ket* {\0}
             \quad \ket[Big] {\0} \]
```

$$\left\langle\frac{\phi}{2}\right| \quad \langle\frac{\phi}{2}| \quad \Big\langle\frac{\phi}{2}\Big|$$
$$\left|\frac{\phi}{2}\right\rangle \quad |\frac{\phi}{2}\rangle \quad \Big|\frac{\phi}{2}\Big\rangle$$

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

[2.6.8]

```
\def\0{\frac\phi2}
\[ \braket {\0} {\psi}   \quad
   \braket*{\0} {\psi}   \quad
   \braket[big] {\0} {\psi} \]
```

$$\left\langle\frac{\phi}{2}\middle|\psi\right\rangle \quad \langle\frac{\phi}{2}|\psi\rangle \quad \big\langle\frac{\phi}{2}\big|\psi\big\rangle$$

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

[2.6.9]

```
\def\0{\frac\phi2}
\[ \braket [1] {\0} \quad
   \braket*[1] {\0} \]
\[ \braket [3] {\0}{A}{\psi}    \]
\[ \braket[3,big] {\0}{A}{\psi}
     \quad
   \braket[Big,3] {\0}{A}{\psi} \]
```

$$\left\langle\frac{\phi}{2}\right\rangle \quad \langle\frac{\phi}{2}\rangle$$
$$\left\langle\frac{\phi}{2}\middle|A\middle|\psi\right\rangle$$
$$\big\langle\frac{\phi}{2}\big|A\big|\psi\big\rangle \quad \Big\langle\frac{\phi}{2}\Big|A\Big|\psi\Big\rangle$$

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 $\langle$:

[2.6.10]

```
\def\0{\frac\phi2}
\[ \ketbra  {\0} {\psi}    \quad
   \ketbra* {\0} {\psi}       \]
\[ \ketbra [Bigg] {\0} {\psi} \]
\[ \ketbra {\0} [_x^y] {\psi} \]
```

$$\left|\frac{\phi}{2}\right\rangle\left\langle\psi\right| \quad |\frac{\phi}{2}\rangle\langle\psi|$$
$$\Bigg|\frac{\phi}{2}\Bigg\rangle\Bigg\langle\psi\Bigg|$$
$$\left|\frac{\phi}{2}\right\rangle_x^y\left\langle\psi\right|$$

2.7 Matrix macros

Unfortunately, physics2 do not support the `\mqty` command from physics. If you are used to this command, you can write like this:

```
\newcommand\mqty[1]{\begin{matrix}#1\end{matrix}}
\newcommand\pmqty[1]{\begin{pmatrix}#1\end{pmatrix}}
$\ab(\mqty{foo})$  or  $\pmqty{foo}$
```

These are equal to physics's `\mqty(foo)` (require amsmath).

physics2's diagmat module provides `\diagmat` command for diagonal matrices. For example,

[2.7.1]

```
\[
  \diagmat { 1, 2, 3 }
\]
```

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{matrix}$$

[2.7.2]

```
\[
  \pdiagmat [ empty = {} ]
    { a, b, c, d }
\]
```

$$\begin{pmatrix} a & & & \\ & b & & \\ & & c & \\ & & & d \end{pmatrix}$$

`\pdiagmat`, `\bdiagmat`, `\Bdiagmat`, `\vdiagmat` and `\Vdiagmat` are also available.

physics2's xmat module provides `\xmat` command for matrices with formatted entries. For example,

[2.7.3]

```
\[
  \xmat{a}{2}{3}
\]
```

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{matrix}$$

[2.7.4]

```
% \usephysicsmodule
%   [showleft=3,showtop=3]{xmat}
\[
  \pxmat{X}{m}{n}
\]
```

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & \cdots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \cdots & X_{2n} \\ X_{31} & X_{32} & X_{33} & \cdots & X_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & X_{m3} & \cdots & X_{mn} \end{pmatrix}$$

[2.7.5]

```
\[
  \xmat [showleft=2,showtop=2,
    format=\texttt{#1[#2][#3]}]
    {x}{m}{n}
\]
```

$$\begin{matrix} \texttt{x[1][1]} & \texttt{x[1][2]} & \cdots & \texttt{x[1][n]} \\ \texttt{x[2][1]} & \texttt{x[2][2]} & \cdots & \texttt{x[2][n]} \\ \vdots & \vdots & \ddots & \vdots \\ \texttt{x[m][1]} & \texttt{x[m][2]} & \cdots & \texttt{x[m][n]} \end{matrix}$$

`\pxmat`, `\bxmat`, `\Bxmat`, `\vxmat` and `\Vxmat` are also available.