

# The physics3 package\*

Mingyu Xia<sup>†</sup>

Released 2026-01-26 v0.1A

## Abstract

This is the document for physics3 package, which defines commands for typesetting math formulae faster and more simply. physics3 is a modularized package, each module provides its own function.

If you are a user of the legacy physics package, you can click [here](#) to see the section for physics users before you start. If you never used physics package before, just read *this* documentation.

The upstream maintainer of the similar package is Tingxuan Zhang<sup>1</sup>. This package is based on his physics2 package, then added some high-level macros and optimized the packing workflow.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Packages required . . . . .	3
1.2	Loading physics3 and its modules . . . . .	3
<b>2</b>	<b>Modules of physics3</b>	<b>4</b>
2.1	Features of the bare physics3 package . . . . .	4
2.2	The ab module – automatic braces . . . . .	4
2.3	The ab.braket and braket module – Dirac bra-ket notation . . . . .	6
2.4	The doubleprod module – tensors’ double product operator . . . . .	9
2.5	The diagmat module – simple diagonal matrices . . . . .	9
2.6	The xmat module – matrices with formatted entries . . . . .	10

---

\*<https://github.com/myhsia/physics3>, <https://ctan.org/pkg/physics3>

<sup>†</sup>[xiamingyu@westlake.edu.cn](mailto:xiamingyu@westlake.edu.cn)

<sup>1</sup>[alphaztx@163.com](mailto:alphaztx@163.com)

2.7	The operator module . . . . .	11
2.8	The bm-um module . . . . .	11
2.9	The ab.legacy module . . . . .	12
2.10	The qtext.legacy module . . . . .	12
<b>3</b>	<b>For the legacy physics users</b>	<b>13</b>
3.1	Legacy problems with physics package . . . . .	13
3.2	Loading physics3 . . . . .	14
3.3	List of commands . . . . .	14
<b>4</b>	<b>Code Implementation</b>	<b>18</b>
4.1	The physics3 package . . . . .	18
4.2	The ab module . . . . .	22
4.3	The ab.braket module . . . . .	29
4.4	The braket module . . . . .	33
4.5	The doubleprod module . . . . .	38
4.6	The diagmat module . . . . .	39
4.7	The xmat module . . . . .	41
4.8	The operator module . . . . .	45
4.9	The bm-um module . . . . .	46
4.10	The ab.legacy module . . . . .	47
4.11	The qtext.legacy module . . . . .	48
	<b>Index</b>	<b>49</b>

# 1 Introduction

The physics3 package aims to provide a bundle of commands for typesetting math faster in different modules. The commands provided by physics3 and its different modules are designed to be short and easy to memorize.

## 1.1 Packages required

The physics3 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 physics3 might require different packages. It will be explained in the following sections that which module requires which package.

The physics3 package requires  $\LaTeX 2_{\epsilon}$  kernel released after 2020/10. Please make sure that your  $\LaTeX$  distribution is not too old.

## 1.2 Loading physics3 and its modules

Just like loading any package, write

```
\usepackage{physics3}
```

in the preamble to load the physics3 package. In the current version, physics3 doesn't provide a package option.

physics3 itself doesn't provide many features. You need to load different modules of physics3 by using the `\usephxmodu` command only *after* loading this package in the preamble to have different features applied to your document.

---

```
\usephxmodu <options> {<module>}
```

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

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

The options of a physics3 module can be a comma-separated key-value list. These two lines load the `ab` module with option `tightbraces = true` and load `ab.braket` and `doubleprod` modules.


The following section introduce all the user-level modules of physics3. The modules have similar names like `<module>.legacy` 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.



<pre> \$ \ab ( \frac{1}{2} ) \$ \quad \$ \ab [ \frac{1}{2} ] \$ \quad \$ \ab\{ \frac{1}{2} \} \$ </pre>	$\left(\frac{1}{2}\right) \quad \left[\frac{1}{2}\right] \quad \left\{\frac{1}{2}\right\}$
---	--

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,

<pre> \$ \ab &lt;\frac{1}{2}&gt; \$ \quad \$ \ab\biggg \frac{1}{2}  \$ \quad \$ \ab* \ \frac{1}{2}\  \$ </pre>	$\left\langle\frac{1}{2}\right\rangle \quad \left \frac{1}{2}\right  \quad \left\ \frac{1}{2}\right\ $
--	--

 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.

**TeXhackers note:** The `ab` module uses “document commands” module of  $\text{\LaTeX} 2_{\epsilon}$  kernel (source file: `ltxcmd.dtx`). This  $\text{\LaTeX} 2_{\epsilon}$  kernel module provides a document-level command parser. `\ab` is a complex encapsulation of some internal document-level commands. Take an example, if you define a document-level command like this:

```
\NewDocumentCommand \foo { r() } {:#1::}
```

You can write `\foo(bar)` legally, but `\foo()` will be regarded illegal when you write another document-level command or end the paragraph. Similarly, things like `\ab()` will also cause errors.

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

<pre> \def\0{\frac{1}{2}} \$ \pab{\0} \$ \$ \bab{\0} \$ \$ \Bab{\0} \$ \$ \aab{\0} \$ \$ \vab{\0} \$ \$ \Vab{\0} \$ </pre>	$\left(\frac{1}{2}\right) \quad \left[\frac{1}{2}\right] \quad \left\{\frac{1}{2}\right\} \quad \left\langle\frac{1}{2}\right\rangle \quad \left \frac{1}{2}\right  \quad \left\ \frac{1}{2}\right\ $
--	---

These `\Xab` commands can take an optional star and an optional [`biggg`] argument. Star stands for using the default sizes. For example,

<pre> \def\0{n+\frac{1}{2}} \$ \pab[Big]{\0} \$ \quad \$ \bab*{\0} \$ </pre>	$\left(n + \frac{1}{2}\right) \quad \left[n + \frac{1}{2}\right]$
--	---

`tightbraces = <true|false>`: Influences whether thin skips are reserved around the paired delimiters. It only works with the automatically sized delimiters.

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

This two modules both contain four basic commands

`\bra`, `\ket`, `\braket`, `\ketbra`

These commands in the two modules share the same goal, but differ in syntax.

`ab.braket` A star (\*) or a size command can follow these commands, which are similar to the syntaxes of `\ab` module. The size commands can take the control sequences:

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

`braket` A star (\*) or a square-bracket-delimited size option, the size option can take the following values:

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

The star in the two modules stands for “do not size the bra-ket automatically”. Four basic commands’ syntaxes are described as follows respectively. **Please notice that the two module are conflict with each other, so don’t use them together.** One can choose one module according to personal habit.

<code>\bra</code>	<code>\bra</code> <i>&lt; * or size command &gt;</i> <code>&lt;</code> <i>&lt;subformula&gt;</i> <code> </code>	--> <code>ab.braket</code> modu.
<code>\ket</code>	<code>\ket</code> <i>&lt; * or size command &gt;</i> <code> </code> <i>&lt;subformula&gt;</i> <code>&gt;</code>	--> <code>ab.braket</code> modu.
<code>\bra</code>	<code>\bra</code> <i>&lt;&lt;* or [<i>&lt;size option&gt;]&gt;&gt;</i> <code>{</code> <i>&lt;subformula&gt;</i> <code>}</code></i>	--> <code>braket</code> modu.
<code>\ket</code>	<code>\ket</code> <i>&lt;&lt;* or [<i>&lt;size option&gt;]&gt;&gt;</i> <code>{</code> <i>&lt;subformula&gt;</i> <code>}</code></i>	--> <code>braket</code> modu.

- In `ab.braket` module, the argument of `\bra` should be delimited with `<` and `|`, and the argument of `\ket` should be delimited with `|` and `>`.



If you want to write “`>`” and “`<`” for relations in the argument of `\bra` and `\ket`, you can write `\mathrel{>}` and `\mathrel{<}`.

- In `braket` module, `\bra` and `\ket` take one mandatory argument, which should be braced with `{` and `}`.

For example, in `ab.braket` module,

```

\def\0{\frac{\phi}{2}}
$ \bra<\0| $ \quad $ \bra*<\0|      $%
      \quad $ \bra\Big<\phi| $ \[1ex]
$ \ket|\0> $ \quad $ \ket*|\0>      $%
      \quad $ \ket\Big|\psi> $
```

$\langle \frac{\phi}{2}  $	$\langle \frac{\phi}{2}  $	$\langle \phi  $
$ \frac{\phi}{2}\rangle$	$ \frac{\phi}{2}\rangle$	$ \psi\rangle$

and in braket module

<pre>\def\0{\frac\phi2} \$ \bra \0 \$\quad \$ \bra* \0 \$%       \quad \$ \bra[Big] \psi \$\\[1ex] \$ \ket \0 \$\quad \$ \ket* \0 \$%       \quad \$ \ket[Big] \psi \$</pre>	$\left\langle \frac{\phi}{2} \right  \quad \left\langle \frac{\phi}{2} \right  \quad \langle \psi  $ $\left  \frac{\phi}{2} \right\rangle \quad \left  \frac{\phi}{2} \right\rangle \quad  \psi\rangle$
--	---

---

	<code>\braket</code>	<code>\braket &lt;* or size command&gt;</code>	<code>&lt; &lt;subformula&gt;</code>	<code>&gt;</code>	<code>--&gt; ab.braket modu.</code>
		<code>\braket &lt;* or [<i>size option</i>, <i>i</i>]&gt;</code>	<code>i*{&lt;subformula&gt;}</code>	<code>&gt;</code>	<code>--&gt; braket modu.</code>

- In `ab.braket` module, the argument of `\braket` should be delimited with `<` and `>`, every “|” will be regarded as an extensible vertical bar in the `<subformula>` argument.



If you want to write “>” and “<” for relations in the argument of `\braket` and `\ketbra` in this module, you can write `\>` and `\<`. It is quite different from `\mathrel{>}` or `\mathrel{<}` because in these commands’ argument, `>` and `<` will be redefined.

- In `braket` module, the `\braket` command, in default, can take two arguments. If you want `\braket` to take one or three arguments, you can specific the number of arguments by number `i` in the square bracket, separate from the size option with a comma.

For example, in `ab.braket` and `braket` module,

<pre>\def\0{\frac\phi2} \$ \braket &lt;\hat A&gt; \$\quad \$ \braket &lt;\0 \hat A \psi&gt; \$\quad \$ \braket\big &lt;\0 \psi&gt; \$\quad\[1ex] \$ \braket* &lt;\0 \psi&gt; \$\quad \$ \braket\Big &lt;\0 \psi&gt; \$\quad \$ \ab \{\braket&lt;\psi \hat A \psi&gt;\}  \$</pre>	$\langle \hat{A} \rangle \quad \left\langle \frac{\phi}{2} \right  \hat{A} \left  \psi \right\rangle \quad \left\langle \frac{\phi}{2} \right  \psi \rangle$ $\left\langle \frac{\phi}{2} \right  \psi \rangle \quad \left\langle \frac{\phi}{2} \right  \psi \rangle \quad \left  \langle \psi   \hat{A}   \psi \rangle \right $
--	---



Commands from `ab.braket` should not be placed barely in `\ab|<subformula>|` or errors will arise. So, adding braces like the last line in the example above.

<pre>\def\0{\frac\phi2} \$ \braket [1] {\hat A} \$\quad \$ \braket [3] {\0}{\hat A}{\psi} \$\quad \$ \braket[big] {\0}{\psi} \$\quad\[1ex] \$ \braket* {\0}{\psi} \$\quad \$ \braket[Big] {\0}{\psi} \$\quad \$ \ab \{\braket[3]{\psi}{\hat A}{\psi}\}  \$</pre>	$\langle \hat{A} \rangle \quad \left\langle \frac{\phi}{2} \right  \hat{A} \left  \psi \right\rangle \quad \left\langle \frac{\phi}{2} \right  \psi \rangle$ $\left\langle \frac{\phi}{2} \right  \psi \rangle \quad \left\langle \frac{\phi}{2} \right  \psi \rangle \quad \left  \langle \psi   \hat{A}   \psi \rangle \right $
--	---

<hr/> <code>\ketbra</code>	<code>\ketbra</code>	<code>\langle * or size command \rangle</code>	<code>  \langle subformula_1 \rangle &gt;</code>	
		<code>\langle optional formula \rangle</code>	<code>&lt; \langle subformula_2 \rangle  </code>	<code>--&gt; ab.braket modu.</code>
	<code>\ketbra</code>	<code>\langle * or [size option] \rangle</code>	<code>\{ \langle subformula_1 \rangle \}</code>	
		<code>\langle optional formula \rangle</code>	<code>\{ \langle subformula_2 \rangle \}</code>	<code>--&gt; braket modu.</code>

- In `ab.braket` module, the argument of `\ketbra` should be delimited with `|` and `|, >` and `<` will be regarded as extensible `\rangle` and `\langle` in the argument.
- In `braket` module, the `\ketbra` command takes two mandatory arguments by default, and one optional argument between the two mandatory arguments `\rangle` and `\langle`.

For example, in `ab.braket` and `braket` module,

<pre>\def\0{\frac\phi2} \$ \ketbra        \0 &gt;&lt; \psi   \$\quad \$ \ketbra*      \0 &gt;&lt; \psi   \$\quad \$ \ketbra\Big   \0 &gt;_x^y&lt; \psi   \$</pre>	$\left  \frac{\phi}{2} \right\rangle \langle \psi   \quad \left  \frac{\phi}{2} \right\rangle \langle \psi   \quad \left  \frac{\phi}{2} \right\rangle_x^y \langle \psi  $
<pre>\def\0{\frac\phi2} \$ \ketbra      {0} {\psi} \$\quad \$ \ketbra*    {0} {\psi} \$\quad \$ \ketbra [Big] {0} [_x^y] {\psi} \$</pre>	$\left  \frac{\phi}{2} \right\rangle \langle \psi   \quad \left  \frac{\phi}{2} \right\rangle \langle \psi   \quad \left  \frac{\phi}{2} \right\rangle_x^y \langle \psi  $

For convince, the two modules also contain three advanced commands

`\bknorm`, `\kbproj`, `\expval`

These commands have the same syntax in the two modules.

<hr/> <code>\bknorm</code>	<code>\bknorm</code>	<code>\langle * \rangle</code>	<code>[[size]]</code>	<code>\{basis\}</code>
<code>\kbproj</code>	<code>\kbproj</code>	<code>\langle * \rangle</code>	<code>[[size]]</code>	<code>\{basis\}</code>
<code>\expval</code>	<code>\expval</code>	<code>\langle * \rangle</code>	<code>[[basis, size]]</code>	<code>\{operator\}</code>

For example,

<pre>\def \0{\frac\alpha2} \edef\1{\uparrow, \downarrow} \$ \bknorm \0 \$\quad \$ \bknorm [big]\0 \$ \quad \$ \bknorm*\0 \$\[[2ex] \$ \kbproj \0 \$\quad \$ \kbproj [big]\0 \$ \quad \$ \kbproj*\0 \$\[[2ex] \$ \expval [\1, Big] {\hat A} \$\quad \$ \expval*[\psi, Big]\0 \$</pre>	$\left\langle \frac{\alpha}{2} \middle  \frac{\alpha}{2} \right\rangle \quad \left\langle \frac{\alpha}{2} \middle  \frac{\alpha}{2} \right\rangle \quad \left\langle \frac{\alpha}{2} \middle  \frac{\alpha}{2} \right\rangle$ $\left  \frac{\alpha}{2} \right\rangle \langle \frac{\alpha}{2}   \quad \left  \frac{\alpha}{2} \right\rangle \langle \frac{\alpha}{2}   \quad \left  \frac{\alpha}{2} \right\rangle \langle \frac{\alpha}{2}  $ $\langle \uparrow, \downarrow   \hat{A}   \uparrow, \downarrow \rangle \quad \langle \psi   \frac{\alpha}{2}   \psi \rangle$
--	---



## 2.4 The doubleprod module – tensors’ double product operator

This module provides the `\doublecross` and `\doubledot` commands, which are regarded as binary operators by  $\TeX$ . Take an example of this module:

<code>\$ A \doublecross B \doubledot C \$</code>
--

$A \times B : C$
------------------

**crossscale**, **dotscale** =  $\langle fp \ num \rangle$ : Scales of “ $\times$ ” and “ $\cdot$ ” (Default: .8 and 1).

**crossopenup**, **dotopenup** =  $\langle fp \ num \rangle$ : Spaces between “ $\times$ ”s and “ $\cdot$ ”s =  $\langle fp \ num \rangle * \text{font size}$  (Default: .02 and .2).

**doubledot**, **crosssymbol** =  $\langle symbol \rangle$ : Pieces of `\doublecross` and `\doubledot` (Default: `\times`, `\ldotp`).

Their default values can be configured in module option. For example,

```
\usephxmodu[crossscale = .75, dotscale = 1.2,
             crossopenup = .05, dotopenup = .25]{doubleprod}
```

## 2.5 The diagmat module – simple diagonal matrices

This module requires the `mathtools` package and provides six `\*diagmat` commands:

<code>\diagmat</code>	$\langle \text{delimiter type} \rangle \text{diagmat}(\ast) [\langle \text{options} \rangle] \{ \langle \text{diag} \rangle \}$
<code>\pdiagmat</code>	where $\langle \text{diag} \rangle$ is the diagonal of the diagonal matrix. The entries should be separated
<code>\bdiagmat</code>	by commas. Prefixes like p, b, V have the same meanings as <code>pmatrix</code> , <code>bmatrix</code> and
<code>\Bdiagmat</code>	<code>Vmatrix</code> in the <code>mathtools</code> package, a star ( $\ast$ ) follows the <code>\*diagmat</code> commands corre-
<code>\vdiagmat</code>	spond to <code>\smallmatrix</code> in the <code>mathtools</code> package.
<code>\Vdiagmat</code>	

**empty** =  $\langle \text{token list} \rangle$ : Value of `\diagmat`’s empty entries (Default is blank).

**align** =  $\langle l | c | r \rangle$ : Align of `\diagmat`’s diagonal entries (Default: r).

Their default values can be configured in the module option like this:

```
\usephxmodu[empty = \cdot, align = r]{diagmat}
```

For example,

<code>\$ \pdiagmat [empty = \mathbf{0}] {</code> <code>\diagmat*[align = r]{1, \sqrt[3]{4}},</code> <code>\diagmat*{a, b} } \$,</code> <code>\$ \bdiagmat* {1, -1, -1, -1} \$</code>
---

$\left( \begin{array}{cc} 1 & \mathbf{0} \\ \mathbf{0} & \sqrt[3]{4} \end{array} \right), \left[ \begin{array}{ccc} 1 & -1 & -1 \\ -1 & -1 & -1 \end{array} \right]$
--

## 2.6 The xmat module – matrices with formatted entries

This module requires the mathtools package and provides six `\*xmat` commands:

<code>\xmat</code>	<code>\langle delimiter type \rangle xmat \langle * \rangle [ \langle options \rangle ] { \langle entry \rangle } { \langle rows shown \rangle } { \langle cols shown \rangle }</code>
<code>\pxmat</code>	Prefixes like p, b, V have the same meanings as pmatrix, bmatrix and Vmatrix
<code>\bxmat</code>	in the mathtools package, a star (*) follows the <code>\*xmat</code> commands correspond to
<code>\Bxmat</code>	<code>*smallmatrix</code> in the mathtools package. For example,
<code>\vxmat</code>	
<code>\Vxmat</code>	

If `\langle rows shown \rangle` or `\langle cols shown \rangle` contains non-digit characters, extra dots will be added. For example,

<code>\$ \bxmat*[showtop = 1, showleft = 2] {X}{m}{n} \$</code>	$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}$
---	---

`showtop`, `showleft` = `\langle int num \rangle`: Numbers of rows and columns to be shown at the top and the left side (Default: `MaxMatrixCols - 2`).

Their default values can be configured in the module option like this:

```
\usephxmodu[showtop = 1, showleft = 2]{xmat}
```

This will also influence “`\xmat`”s with digital `\langle rows shown \rangle` and `\langle cols shown \rangle` they are larger than the values corresponding to `showtop` and `showleft`. For example,

<code>\$ \pxmat*[showtop = 1, showleft = 2]{A}{8}{8} \$</code>	$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{18} \\ \vdots & \vdots & \ddots & \vdots \\ A_{81} & A_{82} & \cdots & A_{88} \end{pmatrix}$
--	---

The `format` option allows users to use a new entry format.

`format` = `\langle custom entry format in terms of #1, #2, and #3 \rangle`:

`#1`: The common entry, or the first mandatory `\langle entry \rangle` argument of `\xmat`;

`#2`: The row index;

`#3`: The column index.

This option should be only used in the optional argument of `\xmat`. For example,

<code>\$ \Bxmat*[showtop = 2, showleft = 1, format = \texttt{\#1[\#2][\#3]}] {x}{m}{n} \$</code>	$\left\{ \begin{array}{l} x[1][1] \cdots x[1][n] \\ x[2][1] \cdots x[2][n] \\ \vdots \\ x[m][1] \cdots x[m][n] \end{array} \right\}$
--	--

## 2.7 The operator module

This module provides a series of commands for log-like operators and some commands for nabla-related operators ( $\nabla$ )

```

\asin \acos \atan \acsc \asec \acot \rank \erf
\Tr \tr \Res \res \PV \pv \upe \iu
\Re \Im \identity \grad \curl \div \laplacian

```

- The first two lines of commands yield what they look like in math mode
- `\PV` yields “ $\mathcal{P}$ ” as an ordinary symbol and `\pv` yields “p.v.”.
- `\Re` and `\Im` are redefined as “Re” and “Im”.  $\Re$  and  $\Im$  are redefined as `\Resymbol` and `\Imsymbol`, in default.
- The “ $\div$ ” symbol was redefined as `\divsymbol`.

**ReIm** = `\langle true|false \rangle`: Determines whether to redefine `\Re` and `\Im`. If you want to reserve the definition of `\Re` and `\Im`, you can write like this:

```
\usephxmodu[ReIm = false]{operator}
```

For example,

<pre> % \usephxmodu{braket} \$ \asin x \$, \$ \PV f(z) = \pv f(z) \$\$\[[1ex] \$ \identity = \int d^2 \alpha   \kbproj\alpha/\pi \$\$\[[1ex] \$ \Re z = \Im z = \upe^{\iu\pi} = -1\$\$\[[1ex] \$ \grad V\$, \$ \div (x,y,z)\$, \$ \curl(x,y,z) \$ </pre>	$\operatorname{asin} x, \mathcal{P}f(z) = \text{p.v. } f(z)$ $\mathbb{1} = \int d^2\alpha  \alpha\rangle\langle\alpha /\pi$ $\operatorname{Re} z = \operatorname{Im} z = e^{i\pi} = -1$ $\nabla V, \nabla \cdot (x, y, z), \nabla \times (x, y, z)$
--	---

This module requires the `fixdif` package with file date 2023/01/31 at minimum.

## 2.8 The `bm-um` module

Similarly to the `bm` package, this module also provides a `\bm` command, but can take a *series of* math characters as its argument, which works for the `unicode-math` package.

The Latin letters or Greek letters in the argument will be switched 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,

<pre>\$ \bm {0 A z \alpha \Omega} \$</pre>	$\mathbf{0Az\alpha\Omega}$
--	----------------------------

## 2.9 The `ab.legacy` module

---

<code>\abs</code> <code>\norm</code> <code>\eval</code> ( <code>\peval</code> <code>\beval</code> ) <code>\order</code>	<code>\langle cmd \rangle * [ \langle biggg \rangle ] { \langle subformula \rangle } .</code>
--	---

---

This module provides the above commands. They share the same syntax as `Star` and `\langle biggg \rangle` are optional. `Star` stands for “use the default size”.

For example,

<pre>\def\0{1+\frac{1}{2}} \$ \abs\0 \$, \$ \norm[Big]\0 \$, \$ \order*\0 \$</pre>	$\left 1 + \frac{1}{2}\right , \left\ 1 + \frac{1}{2}\right\ , \mathcal{O}\left(1 + \frac{1}{2}\right)$
<pre>\def\0{1+\frac{x}{2}} \$ \eval      {\0}_a^b \$, \$ \peval*    {\0}_a^b \$, \$ \beval[big]{\0}_a^b \$</pre>	$1 + \frac{x}{2}\Big _a, \left(1 + \frac{x}{2}\right)_a^b, \left[1 + \frac{x}{2}\right]_a^b$

You can set the “order” symbol in this module through the `order` option like this:

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

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

## 2.10 The `qtext.legacy` module

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

<pre>% \usephxmodu{qtext.legacy} \$ A \qq {foo bar} B \$\[1ex] \$ A \qq*{foo bar} B \$\[1ex] \$ C \qcc D \qcc* E \$\[1ex] \$ F \qif G \qthen H \$</pre>	$A \quad \text{foo bar} \quad B$ $A\text{foo bar} \quad B$ $C \quad c.c \quad Dc.c \quad E$ $F \quad \text{if } G \quad \text{then } H$
---	--

All the commands described in §2.4 of [physics documentation](#) are supported when using `qtext.legacy` module, but I don’t recommend using this module unless you are maintaining a document written with `physics`’s `\q{foo}` commands.

### 3 For the legacy physics users

This section describes physics3 package for those who are used to the physics package, which 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.

In this section, the modules of physics3 will be introduced in the same order as the physics documentation.

#### 3.1 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  $\TeX 2_{\epsilon}$  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 `\imats` in Lua $\TeX$ ), 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.

## 3.2 Loading physics3

The physics3 package includes different modules, among which every module focuses on one single function.

Write the following line in the preamble to load physics3:

```
\usepackage{physics3}
```

But this is not enough. physics3 contains different modules. If you want to load any module of physics3, write this line after loading physics3 package:

```
\usephxmodu{<module list>}
```

For example, “\usephxmodu{ab, doubleprod}” loads the ab module and the doubleprod module.

You can also load a module with options:

```
\usephxmodu[<option list>]{<module>}
```

For example, “\usephxmodu[legacy]{ab}” loads ab with the option “legacy”.



Attention, if you used any font package in your document, remember that physics3 requires to be loaded *after* font packages.

## 3.3 List of commands

### 3.3.1 Automatic bracing

As mentioned in §3.1, the \qty command from physics will conflict with siunitx. The command for automatic braces in physics3 is \ab, short for **automatic braces**.

physics also provides the following commands:

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



These commands are not originally supported by physics3, but the first four commands can be used through the ab.legacy module of physics3:

```
\usephxmodu{ab.legacy}
```

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.

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:

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

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

### 3.3.2 Vector notation

Unfortunately, there is not a plan for `physics3` 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 `physics3`. 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 because, if you choose to present a vector in bold, there’s almost no need to put a `\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 **0**, and `\symbfit{A}` yields **A**.

The `\vdot` and `\cross` commands are not supported in any module of `physics3`. 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 “ $\nabla$ ” are supported through `operator` 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 “÷” (if there exists one) is redefined as `\divsymbol`.

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

The operator requires the `fixdif` package at least version 2.0 (file date on or after 2023/01/31).

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

### 3.3.3 Operators

There’s no plan for `physics3` 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 `ℒTeX2ε` kernel. They can be used with the `operator` module; this module does not support the syntax of `physics` either.

The `\Re` and `\Im` commands are redefined as operators “Re” and “Im”, while `ℜ` and `ℑ` are reserved as `\Resymbol` and `\Imsymbol`. `ℜ` and `ℑ` are ordinary symbols but `Re` and `Im` are operators.

### 3.3.4 Quick quad text

All the commands described in §2.4 of [physics documentation](#) are supported when using `qtext.legacy` module. Click here to see the `qtext.legacy` module.

### 3.3.5 Derivatives

There is no plan for `physics3` 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,



```
% \usepackage{fixdif, derivative}
$ \pdv{f}{x,y,z} \d x $\[1ex]
Math ($\d x$) v.s. \ Text (\d x)
```

$$\frac{\partial^3 f}{\partial x \partial y \partial z} dx$$

Math (dx) v.s. Text (x)

Here are the documentations of [fixdif](#) and [derivative](#). `fixdif`'s commands behave better in superscripts and subscripts.

### 3.3.6 Dirac bra-ket notation

There are two solutions to Dirac bra-ket in physics3 — `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.

### 3.3.7 Matrix macros

Unfortunately, physics3 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`).

physics3's `diagmat` module provides `\diagmat` command for diagonal matrices. `\pdiagmat`, `\bdiagmat`, `\Bdiagmat`, `\vdiagmat` and `\Vdiagmat` are also available.

physics3's `xmat` module provides `\xmat` command for matrices with formatted entries. `\pxmat`, `\bxmat`, `\Bxmat`, `\vxmat` and `\Vxmat` are also available.

## 4 Code Implementation

### 4.1 The physics3 package

```
1 <*package>
```

We use `phx` as the namespace for physics3 modules.

```
2 <@@=phx>
3 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
4 \def \phx@date {2026-01-23}
5 \def \phx@version {v0.1A}
6 \ProvidesExplPackage {physics3} {2026-01-26} {v0.1A}
7 {Tools for typesetting math for physics.}
```

#### 4.1.1 Common variables

```
\l__phx_tmpa_box \phx@temp<register type><a or b>
```

Some  $\LaTeX 2_{\epsilon}$  variables starting with “`\phx@temp`”. These variables can be shared by any module of physics3.

```
8 \box_new:N \l__phx_tmpa_box
```

*(End of definition for `\l__phx_tmpa_box`.)*

#### 4.1.2 Package requirements and module-loading methods

physics3 requires `keyval` (part of the graphics bundle) to process options of modules.

```
9 \RequirePackage{keyval}
10 \def\phx@true{true}
11 \def\phx@false{false}
```

```
\__phx_define_key:nnnn \phx@define@key {<module>} {<key>} [<default value>] {<code>}
\__phx_setkeys:nn \phx@setkeys {<module>} {<key-val list>}
\__phx_processkeyopt:n \phx@processkeyopt {<module>}
```

The position of `\phx@processkeyopt` in a physics3 module is just the same as the position of `\ProcessOptions` in a regular  $\LaTeX$  package.

```
12 \cs_new_nopar:Npn \__phx_define_key:nnnn #1#2#3#4
13 { \define@key {phx-#1} {#2} [#3] {#4} }
14 \cs_new_nopar:Npn \__phx_setkeys:nn #1#2
15 { \setkeys {phx-#1} {#2} }
16 \cs_new:Npn \__phx_processkeyopt:n #1
17 {
18 \let\reserved@a\@empty
19 \edef\reserved@a{\@optionlist{\@currname.\@currentt}}%
```

```

20 \edef\reserved@a{\noexpand\phx@setkeys{#1}{\reserved@a}}%
21 \reserved@a% the next line thanks to `geometry'
22 \AtEndOfPackage{\let\@unprocessedoptions\relax}
23 }
24 \cs_set_eq:NN \phx@define@key \__phx_define_key:nnnn
25 \cs_set_eq:NN \phx@setkeys \__phx_setkeys:nn
26 \cs_set_eq:NN \phx@processkeyopt \__phx_processkeyopt:n

```

(End of definition for `\__phx_define_key:nnnn`, `\__phx_setkeys:nn`, and `\__phx_processkeyopt:n`.)

We use almost the same way to load physics3 modules as  $\text{\LaTeX} 2_{\epsilon}$  kernel does. We use a lot of kernel commands in  $\text{\LaTeX} 2_{\epsilon}$ .

```

\usephxmodu \phx@requiremodule

```

`\usephxmodu` [*key-val options*]{*module*} [*key-val options*]  
`\phx@requiremodule` [*key-val options*]{*module*} [*key-val options*]  
`\usephxmodu` is a user command, and `\phx@requiremodule` is a developer command.

```

27 \def\usephxmodu{\phx@FWoptions\@pkgextension}
28 \let\phx@requiremodule\usephxmodu
29 \@onlypreamble\usephxmodu
30 \def\phx@FWoptions#1{\@ifnextchar [%]
31   {\phx@FWoptions#1}{\phx@FWoptions#1 []}}
32 \@onlypreamble\phx@FWoptions
33 \def\phx@FWoptions#1[#2]#3{\@ifnextchar [%]
34   {\phx@FWoptions#1[#2]#3}{\phx@FWoptions#1[#2]#3 []}}
35 \@onlypreamble\phx@FWoptions
36 \def\phx@FWoptions#1[#2]#3[#4]{%
37   \def\reserved@b##1,{%
38     \ifx\@nnil##1\relax\else
39       \ifx\@nnil##1\@nnil\else
40         \noexpand\@onefilewithoptions{phx-##1}{\unexpanded{#2}} [{#4}]%
41         \noexpand\@pkgextension
42       \fi
43       \expandafter\reserved@b
44     \fi}%
45   \edef\reserved@a{\zap@space#3~\@empty}%
46   \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nnil,}%
47   \reserved@a}
48 \@onlypreamble\phx@FWoptions

```

(End of definition for `\usephxmodu` and `\phx@requiremodule`. These functions are documented on page 3.)

### 4.1.3 The (used to be) common module

The code below used to be the automatically-loaded common module, but now we load it together with physics3's code. This change may bring better performance in Windows system.

Check if unicode-math loaded and (re)define the vert symbols. The `\relax`'s at the ends of `\vert` and `\Vert`'s definitions must not be removed. They are for `\ifx` to compare. `unicode-math` sets these symbols `\fam1`, `\symoperators` is equal to 1 in  $\text{\LaTeX} 2_{\epsilon}$  kernel. Moreover, we make `\mid` as a delimiter but it may not work.

```

49 \AtBeginDocument{\ifcsname symrm\endcsname
50   \protected\def|\{\Udelimiter 0 \symoperators "2016 }%
51   \protected\def\vert{\Udelimiter 0 \symoperators "007C\relax}%
52   \protected\def\Vert{\Udelimiter 0 \symoperators "2016\relax}%
53   \protected\def\mid{\Udelimiter 3 \symoperators "007C }%
54 \fi}
55 \protected\def\Vert{\delimiter"026B30D\relax}
56 \protected\def\mid{\delimiter"326A30C }
```

```

\delopen   \delopen <left delimiter>
\delclose  \delclose <right delimiter>
```

Actually in  $\text{\TeX}$ , `\left` and `\right` will enclose the subformula as “inner”, but `\delopen` and `\delclose` will make the subformula an empty open node and a non-empty close node.

```

57 \DeclareRobustCommand\delopen{\mathopen{}\mathclose\bgroup\left}
58 \DeclareRobustCommand\delclose{\aftergroup\egroup\right}
59 % Extension to 2e kernel's or amsmath's biggggg commands.
```

*(End of definition for `\delopen` and `\delclose`. These functions are documented on page 4.)*

`\bBigg@` is a command from `amsmath`. The code below should update with `amsmath` together.

```

60 \ifdefined\bBigg@
61   \DeclareRobustCommand\biggg{\bBigg@{3}}
62   \DeclareRobustCommand\Biggg{\bBigg@{3.5}}
63 \else
64   \DeclareRobustCommand\biggg[1]{\leavevmode@ifvmode
65     {\hbox{\$ \left#1 \vbox to 20.5\p@{\} \right. \n@space$}}}
66   \DeclareRobustCommand\Biggg[1]{\leavevmode@ifvmode
67     {\hbox{\$ \left#1 \vbox to 23.5\p@{\} \right. \n@space$}}}
68   \AtBeginDocument{\ifdefined\bBigg@
69     \DeclareRobustCommand\biggg{\bBigg@{3}}%
70     \DeclareRobustCommand\Biggg{\bBigg@{3.5}}%
```

```

71 \fi}
72 \fi
73 \DeclareRobustCommand\bigggl{\mathopen\biggg}
74 \DeclareRobustCommand\bigggm{\mathrel\biggg}
75 \DeclareRobustCommand\bigggr{\mathclose\biggg}
76 \DeclareRobustCommand\Bigggl{\mathopen\Biggg}
77 \DeclareRobustCommand\Bigggm{\mathrel\Biggg}
78 \DeclareRobustCommand\Bigggr{\mathclose\Biggg}

```

`\_phx_mathvphantom:n`

`\phx@mathvphantom {⟨math mode material⟩}`

This command is just like `\vphantom` in  $\TeX 2_{\epsilon}$  kernel but only works in math mode.

```

79 \cs_new:Npn \_phx_mathvphantom:n #1
80 {
81   \hbox_set:Nn \l__phx_tmpa_box {}
82   \tex_mathchoice:D
83   {
84     \hbox_set:Nn \l_tmpa_box { $\displaystyle#1$ }
85     \box_set_ht:Nn \l__phx_tmpa_box { \box_ht:N \l_tmpa_box }
86     \box_set_dp:Nn \l__phx_tmpa_box { \box_dp:N \l_tmpa_box }
87     \box_use_drop:N \l__phx_tmpa_box
88   }
89   {
90     \hbox_set:Nn \l_tmpa_box { $\textstyle#1$ }
91     \box_set_ht:Nn \l__phx_tmpa_box { \box_ht:N \l_tmpa_box }
92     \box_set_dp:Nn \l__phx_tmpa_box { \box_dp:N \l_tmpa_box }
93     \box_use_drop:N \l__phx_tmpa_box
94   }
95   {
96     \hbox_set:Nn \l_tmpa_box { $\scriptstyle#1$ }
97     \box_set_ht:Nn \l__phx_tmpa_box { \box_ht:N \l_tmpa_box }
98     \box_set_dp:Nn \l__phx_tmpa_box { \box_dp:N \l_tmpa_box }
99     \box_use_drop:N \l__phx_tmpa_box
100  }
101  {
102    \hbox_set:Nn \l_tmpa_box { $\scriptscriptstyle#1$ }
103    \box_set_ht:Nn \l__phx_tmpa_box { \box_ht:N \l_tmpa_box }
104    \box_set_dp:Nn \l__phx_tmpa_box { \box_dp:N \l_tmpa_box }
105    \box_use_drop:N \l__phx_tmpa_box
106  }
107 }
108 \cs_set_eq:NN \phx@mathvphantom \_phx_mathvphantom:n

```

(End of definition for `\_phx_mathvphantom:n`.)

#### 4.1.4 The (used to be) `explsetup` module

`\l__phx_tmpa_tl` Some common variables and functions for experimental L<sup>A</sup>T<sub>E</sub>X3 syntax.

```
\l__phx_tmpb_tl 109 \tl_new:N \l__phx_tmpa_tl
110 \tl_new:N \l__phx_tmpb_tl
```

(End of definition for `\l__phx_tmpa_tl` and `\l__phx_tmpb_tl`.)

The function that can gobble one token.

```
111 \cs_new:Npn \_phx_gobble_i:n #1 { }
112 \file_input_stop:
113 </package>
```

Temporarily disable the namespace.

```
114 <@@=>
```

## 4.2 The `ab` module

`<*gibberish>`

This module is important but the code is hard to read. One of the motivations I manage physics3 with DocStrip is that, when I tried to write a new module based on `ab` after 5 months when I maintained physics3 the last time, I found that I could not understand the code I wrote at all! Therefore, it's significant to comment out the alien code in `ab`.

`</gibberish>`

```
115 <*ab>
116 \ProvidesFile {phx-ab.sty}
117  [\phx@date\ `ab' (autobraces) module of physics3]
```

If you don't know when to use `\phx@define@key`, `\phx@setkeys` and `\phx@processkeyopt` in a module, see ahead. In `ab`, the `|tightbraces|` option can control if the automatically-sized braces are tight or not. Do you remember `\delopen` and `\delclose`?

```
118 \phx@define@key{ab}{tightbraces}{true}{\def\@phx@abtight{#1}}
119 \phx@setkeys{ab}{tightbraces=true}
120 \phx@processkeyopt{ab}
```

```

\phx@abopen      \phx@abopen <left delimiter>
\phx@abclose     \phx@abclose <right delimiter>

```

They are defined either `{\delopen, \delclose}` or `{\left, \right}`. If a module requires `ab`, these two commands are likely to be used.

```

121 \ifx\@phx@abtight\phx@true
122   \let\phx@abopen\delopen
123   \let\phx@abclose\delclose
124 \else
125   \let\phx@abopen\left
126   \let\phx@abclose\right
127 \fi

```

*(End of definition for \phx@abopen and \phx@abclose.)*

#### 4.2.1 The implementation of `\ab`

This is the alienest part of `ab`. It's better to draw something rather than write boring comments. First let's take a look at `\ab`'s syntax. After `\ab` should be a pair of delimiters; take `()` as an example. Between `\ab` and `"(` can be a `biggg` command or `star`, or even nothing. `\ab` is defined as follows:

```

\ab ← begindef
      \phx@d@lx {mb} {ab}
enddef

```

where `ab` is the branch name of `\ab()`, and `mb` is the branch name of `\ab\big()` and `\ab*()`. Then let's see the syntax of `\phx@d@lx`.

```

\phx@d@lx {<biggg or star branch name>}
          {<automatic branch name>} {#3}

```

Here exists an `#3`. `#3` is one token immediately following `\ab`, which can be `{a biggg command or a star }` or a `"(`, under our assumption.

`\phx@d@lx` is defined as follows:

```

\phx@d@lx ← begindef( #1: biggg or star branch name, <mb>; #2:
    automatic branch name, <ab>; #3, the token after \ab
)
    if#3 == biggg or #3 == star (⇔ csname {phx@del\string#3}
    is defined)then
        let <next cs> = csname {phx@d@lx<mb>}
    else
        let <next cs> = csname {phx@d@lx<ab>}
    endif
    <next cs> #3
enddef

```

The condition should be true when #3 is `\big` or `*`, and it should be false when #3 is `(`. Accordingly, in math mode,

$$\begin{aligned} \backslash ab \backslash big ( &\rightarrow \backslash phx@d@lxmb \backslash big ( \\ \backslash ab ( &\rightarrow \backslash phx@d@lxab ( \end{aligned}$$

Then we meet two new commands — `\phx@d@lxmb` and `\phx@d@lxab`. Syntax is as follows.

```

\phx@d@lxmb <biggg or *> <left delimiter>
    <subformula> <right delimiter>
\phx@d@lxab          <left delimiter>
    <subformula> <right delimiter>

```

Notice that `ab` and `mb` in the above commands are names of `\ab`'s two branches — they are like namespaces. `\phx@d@lxmb` and `\phx@d@lxab` are defined by the following two lines:

```

\phx@d@l@genxm{mb} \phx@d@l@genxa{ab}

```



`\phx@d@l@genxm` and `\phx@d@l@genxa` are defined as follows:

```

\phx@d@l@genxm ← begindef(#1: biggg or star branch name, <mb>)
    \phx@d@l<mb> ← begindef(##1: biggg or star;
        ##2: left delimiter)
        \begingroup
        if##1 == starthen
            <temp> ← \relax
        else
            <temp> ← ##1
        endif
        csname {phx@<mb>@\string##2}
            <temp> ##2
        % requires an \endgroup af-
        % ter the right delimiter
        enddef
    enddef

\phx@d@l@genxa ← begindef(#1: automatic branch name, <ab>)
    \phx@d@l<ab> ← begindef(##1: left delimiter)
        csname {phx@<ab>@\string##1}
            ##1
        enddef
    enddef

```

So we can get

```

\ab \big ( → \begingroup csname {phx@mb@{} \big (
\ab * ( → \begingroup csname {phx@mb@{} \relax (
\ab ( → csname {phx@ab@{} (

```

The csnames above (`\phx@mb@()` and `\phx@ab@()`) are generated with `\phx@AB@gen`.

```

\phx@AB@gen {<branch name>} <left delimiter>
    {<arg spec>} {<definition>}

```

If `<branch name>` is `mb`, `{<arg spec>}` should be `mr()`, where `m` is for `biggg` or `star`; If `<branch name>` is `ab`, `{<arg spec>}` should be `r()`.

**T<sub>E</sub>Xhackers note:** The “(” in the example above must not be replaced by a subformula braced by a pair of {}.

```

\phx@AB@gen      \phx@AB@gen {\langle branch name \rangle} \langle left delimiter \rangle
                  {\langle arg spec \rangle} {\langle definition \rangle}

128 \def\phx@AB@gen#1#2{\expandafter
129   \DeclareDocumentCommand\csname phx@#1@string#2\endcsname}
130   \phx@AB@gen{ab}{\langle r \rangle}%
131   {\phx@abopen{#1\phx@abclose}}
132   \phx@AB@gen{ab}[\langle r [] \rangle]%
133   {\phx@abopen[#1\phx@abclose]}
134   \phx@AB@gen{ab}\{\langle r \{ \} \rangle}%
135   {\phx@abopen\{#1\phx@abclose\}}
136   \phx@AB@gen{ab}|\langle r | | \rangle}%
137   {\phx@abopen|#1\phx@abclose|}
138   \phx@AB@gen{ab}\|\langle r \| \| \rangle}%
139   {\phx@abopen\|#1\phx@abclose\|}
140   \phx@AB@gen{ab}\langle \langle r \rangle \rangle}%
141   {\phx@abopen\langle #1\phx@abclose \rangle}
142   \phx@AB@gen{ab}\lbrace\langle r \lbrace \rbrace \rangle}%
143   {\phx@abopen\lbrace#1\phx@abclose\rbrace}
144   \phx@AB@gen{ab}\vert\langle r \vert \vert \rangle}%
145   {\phx@abopen\vert#1\phx@abclose\vert}
146   \phx@AB@gen{ab}\Vert\langle r \Vert \Vert \rangle}%
147   {\phx@abopen\Vert#1\phx@abclose\Vert}
148   \phx@AB@gen{ab}\langle \langle r \rangle \rangle}%
149   {\phx@abopen\langle #1\phx@abclose \rangle}

```

\endgroup’s in the end of the following definitions are corresponding to \begingroup’s in the definition of \phx@d@l@genxm.

```

150 \phx@AB@gen{mb}{\langle mr \rangle}%
151   {\mathopen#1(#2\mathclose#1)\endgroup}
152 \phx@AB@gen{mb}[\langle mr [] \rangle]%
153   {\mathopen#1[#2\mathclose#1]\endgroup}
154 \phx@AB@gen{mb}\{\langle mr \{ \} \rangle}%
155   {\mathopen#1\lbrace#2\mathclose#1\rbrace\endgroup}
156 \phx@AB@gen{mb}|\langle mr | | \rangle}%
157   {\mathopen#1\vert#2\mathclose#1\vert\endgroup}
158 \phx@AB@gen{mb}\|\langle mr \| \| \rangle}%
159   {\mathopen#1\Vert#2\mathclose#1\Vert\endgroup}
160 \phx@AB@gen{mb}\langle \langle mr \rangle \rangle}%

```

```

161   {\mathopen#1\langle#2\mathclose#1\rangle\endgroup}
162 \phx@AB@gen{mb}\lbrace{mr\lbrace\rbrace}%
163   {\mathopen#1\lbrace#2\mathclose#1\rbrace\endgroup}
164 \phx@AB@gen{mb}\vert{mr\vert\vert}%
165   {\mathopen#1\vert#2\mathclose#1\vert\endgroup}
166 \phx@AB@gen{mb}\Vert{mr\Vert\Vert}%
167   {\mathopen#1\Vert#2\mathclose#1\Vert\endgroup}
168 \phx@AB@gen{mb}\langle{mr\langle\rangle}%
169   {\mathopen#1\langle#2\mathclose#1\rangle\endgroup}

```

(End of definition for \phx@AB@gen.)

\phx@del The syntax seems not important. These following lines seems only for \ifcsname to judge if the commands are defined.

```

170 \def\phx@del#1#2#3{\phx@abopen#1#3\phx@abclose#2}
171 \expandafter\def\csname phx@del\string*\endcsname
172   #1#2#3{\mathopen#1#3\mathclose#2}
173 \expandafter\def\csname phx@del\string\big\endcsname
174   #1#2#3{\bigl#1#3\bigr#2}
175 \expandafter\def\csname phx@del\string\Big\endcsname
176   #1#2#3{\Bigl#1#3\Bigr#2}
177 \expandafter\def\csname phx@del\string\bigg\endcsname
178   #1#2#3{\biggl#1#3\biggr#2}
179 \expandafter\def\csname phx@del\string\Bigg\endcsname
180   #1#2#3{\Biggl#1#3\Biggr#2}
181 \expandafter\def\csname phx@del\string\biggg\endcsname
182   #1#2#3{\bigggl#1#3\bigggr#2}
183 \expandafter\def\csname phx@del\string\Biggg\endcsname
184   #1#2#3{\Bigggl#1#3\Bigggr#2}

```

(End of definition for \phx@del.)

```

\phx@d@lx   \phx@d@lx {\biggg or star branch name}} {\automatic branch name}}
            {#3}

```

```

185 \def\phx@d@lx#1#2#3{%
186   \ifcsname phx@del\string#3\endcsname
187     \def\reserved@a{#1}% #3 is star or \biggg
188   \else
189     \def\reserved@a{#2}% #3 is delimiter
190   \fi
191   \csname phx@d@lx\reserved@a\endcsname#3}

```

(End of definition for \phx@d@lx.)

```

\phx@d@l@genxm      \phx@d@l@genxm {(biggg or star branch name)}
\phx@d@l@genxa      \phx@d@l@genxa {(automatic branch name)}

192 \def\phx@d@l@genxm#1{%
193   \expandafter\def\csname phx@d@lx#1\endcsname##1##2{%
194     \begingroup % \endgroup is at the end of #4 of \phx@AB@gen
195     \ifx##1*\let\phx@tempa=\relax\else\let\phx@tempa=##1\fi
196     \csname phx@#1@\string##2\endcsname\phx@tempa##2}}
197 \def\phx@d@l@genxa#1{%
198   \expandafter\def\csname phx@d@lx#1\endcsname##1{%
199     \csname phx@#1@\string##1\endcsname##1}}

```

*(End of definition for \phx@d@l@genxm and \phx@d@l@genxa.)*

```

\phx@d@lxmb          \phx@d@lxmb (biggg or *) (left delimiter)
\phx@d@lxab          \phx@d@lxab (left delimiter)
                     \phx@d@lxab          (left delimiter)
                     (subformula) (right delimiter)
                     (subformula) (right delimiter)

200 \phx@d@l@genxm{mb}
201 \phx@d@l@genxa{ab}

```

*(End of definition for \phx@d@lxmb and \phx@d@lxab.)*

**\ab** The users' command \ab.

```
202 \DeclareRobustCommand\ab{\phx@d@lx{mb}{ab}}
```

*(End of definition for \ab. This function is documented on page 4.)*

#### 4.2.2 \pab-like commands

This is so simple. No need to comment a lot.

```

\phx@d@l@geny      \phx@d@l@geny (command) (left delimiter) (right delimiter)
                   This command used to define commands like \pab.

203 \def\phx@d@l@geny#1#2#3{%

#1 : star;          #2 : bigg (csname);          #3 : subformula.

204 \DeclareDocumentCommand#1{som}{
205   \IfBooleanTF{##1}%
206     {#2##3#3}%
207     {\IfValueTF{##2}%
208       {\csname##2l\endcsname#2##3\csname##2r\endcsname#3}%

```

```

209         {\phx@abopen#2##3\phx@abclose#3}%
210     }%
211 }%
212 }
213 \phx@d@l@geny\pab()
214 \phx@d@l@geny\bab[]
215 \phx@d@l@geny\Bab\lbrace\rbrace
216 \phx@d@l@geny\vab\vert\vert
217 \phx@d@l@geny\aab\langle\rangle
218 \phx@d@l@geny\Vab\Vert\Vert
219 \protect \endinput
220 </ab>

```

(End of definition for \phx@d@l@geny.)

### 4.3 The ab.braket module

```

221 <*ab.braket>
222 \ProvidesFile {phx-ab.braket.sty}
223   [\phx@date\ `ab.braket' module of physics3]

```

This module requires \phx@abopen and \phx@abclose from ab. This module may have conflict with braket.

```

224 \phx@requiremodule{ab}
225 \ifdefined\phx@bra@@
226   \PackageWarning{physics3}{You cannot load `ab.braket' and `braket'
227     modules together.\MessageBreak Only `ab.braket' module works now.}
228 \fi

```

**\bra**        \bra <  $\langle subformula \rangle$  |

(End of definition for \bra. This function is documented on page 6.)

```

229 \phx@AB@gen{br.m}<{mr<|}{\mathopen#1\langle#2\mathclose#1\vert\endgroup}
230 \phx@AB@gen{br.a}<{r<|}{\phx@abopen\langle#1\phx@abclose\vert}
231 \phx@d@l@genxm{br.m}
232 \phx@d@l@genxa{br.a}
233 \DeclareRobustCommand\bra{\phx@d@l{x}{br.m}{br.a}}

```

**\ket**        \ket |  $\langle subformula \rangle$  >

(End of definition for \ket. This function is documented on page 6.)

```

234 \phx@AB@gen{kt.m}|{mr|>}{\mathopen#1\vert#2\mathclose#1\rangle\endgroup}
235 \phx@AB@gen{kt.a}|{r|>}{\phx@abopen\vert#1\phx@abclose\rangle}

```

```

236 \phx@d@l@genxm{kt.m}
237 \phx@d@l@genxa{kt.a}
238 \DeclareRobustCommand\ket{\phx@d@lx{kt.m}{kt.a}}

```

**\braket**             $\langle \textit{subformula 1} \mid \langle \textit{subformula 2} \rangle \llbracket \langle \textit{subformula 3} \rangle \dots \rrbracket \rangle$

```

239 \begingroup
240 \catcode`\|=\active
241 \gdef\phx@mb@bk#1#2{\begingroup
242   \mathcode`\|="8000\def|\{\egroup#1\vert\bgroup}%
243   \def\<\mathrel{\<}\def\>\{\mathrel{\>}}%
244   \mathopen#1\langle\bgroup#2\egroup\mathclose#1\rangle\endgroup}
245 \gdef\phx@ab@bk#1{\begingroup
246   \mathcode`\|="8000\def|\{\egroup\phx@ab@bkv\bgroup}%
247   \def\<\mathrel{\<}\def\>\{\mathrel{\>}}%
248   \phx@abopen\langle\bgroup#1\egroup\phx@abclose\rangle\endgroup}
249 \endgroup
250 \def\phx@ab@bkv{\middle\vert}
251 \phx@AB@gen{bk.m}<\mr<>\phx@mb@bk#1{#2}\endgroup}
252 \phx@AB@gen{bk.a}<\r<>\phx@ab@bk{#1}
253 \phx@d@l@genxm{bk.m}
254 \phx@d@l@genxa{bk.a}
255 \DeclareRobustCommand \braket {\phx@d@lx{bk.m}{bk.a}}

```

*(End of definition for \braket. This function is documented on page 7.)*

**\ketbra**             $\langle \textit{subformula 1} \rangle \langle \textit{subformula 2} \rangle \langle \textit{subformula 3} \rangle \mid$

```

256 \begingroup
257 \catcode`\<=\active
258 \catcode`\>=\active
259 \gdef\phx@mb@kb#1#2{\begingroup
260   \mathcode`\<="8000 \mathcode`\>="8000%
261   \def\<{\#1\langle}\def\>\{\#1\rangle}%
262   \def\<\{\phx@ab@l}\def\>\{\phx@ab@r}%
263   \mathopen#1\vert#2\mathclose#1\vert\endgroup}
264 \endgroup
265 \gdef\phx@ab@kb#1#2<#3\phx@end{\begingroup
266   \def\<\{\phx@ab@l}\def\>\{\phx@ab@r}%
267   \phx@abopen\vert\mathopen{\phx@mathvphantom{#3}}#1\phx@abclose\rangle#2%
268   \phx@abopen\langle#3\mathclose{\phx@mathvphantom{#1}}\phx@abclose\vert
269 \endgroup}
270 \AtBeginDocument{\ifcsname symbf\endcsname
271   \def\phx@ab@l{\Umathchar 3 \symoperators "003C }%

```

```

272 \def\phx@abbr{\Umathchar 3 \symoperators "003E }%
273 \fi}
274 \def\phx@abb@l{\mathchar"313C }
275 \def\phx@abbr{\mathchar"313E }
276 \phx@AB@gen{kb.m}|{mr||}{\phx@mb@kb#1{#2}\endgroup}
277 \phx@AB@gen{kb.a}|{r||}{\phx@ab@kb#1\phx@end}
278 \phx@d@l@genxm{kb.m}
279 \phx@d@l@genxa{kb.a}
280 \DeclareRobustCommand \ketbra {\phx@d@lx{kb.m}{kb.a}}

```

(End of definition for `\ketbra`. This function is documented on page 8.)

**\bknorm** Define some high-level commands for inner product, outer product of basis and expectation value.

```

\bkbproj
\expval
281 \DeclareDocumentCommand \bknorm { s o m }
282   {%
283     \IfBooleanTF{#1} {\mathopen\langle#3\vert #3\mathclose\rangle}
284     {%
285       \IfValueTF{#2}
286         {\expanded{\noexpand\phx@d@lx{bk.m}{bk.a}}%
287          \expandafter\noexpand\csname #2\endcsname}<#3|#3>}
288         {\phx@d@lx{bk.m}{bk.a}<#3|#3>}%
289     }%
290   }
291 \DeclareDocumentCommand \kbproj { s o m }
292   {%
293     \IfBooleanTF{#1}
294     {%
295       \mathopen\vert#3\mathclose\rangle
296       \mathopen\langle#3\mathclose\vert
297     }
298     {%
299       \IfValueTF{#2}
300         {\expanded{\noexpand\phx@d@lx{kb.m}{kb.a}}%
301          \expandafter\noexpand\csname #2\endcsname}|#3><#3|}
302         {\phx@d@lx{kb.m}{kb.a}|#3><#3|}%
303     }%
304   }

```

Parsing the optional argument of `\expval`.

```

305 \def\@phx@ev@do@pt#1,{\ifx#1\relax\@empty\else
306 \edef\reserved@a{\zap@space#1 \@empty}%
307 \ifx\reserved@a\@empty\else

```

```

308 \ifcsname phx@del\expandafter\expandafter\expandafter
309 \expandafter\expandafter\expandafter\expandafter
310 \expandafter\expandafter\expandafter\expandafter
311 \expandafter\expandafter\expandafter\expandafter \string
312 \expandafter\expandafter\expandafter\expandafter
313 \expandafter\expandafter\expandafter \csname
314 \expandafter\expandafter\expandafter \detokenize
315 \expandafter{\reserved@a}\endcsname \endcsname
316 \xdef\@phx@ev{\reserved@a}%
317 \else
318 \xdef\@phx@ev@basis{%
319 \reserved@a}%
320 \fi
321 \fi
322 \expandafter\@phx@ev@do@pt\fi}
323 \def\phx@ev@doopt#1{%
324 \edef\@phx@ev{\@empty}\edef\@phx@ev@basis{\@empty}\@phx@ev@do@pt#1,\relax,}
325 \DeclareDocumentCommand \expval { s o m }
326 {%
327 \phx@ev@doopt{#2}%
328 \IfValueTF {#2}
329 {%
330 \ifx\@phx@ev\@empty
331 \ifx \@phx@ev@basis\@empty \else
332 \IfBooleanTF{#1}
333 {\phx@d@lx{bk.m}{bk.a}*<\@phx@ev@basis|#3|\@phx@ev@basis>}
334 {\phx@d@lx{bk.m}{bk.a}<\@phx@ev@basis|#3|\@phx@ev@basis>}%
335 \fi
336 \else
337 \ifx\@phx@ev@basis\@empty
338 \IfBooleanTF{#1} {\phx@d@lx{bk.m}{bk.a}*<#3>}
339 {%
340 \expanded{%
341 \noexpand \phx@d@lx{bk.m}{bk.a}\expandafter \noexpand
342 \csname \@phx@ev\endcsname}<#3>%
343 }%
344 \else
345 \IfBooleanTF{#1}
346 {%
347 \phx@d@lx{bk.m}{bk.a}*<\@phx@ev@basis|#3|\@phx@ev@basis>%
348 }
349 {%

```



```

350         \expanded{%
351             \noexpand \phx@d@lx{bk.m}{bk.a}\expandafter
352             \noexpand \csname \@phx@ev\endcsname}%
353         <@\phx@ev@basis|#3|\@phx@ev@basis>%
354     }%
355     \fi
356     \fi
357 }
358 {%
359     \IfBooleanTF{#1}
360     {\phx@d@lx{bk.m}{bk.a}*<#3>} {\phx@d@lx{bk.m}{bk.a}<#3>}%
361 }%
362 }

```

(End of definition for `\bknorm`, `\kbproj`, and `\expval`. These functions are documented on page 8.)

```

363 \protect \endinput
364 </ab.braket>

```

#### 4.4 The braket module

```

365 <*braket>
366 \ProvidesFile {phx-braket.sty}
367 [\phx@date\ `braket' module of physics3]

```

This module requires `\phx@abopen` and `\phx@abclose` from `ab`. This module may have conflict with `ab.braket`.

```

368 \phx@requiremodule{ab}
369 \ifdefined\phx@abb@bkv
370 \PackageWarning{physics3}{You cannot load `ab.braket' and `braket'
371 modules together.\MessageBreak Only `braket' module works now.}
372 \fi

```

```

\bra      \bra * [biggg] {subformula}
373 \DeclareDocumentCommand\bra{ s o m }{%
374     \IfBooleanTF{#1}
375     {\mathopen\langle#3\mathclose\vert}
376     {\IfValueTF{#2}
377         {\csname#2l\endcsname\langle#3\csname#2r\endcsname\vert}
378         {\phx@abopen\langle#3\phx@abclose\vert}}%
379     }%
380 }

```

(End of definition for `\bra`. This function is documented on page 6.)

```

\ket * [biggg] {subformula}
381 \DeclareDocumentCommand\ket{ s o m }{%
382 \IfBooleanTF{#1}
383 {\mathopen\vert#3\mathclose\rangle}
384 {\IfValueTF{#2}
385 {\csname#2l\endcsname\vert#3\csname#2r\endcsname\rangle}
386 {\phx@abopen\vert#3\phx@abclose\rangle}%
387 }%
388 }

```

(End of definition for \ket. This function is documented on page 6.)

```

\braket * [biggg],  $n \in \{1, 2, 3\}$  {subformula 1} ...{subformula n}
389 \DeclareDocumentCommand \braket { s O{} }
390 {%
391 \IfBooleanTF{#1}
392 {%
393 \gdef\@phx@bk@argnum{ii}%
394 \phx@bk@doopt{#2}%
395 \gdef\@phx@bk@l{\mathopen}%
396 \gdef\@phx@bk@m{\mathord}%
397 \gdef\@phx@bk@r{\mathclose}%
398 }
399 {%
400 \gdef\@phx@bk@argnum{ii}%
401 \gdef\@phx@bk@l{\phx@abopen}%
402 \gdef\@phx@bk@m{middle}%
403 \gdef\@phx@bk@r{\phx@abclose}%
404 \phx@bk@doopt{#2}%
405 }%
406 \csname phx@bk@in@\@phx@bk@argnum\endcsname
407 }

```

(End of definition for \braket. This function is documented on page 7.)

```

\phx@bk@in@i \phx@bk@in@<n.roman> {subformula 1} ...{subformula n}
\phx@bk@in@ii <n.roman> is n in roman lowercase, where  $n \in \{1, 2, 3\}$ .
\phx@bk@in@iii
408 \def\phx@bk@in@i#1{%
409 \csname\@phx@bk@l\endcsname\langle#1\rangle%
410 \csname\@phx@bk@r\endcsname\rangle}
411 \def\phx@bk@in@ii#1#2{%
412 \csname\@phx@bk@l\endcsname\langle#1\rangle%

```

```

413 \csname\@phx@bk@m\endcsname\vert{#2}%
414 \csname\@phx@bk@r\endcsname\rangle}
415 \def\phx@bk@in@iii#1#2#3{%
416 \csname\@phx@bk@l\endcsname\langle{#1}%
417 \csname\@phx@bk@m\endcsname\vert{#2}%
418 \csname\@phx@bk@m\endcsname\vert{#3}%
419 \csname\@phx@bk@r\endcsname\rangle}

(End of definition for \phx@bk@in@i, \phx@bk@in@ii, and \phx@bk@in@iii.)

```

```

\phx@bk@doopt \phx@bk@doopt {<clist>}
\phx@bk@do@pt Parse the optional argument of \braket. This will add 3 entries to hash.
420 \def\@phx@bk@do@pt#1,{\ifx#1\relax\@empty\else
421 \edef\reserved@a{\zap@space#1 \@empty}%
422 \ifx\reserved@a\@empty\else
423 \ifcsname phx@del\expandafter\string\csname\reserved@a\endcsname\endcsname
424 \xdef\@phx@bk@l{\reserved@a l}%
425 \xdef\@phx@bk@m{\reserved@a}% but not m (m stands for \mathrel)
426 \xdef\@phx@bk@r{\reserved@a r}%
427 \else
428 \ifnum\reserved@a>3%
429 \PackageError{physics3}{\string\braket\space can only take 3
430 mandatory arguments at most}{Check if you had written a number
431 more than 3 in the [optional] argument.}%
432 \fi
433 \xdef\@phx@bk@argnum{\romannumeral\reserved@a}%
434 \fi
435 \fi
436 \expandafter\@phx@bk@do@pt\fi}
437 \def\phx@bk@doopt#1{\@phx@bk@do@pt#1,\relax,}

(End of definition for \phx@bk@doopt and \phx@bk@do@pt.)

```

```

\ketbra \ketbra * [(bigg)] {<subformula 1>}
          [(between 1 and 2)] {<subformula 2>}
438 \DeclareDocumentCommand \ketbra { s o m O{ } m }
439 {%
440 \IfBooleanTF{#1}
441 {%
442 \mathopen\vert#3\mathclose\rangle#4%
443 \mathopen\langle#5\mathclose\vert
444 }

```

```

445     {%
446     \IfValueTF{#2}
447     {\csname#2l\endcsname\vert#3\csname#2r\endcsname\rangle#4%
448     \csname#2l\endcsname\langle#5\csname#2r\endcsname\vert}
449     {\begingroup
450     \phx@abopen\vert
451     \mathopen{\phx@mathvphantom{#5}}#3\phx@abclose\rangle#4%
452     \phx@abopen\langle#5
453     \mathclose{\phx@mathvphantom{#3}}\phx@abclose\vert
454     \endgroup}%
455     }%
456   }

```

(End of definition for `\ketbra`. This function is documented on page 8.)

**\bknorm** Define some high-level commands for inner product, outer product of basis and expectation value.

```

\kbproj
\expval
457 \DeclareDocumentCommand \bknorm { s o m }
458   {%
459   \IfBooleanTF{#1}
460   {%
461   \gdef\@phx@bk@l{\mathopen}%
462   \gdef\@phx@bk@m{\mathord}%
463   \gdef\@phx@bk@r{\mathclose}%
464   }
465   {%
466   \gdef\@phx@bk@l{\phx@abopen}%
467   \gdef\@phx@bk@m{middle}%
468   \gdef\@phx@bk@r{\phx@abclose}%
469   \IfValueT{#2}
470   {%
471   \ifcsname phx@del\expandafter\string\csname#2\endcsname\endcsname
472   \xdef\@phx@bk@l{#2l}%
473   \xdef\@phx@bk@m{#2}%
474   \xdef\@phx@bk@r{#2r}%
475   \fi
476   }%
477   }%
478   \phx@bk@in@ii {#3} {#3}%
479   }
480 \DeclareDocumentCommand \kbproj { s o m }
481   {%

```

```

482 \IfBooleanTF{#1}
483   {%
484     \mathopen\vert#3\mathclose\rangle
485     \mathopen\langle#3\mathclose\vert
486   }
487   {%
488     \IfValueTF{#2}
489       {\csname#2l\endcsname\vert#3\csname#2r\endcsname\rangle
490        \csname#2l\endcsname\langle#3\csname#2r\endcsname\vert}
491     {\begingroup
492      \phx@abopen\vert
493      \mathopen{\phx@mathvphantom{#3}}#3\phx@abclose\rangle
494      \phx@abopen\langle#3
495      \mathclose{\phx@mathvphantom{#3}}\phx@abclose\vert
496      \endgroup}%
497   }%
498 }
499 \def\@phx@ev@do@pt#1,{\ifx#1\relax\@empty\else
500 \edef\reserved@a{\zap@space#1 \@empty}%
501 \ifx\reserved@a\@empty\else
502 \ifcsname phx@del\expandafter\expandafter\expandafter
503 \expandafter\expandafter\expandafter\expandafter
504 \expandafter\expandafter\expandafter\expandafter
505 \expandafter\expandafter\expandafter\expandafter \string
506 \expandafter\expandafter\expandafter\expandafter
507 \expandafter\expandafter\expandafter \csname
508 \expandafter\expandafter\expandafter \detokenize
509 \expandafter{\reserved@a}\endcsname \endcsname
510 \xdef\@phx@ev{\reserved@a}%
511 \else
512 \xdef\@phx@ev@basis{%
513 \reserved@a}%
514 \fi
515 \fi
516 \expandafter\@phx@ev@do@pt\fi}
517 \def\phx@ev@doopt#1{%
518 \edef\@phx@ev{\@empty}\edef\@phx@ev@basis{\@empty}\@phx@ev@do@pt#1,\relax,}
519 \DeclareDocumentCommand \expval { s O{ } m }
520 {
521 \phx@ev@doopt{#2}%
522 \IfBooleanTF{#1}
523   {%

```

```

524     \gdef\@phx@bk@l{mathopen}%
525     \gdef\@phx@bk@m{mathord}%
526     \gdef\@phx@bk@r{mathclose}%
527   }
528   {%
529     \gdef\@phx@bk@l{phx@abopen}%
530     \gdef\@phx@bk@m{middle}%
531     \gdef\@phx@bk@r{phx@abclose}%
532     \ifx \@phx@ev\@empty \else
533       \xdef\@phx@bk@l{\@phx@ev l}%
534       \xdef\@phx@bk@m{\@phx@ev}%
535       \xdef\@phx@bk@r{\@phx@ev r}%
536     \fi
537   }%
538   \ifx \@phx@ev@basis\@empty
539     \phx@bk@in@i {#3}%
540   \else
541     \phx@bk@in@iii {\@phx@ev@basis} {#3} {\@phx@ev@basis}%
542   \fi
543 }

```

(End of definition for `\bknorm`, `\kbproj`, and `\expval`. These functions are documented on page 8.)

```
544 \protect \endinput
```

```
545 </braket>
```

Restore the namespace.

```
546 <@@=phx>
```

## 4.5 The doubleprod module

```
547 <*doubleprod>
```

```
548 \ProvidesExplFile {phx-doubleprod.sty} {\phx@date} {\phx@version}
```

```
549 {'doubleprod' (vertically stacked binary operators) module of physics3}
```

Boolean options.

```
550 \__phx_define_key:nnnn { doubleprod } { crosssymbol } { } { \def\@phx@dbl@c {#1} }
```

```
551 \__phx_define_key:nnnn { doubleprod } { dotsymbol } { } { \def\@phx@dbl@d {#1} }
```

```
552 \__phx_define_key:nnnn { doubleprod } { crossscale } { } { \def\@phx@dbl@sc{#1} }
```

```
553 \__phx_define_key:nnnn { doubleprod } { dotscale } { } { \def\@phx@dbl@sd{#1} }
```

```
554 \__phx_define_key:nnnn { doubleprod } { crossopenup } { } { \def\@phx@dbl@oc{#1} }
```

```
555 \__phx_define_key:nnnn { doubleprod } { dotopenup } { } { \def\@phx@dbl@od{#1} }
```

```
556 \__phx_setkeys:nn { doubleprod }
```

```

557 {
558   crosssymbol = \times, crossscale = 0.8, crossopenup = .02,
559   dotsymbol   = \ldotp, dotscale   = 1,   dotopenup   = .2
560 }
561 \__phx_processkeyopt:n { doubleprod }
562 \def\phx@dbl@gen#1#2#3#4{%
563   \DeclareRobustCommand#1{\mathbin{\vcenter{\baselineskip\z@skip%
564     \lineskip#4\phx@dblcurrf@size%
565     \hbox_set:Nn \l_tmpa_box {\fontsize{#2\phx@dblcurrf@size}\z@#3$}
566     \box_use:N \l_tmpa_box \box_use_drop:N \l_tmpa_box}}}}
567 \def\phx@dblcurrf@size{\dimexpr\font@size pt\relax}
568 \phx@dbl@gen\doublecross\@phx@dbl@sc\@phx@dbl@c\@phx@dbl@oc
569 \phx@dbl@gen\doubledot\@phx@dbl@sd\@phx@dbl@d\@phx@dbl@od
570 \file_input_stop:
571 </doubleprod>

```

#### 4.6 The diagmat module

```

572 <*diagmat>
573 \ProvidesExplFile {phx-diagmat.sty} {\phx@date} {\phx@version}
574   {'diagmat' module of physics3}

```

This module requires mathtools and xpatch.

```

575 \RequirePackage { mathtools, etoolbox }

```

Do expansion of column specification argument (thanks @egreg on [T<sub>E</sub>X StackExchange](#)).

```

576 \patchcmd \MT_matrix_begin:N { \exp_args:Ne \array } { \exp_args:Ne \array } {}
577   { \patchcmd \MT_matrix_begin:N { \array } { \exp_args:Ne \array } {} {} }
578 \__phx_define_key:nnnn { diagmat } { empty } {   }
579   { \tl_gset:Nn \l__phx_mat_empty_tl { #1 } }
580 \__phx_define_key:nnnn { diagmat } { align } { r }
581   { \tl_gset:Nn \l__phx_mat_align_tl { #1 } }

```

This module requires some new variables.

```

\l__phx_mat_diag_clist
\l__phx_mat_dim_int
\l__phx_mat_line_tl
\l__phx_diagmat_tl
\l__phx_mat_empty_tl
\l__phx_mat_align_tl
582 \clist_new:N \l__phx_mat_diag_clist
583 \int_new:N \l__phx_mat_dim_int
584 \tl_new:N \l__phx_mat_line_tl
585 \tl_new:N \l__phx_diagmat_tl
586 \tl_new:N \l__phx_mat_empty_tl
587 \tl_new:N \l__phx_mat_align_tl
588 \__phx_processkeyopt:n { diagmat }
589 \keys_define:nn { phy/diagmat }
590   {
591     empty .tl_set:N = \l__phx_mat_empty_tl,

```

```

592   align   .tl_set:N = \l__phx_mat_align_tl,
593 }

```

(End of definition for `\l__phx_mat_diag_clist` and others.)

```

\diagmat      \langle delimiter type \rangle diagmat [ \langle key-val list \rangle ] { \langle diagonal \rangle }
\pdiagmat
\bdiagmat
\Bdiagmat
\vdiagmat
\Vdiagmat
594 \clist_map_inline:nn { {} , p , b , B , v , V }
595 {
596   \exp_args:Nc \DeclareDocumentCommand { #1diagmat } { s 0 {} m }
597   {
598     \IfBooleanTF { ##1 }
599     { \__phx_diagmat_type:nnn { #1small } { ##2 } { ##3 } }
600     { \__phx_diagmat_type:nnn { #1          } { ##2 } { ##3 } }
601   }
602 }

```

(End of definition for `\diagmat` and others. These functions are documented on page 9.)

```

\__phx_diagmat_type:nnn      \__phx_diagmat_type:nnn { \langle delimiter type \rangle }
                             { \langle key-val list \rangle } { \langle diagonal \rangle }
603 \cs_new:Npn \__phx_diagmat_type:nnn #1#2#3
604 {
605   \group_begin:
606   \clist_set:Nn \l__phx_mat_diag_clist {#3}
607   \int_set:Nn \l__phx_mat_dim_int { \clist_count:N \l__phx_mat_diag_clist }
608   \int_compare:nNnT { \l__phx_mat_dim_int } > { \value { MaxMatrixCols } }
609   { \setcounter { MaxMatrixCols } { \l__phx_mat_dim_int } }
610   \keys_set:nn { phy/diagmat } {#2}
611   \tl_gclear:N \l__phx_diagmat_tl
612   \int_step_inline:nnn { 1 } { \l__phx_mat_dim_int }
613   {
614     \int_step_inline:nnn { 1 } { \l__phx_mat_dim_int }
615     {
616       \int_compare:nNnTF { ##1 } = { ####1 }
617       {
618         \clist_gpop:NN \l__phx_mat_diag_clist \l__phx_tmpa_tl
619         \tl_if_empty:NTF \l__phx_tmpa_tl
620         { \tl_gput_right:Nn \l__phx_mat_line_tl { \l__phx_mat_empty_tl } }

```

Maybe it's better to use `\expandafter\scantokens\expandafter{\l__phx_tmpa_tl}` in the next line.

```

621         { \tl_gput_right:No \l__phx_mat_line_tl { \l__phx_tmpa_tl } }

```



```

622     }
623     { \tl_gput_right:Nn \l__phx_mat_line_tl { \l__phx_mat_empty_tl } }
624 \int_compare:nNnTF { ###1 } = { \l__phx_mat_dim_int }
625   {
626     \tl_gput_right:Nn \l__phx_mat_line_tl { \ }
627   }
628   {
629     \tl_gput_right:Nn \l__phx_mat_line_tl { & }
630   }
631 }
632 \tl_gput_right:No \l__phx_diagmat_tl { \l__phx_mat_line_tl }
633 \tl_gclear:N \l__phx_mat_line_tl
634 }
635 \tl_if_empty:NTF { \l__phx_mat_align_tl }
636 {
637   \begin { #1 matrix }
638     \tl_use:N \l__phx_diagmat_tl
639   \end { #1 matrix }
640 }
641 {
642   \begin { #1 matrix* } [ \l__phx_mat_align_tl ]
643     \tl_use:N \l__phx_diagmat_tl
644   \end { #1 matrix* }
645 }
646 \group_end:
647 }

```

(End of definition for `\__phx_diagmat_type:nnn`.)

```

648 \file_input_stop:
649 </diagmat>

```

## 4.7 The `xmat` module

```

650 <*xmat>
651 \ProvidesExplFile {phx-xmat.sty} {\phx@date} {\phx@version}
652   {'xmat' module of physics3}

```

This module requires `mathtools` and `xpatch`.

```

653 \RequirePackage { mathtools, etoolbox }

```

Do expansion of column specification argument (thanks @egreg on [T<sub>E</sub>X StackExchange](#)).

```

654 \patchcmd \MT_matrix_begin:N { \exp_args:Ne \array } { \exp_args:Ne \array } {}
655   { \patchcmd \MT_matrix_begin:N { \array } { \exp_args:Ne \array } {} {} }

```

```

656 \_phx_define_key:nnnn { xmat } { showtop } { }
657   { \int_gset:Nn \l__phx_xmat_showtop_int { #1 } }
658 \_phx_define_key:nnnn { xmat } { showleft } { }
659   { \int_gset:Nn \l__phx_xmat_showleft_int { #1 } }

```

\l\_\_phx\_xmat\_extra\_vdots\_bool This module requires some new variables.

```

\l__phx_xmat_extra_cdots_bool 660 \bool_new:N \l__phx_xmat_extra_vdots_bool
\l__phx_xmat_showtop_int 661 \bool_new:N \l__phx_xmat_extra_cdots_bool
\l__phx_xmat_showleft_int 662 \int_new:N \l__phx_xmat_showtop_int
\l__phx_xmat_tl 663 \int_new:N \l__phx_xmat_showleft_int
664 \tl_new:N \l__phx_xmat_tl
665 \int_gset:Nn \l__phx_xmat_showtop_int { \value { MaxMatrixCols } - 2 }
666 \int_gset:Nn \l__phx_xmat_showleft_int { \value { MaxMatrixCols } - 2 }

```

(End of definition for \l\_\_phx\_xmat\_extra\_vdots\_bool and others.)

```

667 \cs_new:Npn \_phx_xmat_entry_format:nnn #1#2#3
668   {
669     #1 \c_math_subscript_token { #2 #3 }
670   }
671 \_phx_processkeyopt:n { xmat }
672 \clist_map_inline:nn { {}, p, b, B, v, V }
673   {
674     \exp_args:Nc \DeclareDocumentCommand { #1xmat } { s O{} m m m }
675     {
676       \IfBooleanTF { ##1 }
677         { \_phx_xmat_type:nnnnn { #1small } { ##2 } { ##3 } { ##4 } { ##5 } }
678         { \_phx_xmat_type:nnnnn { #1 } { ##2 } { ##3 } { ##4 } { ##5 } }
679     }
680   }
681 \keys_define:nn { phx / xmat }
682   {
683     format .cs_set:Np = \_phx_xmat_entry_format:nnn #1#2#3 ,
684     showtop .int_set:N = \l__phx_xmat_showtop_int ,
685     showleft.int_set:N = \l__phx_xmat_showleft_int ,
686   }

```

\\_phx\_if\_digits\_only\_p:n \\_phx\_if\_digits\_only:nTF {<token list>} {<>true code>} {<false code>}  
 \\_phx\_if\_digits\_only:nTF Use L<sup>A</sup>T<sub>E</sub>X3 regular expression to tell if <token list> (the numbers of rows or columns) contain digits only.

(End of definition for \\_phx\_if\_digits\_only:nTF)

```

687 \prg_new_conditional:Npnn \_phx_if_digits_only:n #1 { TF }

```

```

688 {
689   \regex_match:nnTF { \A [[:digit:]]* \Z } { #1 }
690   { \prg_return_true: } { \prg_return_false: }
691 }

```

```

\__phx_xmat_type:nnnnn   \__phx_xmat_type:nnnnn {<delimiter type>} {<key-val list>}
                        {<common entry>} {<row number>} {<column number>}

```

(End of definition for \\_\_phx\_xmat\_type:nnnnn.)

```

692 \cs_new:Npn \__phx_xmat_type:nnnnn #1#2#3#4#5
693 {
694   \group_begin:
695   \tl_gclear:N \l__phx_xmat_tl
696   \keys_set:nn { phx / xmat } { #2 } %
697   \__phx_if_digits_only:nTF { #4 }
698   {
699     \int_compare:nNnTF { #4 } < { \l__phx_xmat_showtop_int + 1 }
700     {
701       \int_set:Nn \l__phx_xmat_showtop_int { #4 }
702       \bool_set_false:N \l__phx_xmat_extra_vdots_bool
703     }
704     {
705       \bool_set_true:N \l__phx_xmat_extra_vdots_bool
706     }
707   }
708   {
709     \bool_set_true:N \l__phx_xmat_extra_vdots_bool
710   }
711   \__phx_if_digits_only:nTF { #5 }
712   {
713     \int_compare:nNnTF { #5 } < { \l__phx_xmat_showleft_int + 1 }
714     {
715       \int_set:Nn \l__phx_xmat_showleft_int { #5 }
716       \bool_set_false:N \l__phx_xmat_extra_cdots_bool
717     }
718     {
719       \bool_set_true:N \l__phx_xmat_extra_cdots_bool
720     }
721   }
722   {
723     \bool_set_true:N \l__phx_xmat_extra_cdots_bool
724   }

```

```

725 \int_step_inline:nn { \l__phx_xmat_showtop_int }
726 {
727   \tl_put_right:Nn \l__phx_xmat_tl
728   { \__phx_xmat_entry_format:nnn { #3 } { ##1 } { 1 } }
729   \int_step_inline:nnn { 2 } { \l__phx_xmat_showleft_int }
730   {
731     \tl_put_right:Nn \l__phx_xmat_tl
732     { & \__phx_xmat_entry_format:nnn { #3 } { ##1 } { ####1 } }
733   }
734   \bool_if:NT \l__phx_xmat_extra_cdots_bool
735   {
736     \tl_put_right:Nn \l__phx_xmat_tl
737     { & \cdots & \__phx_xmat_entry_format:nnn { #3 } { ##1 } { #5 } }
738   }
739   \tl_put_right:Nn \l__phx_xmat_tl { \\ }
740 }
741 \bool_if:NT \l__phx_xmat_extra_vdots_bool
742 {
743   \tl_put_right:Nn \l__phx_xmat_tl { \vdots }
744   \prg_replicate:nn { \l__phx_xmat_showleft_int - 1 }
745   {
746     \tl_put_right:Nn \l__phx_xmat_tl { & \vdots }
747   }
748   % Add \ddots if vdots_bool and cdots_bool be true simultaneously.
749   \bool_if:NT \l__phx_xmat_extra_cdots_bool
750   {
751     \tl_put_right:Nn \l__phx_xmat_tl { & \ddots & \vdots }
752   } % else relax
753   \tl_put_right:Nn \l__phx_xmat_tl { \\ }
754   % The last row.
755   \tl_put_right:Nn \l__phx_xmat_tl
756   { \__phx_xmat_entry_format:nnn { #3 } { #4 } { 1 } }
757   \int_step_inline:nnn { 2 } { \l__phx_xmat_showleft_int }
758   {
759     \tl_put_right:Nn \l__phx_xmat_tl
760     { & \__phx_xmat_entry_format:nnn { #3 } { #4 } { ##1 } }
761   }
762   \bool_if:NT \l__phx_xmat_extra_cdots_bool
763   {
764     \tl_put_right:Nn \l__phx_xmat_tl
765     { & \cdots & \__phx_xmat_entry_format:nnn { #3 } { #4 } { #5 } }
766   }

```

```

767     } % else relax
768     \begin { #1 matrix }
769         \tl_use:N \l__phx_xmat_tl
770     \end { #1 matrix }
771     \group_end:
772 }

773 \file_input_stop:
774 \</xmat>

```

## 4.8 The operator module

```

775 <*operator>
776 \ProvidesExplFile {phx-operator.sty} {\phx@date} {\phx@version}
777 {`operator' module of physics3}
778 \__phx_define_key:nnnn { op.lega } { ReIm } { true } { \def\phx@reserveda{#1} }
779 \__phx_define_key:nnnn { op.lega } { PV } { } { \def\@phx@oplega@PV{#1} }
780 \__phx_define_key:nnnn { op.lega } { pv } { } { \def\@phx@oplega@pv{#1} }
781 \__phx_setkeys:nn { op.lega } { PV = \mathcal{P}, pv = {p.v.}, ReIm = true}
782 \__phx_processkeyopt:n { ab }

\asin
\acos 783 \clist_map_inline:nn
\atan 784 { asin, acos, atan, acsc, asec, acot, Tr, tr, rank, erf, Res, res }
\acsc 785 {
\asec 786 \exp_args:Nc \DeclareRobustCommand {#1}
\acot 787 { \mathop { \operator@font #1 } \nolimits }
788 }
\Tr 789 \DeclareRobustCommand\PV {\mathord{\@phx@oplega@PV}}
\tr 790 \DeclareRobustCommand\pv {\mathop{\operator@font\@phx@oplega@pv}}\nolimits}
\rank 791 \DeclareRobustCommand\upe {\mathrm e}
\erf 792 \DeclareRobustCommand\iu {\mathrm i\mkern1mu}
\Res 793 \DeclareRobustCommand\identity{\mathbb I}

\res (End of definition for \asin and others.)
\pv
\Re Restore \Re and \Im in \Resymbol and \Imsymbol. The \AtBeginDocument hook is
\Im used for the compatibility of unicode-math.
\iu
794 \ifx \phx@reserveda \phx@true
\identity 795 \AtBeginDocument
796 {
797 \let\Resymbol\Re
798 \let\Imsymbol\Im

```

```

799 \DeclareRobustCommand\Re{\mathop{\operator@font Re}\nolimits}%
800 \DeclareRobustCommand\Im{\mathop{\operator@font Im}\nolimits}%
801 }
802 \fi

```

*(End of definition for \Re and \Im.)*

Requires fixdif version 2.x.

```

803 \RequirePackage{fixdif}[2023/01/31]
804 \letdif\phx@nl@nabla{nabla}

```

```

\grad ∇-related operators.
\div 805 \phx@requiremodule{ab}
\curl 806 \AtBeginDocument
\laplacian 807 {
808   \ifcsname div\endcsname\let\divsymbol\div\fi
809   \DeclareRobustCommand \grad    {\phx@nl@nabla\ab}
810   \DeclareRobustCommand \div     {\phx@nl@nabla\cdot\ab}
811   \DeclareRobustCommand \curl    {\phx@nl@nabla\times\ab}
812   \DeclareRobustCommand \laplacian{\phx@nl@nabla^2\ab}
813 }

```

*(End of definition for \grad and others.)*

```

814 \file_input_stop:
815 </operator>

```

## 4.9 The bm-um module

```

816 <*bm-um>
817 \ProvidesExplFile {phx-bm-um.sty} {\phx@date} {\phx@version}
818   {'bm-um' module of physics3}
819 \AtBeginDocument
820   {
821     \cs_if_exist:cF { symbf }
822     {
823       \PackageError { physics3 }
824       {
825         The ~ `bm-um' ~ module ~ requires ~
826         `unicode-math' ~ package
827       }
828     }
829     Have ~ you ~ used ~ `unicode-math' ~
830     in ~ the ~ preamble?

```

```

831     }
832   }
833 }
834 \DeclareDocumentCommand \bm { m }
835 {
836   \mode_if_math:TF
837   {
838     \tl_map_inline:nn { #1 }
839     {
840       \tl_if_head_eq_catcode:nNTF { ##1 } A
841       { \symbfit { ##1 } } { \sympf { ##1 } }
842     }
843   }
844   {
845     \msg_error:nn { physics3 }
846     { The ~ \string\bm\space command ~ requires ~ math ~ mode. }
847   }
848 }
849 \file_input_stop:
850 </bm-um>

```

#### 4.10 The ab.legacy module

```

851 <*ab.legacy>
852 \ProvidesFile {phx-ab.legacy.sty}
853   [\phx@date\ `ab.legacy' module of physics3]

```

Requires ab's tight option.

```

854 \phx@requiremodule{ab}
855 \phx@define@key{ab.legacy}{order}{\mathcal 0}{\def\phx@ab@ordersym{#1}}
856 \phx@setkeys{ab.legacy}{order}
857 \phx@processkeyopt{ab.legacy}
858 \phx@d@l@geny\abs\vert\vert
859 \phx@d@l@geny\norm\Vert\Vert
860 \DeclareDocumentCommand\order{som}{%
861   \phx@ab@ordersym
862   \IfBooleanTF{#1}
863   {(#3)}
864   {\IfValueTF{#2}
865     {\csname#2l\endcsname(#3\csname#2r\endcsname)}
866     {\phx@abopen(#3\phx@abclose)}}%
867   }%
868 }

```

```

869 \phx@d@l@geny\eval.\vert
870 \phx@d@l@geny\peval(\vert
871 \phx@d@l@geny\beval[\vert
872 \protect \endinput
873 </ab.legacy>

```

#### 4.11 The `qtext.legacy` module

This module is written for the compatibility with the bad commands provided by physics only. The commands in this module should NEVER be used!

```

874 <*qtext.legacy>
875 \ProvidesExplFile {phx-qtext.legacy.sty} {\phx@date} {\phx@version}
876   {`qtext.legacy' module of physics3.sty}
877 \RequirePackage{amstext}
878 \def\phx@qtext@#1#2{#1\text{#2}\quad}
879 \DeclareRobustCommand\qqtext{\@ifstar{\phx@qtext@{}}{\phx@qtext@\quad}}
880 \DeclareRobustCommand\qq{\qqtext}
881 \DeclareRobustCommand\qcomma{,\quad}
882 \DeclareRobustCommand\qc{\qcomma}
883 \DeclareRobustCommand\qcc{\@ifstar{\phx@qtext@{c.c}}{\phx@qtext@\quad{c.c}}}
884 \clist_map_inline:nn
885   {
886     if,      then,  else,  otherwise,  unless,
887     give,    using, unless, assume,   since,
888     let,     for,   all,   even,      odd,
889     integer, and,   or,    as,       in
890   }
891   {
892     \exp_args:Nc \DeclareRobustCommand { q#1 }
893       { \@ifstar{ \phx@qtext@ { } {#1} } { \phx@qtext@ \quad {#1} } }
894   }
895 \file_input_stop:
896 </qtext.legacy>

```



# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	B
<code>\*diagmat</code> . . . . .	<code>\Bab</code> . . . . .
<code>\*xmat</code> . . . . .	<code>\bab</code> . . . . .
<code>\&lt;</code> . . . . .	<code>\baselineskip</code> . . . . .
<code>\&lt;</code> . . . . .	<code>\Bdiagmat</code> . . . . .
<code>\&gt;</code> . . . . .	<code>\bdiagmat</code> . . . . .
<code>\&gt;</code> . . . . .	<code>\begin</code> . . . . .
<code>\ </code> . . . . .	<code>\beginngroup</code> . . . . .
<code>\{</code> . . . . .	<code>\beval</code> . . . . .
<code>\}</code> . . . . .	<code>\bgroup</code> . . . . .
<code>\langle cmd \rangle</code> . . . . .	<code>\Big</code> . . . . .
<code>\langle delimiter type \rangle xmat</code> . . . . .	<code>\big</code> . . . . .
<code>\langle delimiter type \rangle diagmat</code> . . . . .	<code>\Bigg</code> . . . . .
<code>\lrcorner</code> . . . . .	<code>\bigg</code> . . . . .
<code>\lvert</code> . . . . .	<code>\biggg</code> . . . . .
	<code>\bigggg</code> . . . . .
<b>A</b>	<code>\bigggl</code> . . . . .
<code>\A</code> . . . . .	<code>\biggggl</code> . . . . .
<code>\aab</code> . . . . .	<code>\Bigggm</code> . . . . .
<code>\ab</code> . . . . .	<code>\bigggm</code> . . . . .
<code>ab.braket (option)</code> . . . . .	<code>\Bigggr</code> . . . . .
<code>\abs</code> . . . . .	<code>\bigggr</code> . . . . .
<code>\acomm</code> . . . . .	<code>\Biggl</code> . . . . .
<code>\acos</code> . . . . .	<code>\biggl</code> . . . . .
<code>\acot</code> . . . . .	<code>\Biggr</code> . . . . .
<code>\acsc</code> . . . . .	<code>\biggr</code> . . . . .
<code>\active</code> . . . . .	<code>\Bigl</code> . . . . .
<code>\aftergroup</code> . . . . .	<code>\bigl</code> . . . . .
<code>\array</code> . . . . .	<code>\Bigr</code> . . . . .
<code>\asec</code> . . . . .	<code>\bigR</code> . . . . .
<code>\asin</code> . . . . .	<code>\bknorm</code> . . . . .
<code>\atan</code> . . . . .	<code>\bm</code> . . . . .
<code>\AtBeginDocument</code> . . . . .	<code>\boldsymbol</code> . . . . .
<code>\AtEndOfPackage</code> . . . . .	bool commands:
	<code>\bool_if:NTF</code> . . . . .

<code>\bool_new:N</code> . . . . .	660, 661	<code>\curl</code> . . . . .	15, <a href="#">805</a>
<code>\bool_set_false:N</code> . . . . .	702, 716	<b>D</b>	
<code>\bool_set_true:N</code> . . . . .	705, 709, 719, 723	<code>\ddots</code> . . . . .	748, 751
box commands:		<code>\DeclareDocumentCommand</code> . . . . .	129,
<code>\box_dp:N</code> . . . . .	86, 92, 98, 104	204, 281, 291, 325, 373, 381, 389,	
<code>\box_ht:N</code> . . . . .	85, 91, 97, 103	438, 457, 480, 519, 596, 674, 834, 860	
<code>\box_new:N</code> . . . . .	8	<code>\DeclareRobustCommand</code> 57, 58, 61, 62,	
<code>\box_set_dp:Nn</code> . . . . .	86, 92, 98, 104	64, 66, 69, 70, 73, 74, 75, 76, 77, 78,	
<code>\box_set_ht:Nn</code> . . . . .	85, 91, 97, 103	202, 233, 238, 255, 280, 563, 786, 789,	
<code>\box_use:N</code> . . . . .	566	790, 791, 792, 793, 799, 800, 809, 810,	
<code>\box_use_drop:N</code> . . . . .	87, 93, 99, 105, 566	811, 812, 879, 880, 881, 882, 883, 892	
<code>\l_tmpa_box</code> . . . . .	84, 85, 86, 90,	<code>\def</code> . . . . .	4, 5, 10, 11, 27, 30, 33, 36, 37, 50,
91, 92, 96, 97, 98, 102, 103, 104, 565, 566		51, 52, 53, 55, 56, 118, 128, 170, 171,	
<code>\bra</code> . . . . .	6, 29, 33, <a href="#">229</a> , 233, <a href="#">373</a>	173, 175, 177, 179, 181, 183, 185, 187,	
braket (option) . . . . .	6	189, 192, 193, 197, 198, 203, 242, 243,	
<code>\braket</code> . . . . .	6, 7, 30, 35, <a href="#">239</a> , <a href="#">389</a> , 429	246, 247, 250, 261, 262, 266, 271, 272,	
<code>\Bxmat</code> . . . . .	10, 17	274, 275, 305, 323, 408, 411, 415, 420,	
<code>\bxmat</code> . . . . .	10, 17	437, 499, 517, 550, 551, 552, 553, 554,	
<b>C</b>		555, 562, 567, 778, 779, 780, 855, 878	
<code>\catcode</code> . . . . .	240, 257, 258	<code>\delclose</code> . . . . .	4, 20, 22, 23, <a href="#">57</a> , 123
<code>\cdot</code> . . . . .	15, 810	<code>\delimiter</code> . . . . .	55, 56
<code>\cdots</code> . . . . .	737, 765	<code>\delopen</code> . . . . .	4, 20, 22, 23, <a href="#">57</a> , 122
clist commands:		<code>\detokenize</code> . . . . .	314, 508
<code>\clist_count:N</code> . . . . .	607	<code>\diagmat</code> . . . . .	9, 17, <a href="#">594</a>
<code>\clist_gpop:NN</code> . . . . .	618	<code>\dimexpr</code> . . . . .	567
<code>\clist_map_inline:n</code> . . . . .		<code>\displaystyle</code> . . . . .	84
594, 672, 783, 884		<code>\div</code> . . . . .	15, 16, <a href="#">805</a>
<code>\clist_new:N</code> . . . . .	582	<code>\divsymbol</code> . . . . .	11, 16, 808
<code>\clist_set:Nn</code> . . . . .	606	<code>\doublecross</code> . . . . .	9, 568
<code>\comm</code> . . . . .	15	<code>\doubledot</code> . . . . .	9, 569
<code>\count</code> . . . . .	13	<b>E</b>	
<code>\cross</code> . . . . .	15	<code>\edef</code> . . . . .	19, 20, 45, 46, 306, 324, 421, 500, 518
cs commands:		<code>\egroup</code> . . . . .	58, 242, 244, 246, 248
<code>\cs_if_exist:NTF</code> . . . . .	821	<code>\else</code> . . . . .	38, 39, 63, 124,
<code>\cs_new:Npn</code> . . . . .	16, 79, 111, 603, 667, 692	188, 195, 305, 307, 317, 331, 336, 344,	
<code>\cs_new_nopar:Npn</code> . . . . .	12, 14	420, 422, 427, 499, 501, 511, 532, 540	
<code>\cs_set_eq:NN</code> . . . . .	24, 25, 26, 108	<code>\end</code> . . . . .	639, 644, 770
<code>\csname</code> 129, 171, 173, 175, 177, 179, 181,		<code>\endcsname</code> . . . . .	49, 129, 171,
183, 191, 193, 196, 198, 199, 208, 287,		173, 175, 177, 179, 181, 183, 186, 191,	
301, 313, 342, 352, 377, 385, 406, 409,		193, 196, 198, 199, 208, 270, 287, 301,	
410, 412, 413, 414, 416, 417, 418, 419,		315, 342, 352, 377, 385, 406, 409, 410,	
423, 447, 448, 471, 489, 490, 507, 865			

412, 413, 414, 416, 417, 418, 419, 423,  
447, 448, 471, 489, 490, 509, 808, 865

`\endgroup` ..... 25,  
26, 151, 153, 155, 157, 159, 161, 163,  
165, 167, 169, 194, 229, 234, 244, 248,  
249, 251, 263, 264, 269, 276, 454, 496

`\endinput` ..... 219, 363, 544, 872

`\erf` ..... [783](#)

`\eval` ..... 14, 869

`\eval□(\peval□\beval)` ..... 12

exp commands:

`\exp_args:Nc` ..... 596, 674, 786, 892

`\exp_args:Ne` ..... 576, 577, 654, 655

`\expandafter` ... 43, 46, 128, 171, 173,  
175, 177, 179, 181, 183, 193, 198, 287,  
301, 308, 309, 310, 311, 312, 313, 314,  
315, 322, 341, 351, 423, 436, 471, 502,  
503, 504, 505, 506, 507, 508, 509, 516

`\expanded` ..... 286, 300, 340, 350

`\expval` ..... 8, 31, [281](#), [457](#)

**F**

`\fam` ..... 20

`\fi` ..... 42, 44, 54, 71, 72,  
127, 190, 195, 228, 273, 320, 321, 322,  
335, 355, 356, 372, 432, 434, 435, 436,  
475, 514, 515, 516, 536, 542, 802, 808

file commands:

`\file_input_stop:` .....  
..... 112, 570, 648, 773, 814, 849, 895

`\fontsize` ..... 565

**G**

`\gdef` 241, 245, 259, 265, 393, 395, 396, 397,  
400, 401, 402, 403, 461, 462, 463, 466,  
467, 468, 524, 525, 526, 529, 530, 531

`\grad` ..... 15, [805](#)

group commands:

`\group_begin:` ..... 605, 694

`\group_end:` ..... 646, 771

**H**

`\hbox` ..... 65, 67

hbox commands:

`\hbox_set:Nn` .. 81, 84, 90, 96, 102, 565

**I**

`\identity` ..... [783](#)

`\IfBooleanTF` ..... 205,  
283, 293, 332, 338, 345, 359, 374, 382,  
391, 440, 459, 482, 522, 598, 676, 862

`\ifcsname` ..... 27,  
49, 186, 270, 308, 423, 471, 502, 808

`\ifdefined` ..... 60, 68, 225, 369

`\ifnum` ..... 428

`\IfValueT` ..... 469

`\IfValueTF` ..... 207,  
285, 299, 328, 376, 384, 446, 488, 864

`\ifx` ..... 20,  
38, 39, 121, 195, 305, 307, 330, 331,  
337, 420, 422, 499, 501, 532, 538, 794

`\Im` ..... 11, 16, 45, [794](#)

`\imat` ..... 13

`\Imsymbol` ..... 11, 16, 45, 798

int commands:

`\int_compare:nNnTF` .....  
..... 608, 616, 624, 699, 713

`\int_gset:Nn` ..... 657, 659, 665, 666

`\int_new:N` ..... 583, 662, 663

`\int_set:Nn` ..... 607, 701, 715

`\int_step_inline:nn` ..... 725

`\int_step_inline:nnn` .....  
..... 612, 614, 729, 757

`\iu` ..... [783](#)

**K**

`\kbproj` ..... 8, [281](#), [457](#)

`\ket` ..... 6, 29, 34, [234](#), 238, [381](#)

`\ketbra` ..... 6–8, 35, [256](#), [438](#)

keys commands:

`\keys_define:nn` ..... 589, 681

`\keys_set:nn` ..... 610, 696

**L**

`\langle` ..... 4, 148, 149,  
161, 168, 169, 217, 229, 230, 244, 248,

261, 268, 283, 296, 375, 377, 378, 409, 412, 416, 443, 448, 452, 485, 490, 494	<code>\newcount</code> . . . . . 13
<code>\laplacian</code> . . . . . 805	<code>\newtoks</code> . . . . . 13
<code>\lbrace</code> . . . . . 4, 142, 143, 155, 162, 163, 215	<code>\noexpand</code> . . . . . 20, 40, 41, 286, 287, 300, 301, 341, 351, 352
<code>\ldotp</code> . . . . . 9, 559	<code>\nolimits</code> . . . . . 787, 790, 799, 800
<code>\left</code> . . . . . 20, 23, 57, 65, 67, 125	<code>\norm</code> . . . . . 12, 859
<code>\let</code> . . . . . 18, 22, 28, 122, 123, 125, 126, 195, 797, 798, 808	
<code>\letdif</code> . . . . . 804	
<code>\lineskip</code> . . . . . 564	
	<b>O</b>
	options:
	<code>ab.braket</code> . . . . . 6
	<code>braket</code> . . . . . 6
	<code>\order</code> . . . . . 12, 860
<b>M</b>	<b>P</b>
<code>\mathbb</code> . . . . . 793	<code>\pab</code> . . . . . 28, 213
<code>\mathbin</code> . . . . . 563	<code>\PackageError</code> . . . . . 429, 823
<code>\mathcal</code> . . . . . 781, 855	<code>\PackageWarning</code> . . . . . 226, 370
<code>\mathchar</code> . . . . . 274, 275	<code>\patchcmd</code> . . . . . 576, 577, 654, 655
<code>\mathclose</code> . . . . . 57, 75, 78, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 172, 229, 234, 244, 263, 268, 283, 295, 296, 375, 383, 442, 443, 453, 484, 485, 495	<code>\pb</code> . . . . . 15
<code>\mathcode</code> . . . . . 242, 246, 260	<code>\pdiagmat</code> . . . . . 9, 17, 594
<code>\mathop</code> . . . . . 787, 790, 799, 800	<code>\peval</code> . . . . . 14, 870
<code>\mathopen</code> . . . . . 57, 73, 76, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 172, 229, 234, 244, 263, 267, 283, 295, 296, 375, 383, 442, 443, 451, 484, 485, 493	phx internal commands:
<code>\mathord</code> . . . . . 789	<code>\__phx_define_key:nnnn</code> . . . . . . 12, 12, 24, 550, 551, 552, 553, 554, 555, 578, 580, 656, 658, 778, 779, 780
<code>\mathrel</code> . . . . . 74, 77, 243, 247, 425	<code>\l__phx_diagmat_tl</code> . . . . . . . . . . 582, 611, 632, 638, 643
<code>\mathrm</code> . . . . . 791, 792	<code>\__phx_diagmat_type:nnn</code> . . . . . . . . . . 40, 599, 600, 603, 603
<code>\MessageBreak</code> . . . . . 227, 371	<code>\__phx_gobble_i:n</code> . . . . . 111
<code>\mid</code> . . . . . 20, 53, 56	<code>\__phx_if_digits_only:n</code> . . . . . 687
<code>\middle</code> . . . . . 250	<code>\__phx_if_digits_only:nTF</code> . . . . . . . . . . 42, 687, 697, 711
<code>\mkern</code> . . . . . 792	<code>\__phx_if_digits_only_p:n</code> . . . . . 687
mode commands:	<code>\l__phx_mat_align_tl</code> . . . . . . . . . . 581, 582, 635, 642
<code>\mode_if_math:TF</code> . . . . . 836	<code>\l__phx_mat_diag_clist</code> . . . . . . . . . . 582, 606, 607, 618
<code>\mqty</code> . . . . . 17	<code>\l__phx_mat_dim_int</code> . . . . . . . . . . 582, 607, 608, 609, 612, 614, 624
msg commands:	<code>\l__phx_mat_empty_tl</code> . . . . . . . . . . 579, 582, 620, 623
<code>\msg_error:nn</code> . . . . . 845	
MT commands:	
<code>\MT_matrix_begin:N</code> 576, 577, 654, 655	
<b>N</b>	
<code>\NeedsTeXFormat</code> . . . . . 3	

<code>\l__phx_mat_line_tl</code>	882
. 582, 620, 621, 623, 626, 629, 632, 633	
<code>\__phx_mathvphantom:n</code>	883
. 79, 79, 108	
<code>\__phx_processkeyopt:n</code>	881, 882
. . . . . 12, 16, 26, 561, 588, 671, 782	
<code>\__phx_setkeys:nn</code>	880
. 12, 14, 25, 556, 781	
<code>\l__phx_tmpa_box</code>	879, 880
. . . . . 8, 81, 85, 86, 87, 91, 92, 93, 97, 98, 99, 103, 104, 105	
<code>\l__phx_tmpa_tl</code>	14
. 40, 109, 618, 619, 621	
<code>\l__phx_tmpb_tl</code>	
. . . . . 109	
<code>\__phx_xmat_entry_format:nnn</code>	
. 667, 683, 728, 732, 737, 756, 760, 765	
<code>\l__phx_xmat_extra_cdots_bool</code>	
. . . . . 660, 716, 719, 723, 734, 749, 762	
<code>\l__phx_xmat_extra_vdots_bool</code>	
. . . . . 660, 702, 705, 709, 741	
<code>\l__phx_xmat_showleft_int</code>	
. 659, 660, 685, 713, 715, 729, 744, 757	
<code>\l__phx_xmat_showtop_int</code>	
. . . . . 657, 660, 684, 699, 701, 725	
<code>\l__phx_xmat_tl</code>	
. . . . . 660, 695, 727, 731, 736, 739, 743, 746, 751, 753, 755, 759, 764, 769	
<code>\__phx_xmat_type:nnnnn</code>	
. . . . . 43, 677, 678, 692, 692	
prg commands:	
<code>\prg_new_conditional:Npnn</code>	687
<code>\prg_replicate:nn</code>	744
<code>\prg_return_false:</code>	690
<code>\prg_return_true:</code>	690
<code>\ProcessOptions</code>	18
<code>\protect</code>	219, 363, 544, 872
<code>\protected</code>	50, 51, 52, 53, 55, 56
<code>\ProvidesExplFile</code>	
. . . . . 548, 573, 651, 776, 817, 875	
<code>\ProvidesExplPackage</code>	6
<code>\ProvidesFile</code>	116, 222, 366, 852
<code>\PV</code>	11, 783
<code>\pv</code>	11, 783
<code>\pxmat</code>	10, 17
<b>Q</b>	
<code>\q</code>	12
<code>\qc</code>	882
<code>\qcc</code>	883
<code>\qcomma</code>	881, 882
<code>\qq</code>	880
<code>\qqtex</code>	879, 880
<code>\qty</code>	14
<code>\quad</code>	12, 878, 879, 881, 883, 893
<b>R</b>	
<code>\rangle</code>	4, 148, 149, 161, 168, 169, 217, 234, 235, 244, 248, 261, 267, 283, 295, 383, 385, 386, 410, 414, 419, 442, 447, 451, 484, 489, 493
<code>\rank</code>	783
<code>\rbrace</code>	4, 142, 143, 155, 162, 163, 215
<code>\Re</code>	11, 16, 45, 794
regex commands:	
<code>\regex_match:nnTF</code>	689
<code>\relax</code>	20, 25, 22, 38, 51, 52, 55, 195, 305, 324, 420, 437, 499, 518, 567
<code>\RequirePackage</code>	9, 575, 653, 803, 877
<code>\Res</code>	783
<code>\res</code>	783
<code>\Resymbol</code>	11, 16, 45, 797
<code>\right</code>	20, 23, 58, 65, 67, 126
<code>\romannumeral</code>	433
<b>S</b>	
<code>\scriptscriptstyle</code>	102
<code>\scriptstyle</code>	96
<code>\setcounter</code>	609
<code>\setkeys</code>	15
<code>\space</code>	429, 846
<code>\string</code>	24, 25, 129, 171, 173, 175, 177, 179, 181, 183, 186, 196, 199, 311, 423, 429, 471, 505, 846
<code>\symbf</code>	15, 841
<code>\symbfit</code>	15, 841
<code>\symoperators</code>	20, 50, 51, 52, 53, 271, 272
<b>T</b>	
TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:	
<code>\@current</code>	19
<code>\@currname</code>	19

<code>\@empty</code>	18, 45, 305, 306, 307, 324, 330, 331, 337, 420, 421, 422, 499, 500, 501, 518, 532, 538	<code>\phx@@ab@kb</code>	265, 277
<code>\@ifnextchar</code>	30, 33	<code>\phx@@end</code>	265, 277
<code>\@ifstar</code>	879, 883, 893	<code>\phx@amb@bk</code>	241, 251
<code>\@nnil</code>	38, 39, 46	<code>\phx@amb@kb</code>	259, 276
<code>\@onefilewithoptions</code>	40	<code>\phx@ab@C</code>	25
<code>\@onlypreamble</code>	29, 32, 35, 48	<code>\phx@AB@gen</code>	25, 26, 128, 194, 229, 230, 234, 235, 251, 252, 276, 277
<code>\@phx@abtight</code>	118, 121	<code>\phx@ab@ordersym</code>	855, 861
<code>\@phx@bk@argnum</code>	393, 400, 406, 433	<code>\phx@abb@bkv</code>	246, 250, 369
<code>\@phx@bk@do@pt</code>	420, 436, 437	<code>\phx@abb@l</code>	262, 266, 271, 274
<code>\@phx@bk@l</code>	395, 401, 409, 412, 416, 424, 461, 466, 472, 524, 529, 533	<code>\phx@abb@r</code>	262, 266, 272, 275
<code>\@phx@bk@m</code>	396, 402, 413, 417, 418, 425, 462, 467, 473, 525, 530, 534	<code>\phx@abc@close</code>	23, 29, 33, 121, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 170, 209, 230, 235, 248, 267, 268, 378, 386, 451, 453, 493, 495, 866
<code>\@phx@bk@r</code>	397, 403, 410, 414, 419, 426, 463, 468, 474, 526, 531, 535	<code>\phx@abopen</code>	23, 29, 33, 121, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 170, 209, 230, 235, 248, 267, 268, 378, 386, 450, 452, 492, 494, 866
<code>\@phx@dbl@c</code>	550, 568	<code>\phx@bk@do@pt</code>	420
<code>\@phx@dbl@d</code>	551, 569	<code>\phx@bk@doopt</code>	35, 394, 404, 420
<code>\@phx@dbl@oc</code>	554, 568	<code>\phx@bk@in@⟨n.roman⟩</code>	34
<code>\@phx@dbl@od</code>	555, 569	<code>\phx@bk@in@i</code>	408, 539
<code>\@phx@dbl@osc</code>	552, 568	<code>\phx@bk@in@iii</code>	408, 478
<code>\@phx@dbl@sd</code>	553, 569	<code>\phx@bk@in@iiii</code>	408, 541
<code>\@phx@ev</code>	316, 324, 330, 342, 352, 510, 518, 532, 533, 534, 535	<code>\phx@bra@@</code>	225
<code>\@phx@ev@basis</code>	318, 324, 331, 333, 334, 337, 347, 353, 512, 518, 538, 541	<code>\phx@d@l@genxa</code>	25, 28, 192, 201, 232, 237, 254, 279
<code>\@phx@ev@do@pt</code>	305, 322, 324, 499, 516, 518	<code>\phx@d@l@genxm</code>	25, 26, 28, 192, 200, 231, 236, 253, 278
<code>\@phx@oplega@PV</code>	779, 789	<code>\phx@d@l@geny</code>	28, 203, 858, 859, 869, 870, 871
<code>\@phx@oplega@pv</code>	780, 790	<code>\phx@d@lx</code>	23, 24, 27, 185, 202, 233, 238, 255, 280, 286, 288, 300, 302, 333, 334, 338, 341, 347, 351, 360
<code>\@pkgextension</code>	27, 41	<code>\phx@d@lx⟨ab⟩</code>	25
<code>\@optionlist</code>	19	<code>\phx@d@lx⟨mb⟩</code>	25
<code>\@unprocessedoptions</code>	22	<code>\phx@d@lxab</code>	24, 28, 200
<code>\bBigg@</code>	20, 60, 61, 62, 68, 69, 70	<code>\phx@d@lxmb</code>	24, 28, 200
<code>\define@key</code>	13	<code>\phx@date</code>	4, 117, 223, 367, 548, 573, 651, 776, 817, 853, 875
<code>\f@size</code>	567	<code>\phx@dbl@gen</code>	562, 568, 569
<code>\leavevmode@ifvmode</code>	64, 66		
<code>\n@space</code>	65, 67		
<code>\operator@font</code>	787, 790, 799, 800		
<code>\p@</code>	65, 67		
<code>\phx@@ab@bk</code>	245, 252		

<code>\phx@dblcurrf@size</code>	<a href="#">...</a>	<a href="#">564</a> , <a href="#">565</a> , <a href="#">567</a>
<code>\phx@define@key</code>		<a href="#">18</a> , <a href="#">22</a> , <a href="#">24</a> , <a href="#">118</a> , <a href="#">855</a>
<code>\phx@del</code>		<a href="#">170</a>
<code>\phx@ev@doopt</code>		<a href="#">323</a> , <a href="#">327</a> , <a href="#">517</a> , <a href="#">521</a>
<code>\phx@false</code>		<a href="#">11</a>
<code>\phx@FW@pti@ns</code>		<a href="#">34</a> , <a href="#">36</a> , <a href="#">48</a>
<code>\phx@FW@ptions</code>		<a href="#">31</a> , <a href="#">33</a> , <a href="#">35</a>
<code>\phx@FW@ptions</code>		<a href="#">27</a> , <a href="#">30</a> , <a href="#">32</a>
<code>\phx@mathvphantom</code>		<a href="#">21</a> , <a href="#">108</a> , <a href="#">267</a> , <a href="#">268</a> , <a href="#">451</a> , <a href="#">453</a> , <a href="#">493</a> , <a href="#">495</a>
<code>\phx@mb@()</code>		<a href="#">25</a>
<code>\phx@nl@nabla</code>		<a href="#">804</a> , <a href="#">809</a> , <a href="#">810</a> , <a href="#">811</a> , <a href="#">812</a>
<code>\phx@processkeyopt</code>		<a href="#">18</a> , <a href="#">22</a> , <a href="#">26</a> , <a href="#">120</a> , <a href="#">857</a>
<code>\phx@qtext@</code>		<a href="#">878</a> , <a href="#">879</a> , <a href="#">883</a> , <a href="#">893</a>
<code>\phx@requiremodule</code>		<a href="#">19</a> , <a href="#">27</a> , <a href="#">224</a> , <a href="#">368</a> , <a href="#">805</a> , <a href="#">854</a>
<code>\phx@reserveda</code>		<a href="#">778</a> , <a href="#">794</a>
<code>\phx@setkeys</code>		<a href="#">18</a> , <a href="#">22</a> , <a href="#">20</a> , <a href="#">25</a> , <a href="#">119</a> , <a href="#">856</a>
<code>\phx@temp</code>		<a href="#">18</a>
<code>\phx@tempa</code>		<a href="#">195</a> , <a href="#">196</a>
<code>\phx@true</code>		<a href="#">10</a> , <a href="#">121</a> , <a href="#">794</a>
<code>\phx@version</code>		<a href="#">5</a> , <a href="#">548</a> , <a href="#">573</a> , <a href="#">651</a> , <a href="#">776</a> , <a href="#">817</a> , <a href="#">875</a>
<code>\reserved@a</code>		<a href="#">18</a> , <a href="#">19</a> , <a href="#">20</a> , <a href="#">21</a> , <a href="#">45</a> , <a href="#">46</a> , <a href="#">47</a> , <a href="#">187</a> , <a href="#">189</a> , <a href="#">191</a> , <a href="#">306</a> , <a href="#">307</a> , <a href="#">315</a> , <a href="#">316</a> , <a href="#">319</a> , <a href="#">421</a> , <a href="#">422</a> , <a href="#">423</a> , <a href="#">424</a> , <a href="#">425</a> , <a href="#">426</a> , <a href="#">428</a> , <a href="#">433</a> , <a href="#">500</a> , <a href="#">501</a> , <a href="#">509</a> , <a href="#">510</a> , <a href="#">513</a>
<code>\reserved@b</code>		<a href="#">37</a> , <a href="#">43</a> , <a href="#">46</a>
<code>\z@</code>		<a href="#">565</a>
<code>\z@skip</code>		<a href="#">563</a>
<code>\zap@space</code>		<a href="#">45</a> , <a href="#">306</a> , <a href="#">421</a> , <a href="#">500</a>
tex commands:		
<code>\tex_mathchoice:D</code>		<a href="#">82</a>
<code>\text</code>		<a href="#">878</a>
<code>\textstyle</code>		<a href="#">90</a>
<code>\times</code>		<a href="#">9</a> , <a href="#">15</a> , <a href="#">558</a> , <a href="#">811</a>
tl commands:		
<code>\tl_gclear:N</code>		<a href="#">611</a> , <a href="#">633</a> , <a href="#">695</a>
<code>\tl_gput_right:Nn</code>		<a href="#">620</a> , <a href="#">621</a> , <a href="#">623</a> , <a href="#">626</a> , <a href="#">629</a> , <a href="#">632</a>
<code>\tl_gset:Nn</code>		<a href="#">579</a> , <a href="#">581</a>
<code>\tl_if_empty:NTF</code>		<a href="#">619</a> , <a href="#">635</a>
<code>\tl_if_head_eq_catcode:nNTF</code>		<a href="#">840</a>
<code>\tl_map_inline:nn</code>		<a href="#">838</a>
<code>\tl_new:N</code>		<a href="#">109</a> , <a href="#">110</a> , <a href="#">584</a> , <a href="#">585</a> , <a href="#">586</a> , <a href="#">587</a> , <a href="#">664</a>
<code>\tl_put_right:Nn</code>		<a href="#">727</a> , <a href="#">731</a> , <a href="#">736</a> , <a href="#">739</a> , <a href="#">743</a> , <a href="#">746</a> , <a href="#">751</a> , <a href="#">753</a> , <a href="#">755</a> , <a href="#">759</a> , <a href="#">764</a>
<code>\tl_use:N</code>		<a href="#">638</a> , <a href="#">643</a> , <a href="#">769</a>
token commands:		
<code>\c_math_subscript_token</code>		<a href="#">669</a>
<code>\Tr</code>		<a href="#">783</a>
<code>\tr</code>		<a href="#">783</a>
<b>U</b>		
<code>\Udelimiter</code>		<a href="#">50</a> , <a href="#">51</a> , <a href="#">52</a> , <a href="#">53</a>
<code>\Umathchar</code>		<a href="#">271</a> , <a href="#">272</a>
<code>\unexpanded</code>		<a href="#">40</a>
<code>\upe</code>		<a href="#">783</a>
<code>\usepackage</code>		<a href="#">3</a>
<code>\usephxmodu</code>		<a href="#">3</a> , <a href="#">14</a> , <a href="#">19</a> , <a href="#">27</a>
<b>V</b>		
<code>\va</code>		<a href="#">15</a>
<code>\Vab</code>		<a href="#">218</a>
<code>\vab</code>		<a href="#">216</a>
<code>\value</code>		<a href="#">608</a> , <a href="#">665</a> , <a href="#">666</a>
<code>\vb</code>		<a href="#">15</a>
<code>\vbox</code>		<a href="#">65</a> , <a href="#">67</a>
<code>\vcenter</code>		<a href="#">563</a>
<code>\Vdiagmat</code>		<a href="#">9</a> , <a href="#">17</a> , <a href="#">594</a>
<code>\vdiagmat</code>		<a href="#">9</a> , <a href="#">17</a> , <a href="#">594</a>
<code>\vdot</code>		<a href="#">15</a>
<code>\vdots</code>		<a href="#">743</a> , <a href="#">746</a> , <a href="#">751</a>
<code>\vec</code>		<a href="#">15</a>
<code>\Vert</code>		<a href="#">4</a> , <a href="#">20</a> , <a href="#">52</a> , <a href="#">55</a> , <a href="#">146</a> , <a href="#">147</a> , <a href="#">159</a> , <a href="#">166</a> , <a href="#">167</a> , <a href="#">218</a> , <a href="#">859</a>
<code>\vert</code>		<a href="#">4</a> , <a href="#">20</a> , <a href="#">51</a> , <a href="#">144</a> , <a href="#">145</a> , <a href="#">157</a> , <a href="#">164</a> , <a href="#">165</a> , <a href="#">216</a> , <a href="#">229</a> , <a href="#">230</a> , <a href="#">234</a> , <a href="#">235</a> , <a href="#">242</a> , <a href="#">250</a> , <a href="#">263</a> , <a href="#">267</a> , <a href="#">268</a> , <a href="#">283</a> , <a href="#">295</a> , <a href="#">296</a> , <a href="#">375</a> , <a href="#">377</a> , <a href="#">378</a> , <a href="#">383</a> , <a href="#">385</a> , <a href="#">386</a> , <a href="#">413</a> , <a href="#">417</a> , <a href="#">418</a> , <a href="#">442</a> , <a href="#">443</a> , <a href="#">447</a> , <a href="#">448</a> , <a href="#">450</a> , <a href="#">453</a> , <a href="#">484</a> , <a href="#">485</a> , <a href="#">489</a> , <a href="#">490</a> , <a href="#">492</a> , <a href="#">495</a> , <a href="#">858</a> , <a href="#">869</a> , <a href="#">870</a> , <a href="#">871</a>
<code>\vphantom</code>		<a href="#">21</a>
<code>\vu</code>		<a href="#">15</a>
<code>\Vxmat</code>		<a href="#">10</a> , <a href="#">17</a>

\vxmat .....	10, 17	\xmat .....	10, 17
	<b>X</b>		<b>Z</b>
\xdef .....	316, 318, 424, 425, 426, 433,		
	472, 473, 474, 510, 512, 533, 534, 535	\Z .....	689