

# Package `math-operator` v. 1.2a User Guide

Conrad Kosowsky

July 2025

`kosowsky.latex@gmail.com`

## Overview

The `math-operator` package defines control sequences for roughly one hundred and fifty math operators, including special functions, probability distributions, pure mathematical constructions, and a variant of `\overline`. The package also provides an interface for users to define new math operators similar to the `amsopn` package. New operators can be medium or bold weight, and they may be declared as `\mathord` or `\mathop` subformulas.

---

$\LaTeX$  users will no doubt be familiar with the control sequences that produce special functions and operators such as `\sin`, `\cos`, or `\sup`. However, the  $\LaTeX$  kernel defines only about 30 such commands, and many less common but still widely used special functions remain undefined as a result. The `math-operator` package addresses this situation by defining control sequences for some hundred and fifty special functions and operators, divided into nine groups, and the package also provides an interface to define even more. The first pages of this user guide describe how to use the package, and the rest of the document lists the control sequences in each group. For documentation of the package code, please see `math-operator_code.pdf`, which is included with the `math-operator` installation and is available on CTAN. I encourage users who are interested in this package to also consult the `amsopn` and `moremath` packages as they may be more useful for you.<sup>1</sup> Users who are looking specifically for operators in quantum mechanics should consult the `linop` and `phfqit` packages.<sup>2</sup>

Users can load `math-operator` with the standard `\usepackage` syntax, and for each operator group, the package defines either all or no control sequences from that group during loading. Each operator group corresponds to two optional package arguments—one argument means define the control sequences of that group, and the other argument means avoid doing so. Table 1 lists the nine operator groups and their corresponding arguments. For every group, the package argument to define control sequences is a shortened version of the group name, and the package argument to avoid doing so is the same keyword prefaced by `no-`. By default, `math-operator` defines all control sequences that appear later in this document.

Users who want to create their own operators or redefine the commands in this package

---

Acknowledgements: Thanks to Andrew Baker for pointing out a bug in a previous version of `math-operator`.

<sup>1</sup> $\LaTeX$ 3 Project and American Mathematical Society, “amsopn—Typeset mathematical operator names,” <https://ctan.org/pkg/amsopn>; Marcel Ilg, “moremath—Additional commands for typesetting maths,” <https://ctan.org/pkg/moremath>.

<sup>2</sup>Johannes Weytjens, “linop—Typeset linear operators as they appear in quantum theory or linear algebra,” <https://ctan.org/pkg/linop>; Philippe Faist, “phfqit—Macros for typesetting Quantum Information Theory,” <https://ctan.org/pkg/phfqit>.

**Table 1: Optional Arguments for `math-operator`**

Group	To define commands (default)	To avoid defining commands
Blackboard bold	<code>blackboard</code>	<code>no-blackboard</code>
Category theory	<code>category</code>	<code>no-category</code>
Jacobi elliptic functions	<code>jacobi</code>	<code>no-jacobi</code>
Linear algebra	<code>linear</code>	<code>no-linear</code>
The command <code>\overbar</code>	<code>overbar</code>	<code>no-overbar</code>
Probability distributions	<code>probability</code>	<code>no-probability</code>
Special functions	<code>special</code>	<code>no-special</code>
Standard math operators	<code>standard</code>	<code>no-standard</code>
Trigonometric functions	<code>trigonometry</code>	<code>no-trigonometry</code>

should use one of the four control sequences in Table 2. The entries in Table 2 should appear only in the document preamble, and their syntax is identical and looks like

```
\DeclareMathOperator<optional *>{<control sequence>}{<operator text>}
```

When you use one of these macros, `math-operator` defines the `<control sequence>` to produce `<operator text>` in math mode. The optional asterisk controls the placement of superscripts and subscripts. Without an asterisk (the default version of the command), any superscripts and subscripts will render normally, but with an asterisk, they will appear above and below the operator. For example, to make a control sequence `\erf` for the error function, the `sty` file for `math-operator` contains

```
\DeclareMathOperator{\erf}{erf}
```

The syntax and implementation of these macros is very similar to the `amsopn` package.

The entries of Table 2 differ in the appearance of the resulting operator. The commands in the first column produce operators with medium text, and the commands in the second column produce operators with bold text. The difference between the rows is more subtle and boils down to the automatic spacing before and after the operator.<sup>3</sup> The macros from the first row instruct `TEX` to treat the operator like an ordinary variable, so they are most appropriate for sets and categories. The macros from the second row instruct `TEX` to horizontally position the operator like a summation or integral sign, and they are appropriate for functions and probability distributions. But if you are not overly fastidious, for most uses of this package other than category theory, you will probably be fine to just use `\DeclareMathOperator`.

The macros in Table 2 will happily redefine any operator commands, but they will not overwrite other control sequences unless you specifically tell them to do so. The count variable `\operatordefmode` controls the package behavior in this regard as follows:

- Negative: redefine the control sequence
- 0: silently ignore (message written in the `log` file)
- 1: issue a warning and do not redefine
- 2 or greater: raise an error

---

<sup>3</sup>`TEX`'s eight classes of math subformulas are beyond the scope of this user guide, but in summary, the horizontal position of different characters in an equation depends on their math classes. See Donald Knuth, *The T<sub>E</sub>Xbook* (Addison Wesley, 1986), 170; David Salomon, *The Advanced T<sub>E</sub>Xbook* (Springer, 1995), 256–258.

Table 2: Commands to Define New Operators

	Medium weight	Bold weight
Treated as <code>\mathord</code>	<code>\DeclareMathText</code>	<code>\DeclareBoldMathText</code>
Treated as <code>\mathop</code>	<code>\DeclareMathOperator</code>	<code>\DeclareBoldMathOperator</code>

By default, `math-operator` sets `\operatordefmode` to 1, so you will see a warning on the terminal or console if you try to convert a control sequence that is already defined into a math operator. However, if you really want to redefine a control sequence to be a math operator, you can say

```
\operatordefmode=-1
```

before calling a command from Table 2.

One operator group warrants additional explanation. The package argument `overbar` corresponds to the single control sequence `\overbar`, which adds a horizontal line above a math subformula. The line will be wider than `\bar` but narrower than `\overline`, and the syntax is

```
\overbar<optional *>[<optional decimal>]{<math>}
```

The `<decimal>` should be between 0 and 1, and it controls the width of the overline. Specifically, `\overbar` typesets `<math>`, creates a horizontal line that is `<decimal>` times the width of the math subformula, and places the line above the typeset subformula. By default, `<decimal>` is 0.8. With an asterisk, `\overbar` positions the overline halfway over the subformula. For example, the code

```
\overbar*[0.9]{xyz}
```

will put an overline above `xyz` that is 90% the length of `xyz` and position it exactly halfway between the start of the `x` and the end of the `z`.

When `\overbar` does not have an asterisk (the default version of the command), the count variable `\overbaroffset` controls the horizontal placement of the line.<sup>4</sup> As is standard in T<sub>E</sub>X, this variable should take values between 0 and 1000, and `math-operator` divides `\overbaroffset` by 1000 to form a fraction. It then places the overline that fraction of the way across the top of the subformula. The default value is 800. For example, saying

```
\overbaroffset=0
```

will make all following `\overbar` lines appear completely on the left side of the subformula, and an asterisk is equivalent to setting `\overbaroffset` to 500.

Some operator names contain characters that are not letters, and `math-operator` provides three control sequences for non-letter characters: `\operatorhyphen` produces a hyphen, `\operatorssquared` produces a superscript 2, and `\operatorinverse` produces a superscript `-1`. If you use a Unicode font for the operator font, `math-operator` typesets these symbols in the operator font, and inside the bold commands from Table 2, they will be bold. Additionally, each command has a starred version that forces the characters to be bold no matter where they appear in an equation. If you do not use a Unicode font, the hyphen and

<sup>4</sup>Version 1.0 of this package used `\operatorbaroffset` instead of `\overbaroffset`. To preserve backwards compatibility, `\operatorbaroffset` still works for this purpose. But please don't use `\operatorbaroffset` because I may take it out in a few years.

the squared symbol will work the same way, but `\operatorinverse` instead typesets the contents of `\defaultinverse`, which is normally  $-1$ , without using the operator font and without using boldface in the starred version of the command.<sup>5</sup> If you are in this situation and are unsatisfied with the formatting of your  $-1$  expressions, I encourage you to redefine `\defaultinverse` to be more to your liking with `\def` or `\renewcommand`.

A word about superscripts: I implemented the commands from Table 2 in a way that  $\text{\TeX}$  treats operator names as subformulas. This means that any super or subscripts will always appear after or above or below the operator, even if you end the operator name with a super or subscript. For example, if you define an inverse projection map as

```
\DeclareMathOperator{\projinv}{proj\operatorinverse}
```

in your document preamble and then attach a subscript as in

```
\projinv_x
```

you will see  $\text{proj}^{-1}_x$  instead of  $\text{proj}_x^{-1}$ . If your operator name ends with a super or subscript, I recommend putting everything before the final super or subscript into the appropriate command from Table 2 and then defining a separate macro using `\def` or `\newcommand`. For example, it is better to define an inverse projection map as

```
\def\projinv{\proj\operatorinv}
```

rather than using `\DeclareMathOperator`. I took this approach in designing `math-operator`, so to redefine operators from this package that end in superscripts, you should use `\def` or `\renewcommand` instead of a control sequence from Table 2.

Finally, because it may redefine  $\backslash P$ , `math-operator` defines `\pilcrow` to typeset the ¶ symbol in text and math modes.

\*\*\*

## Blackboard Bold

Note: to use the blackboard-bold commands listed here, you must load a package that defines `\mathbb` such as `amssymb` or `mathfont`, and if you do not do so before using these control sequences, you'll get an error. I am assuming that `\mathbb` provides access to blackboard-bold letters.

<code>\N</code>	$\mathbb{N}$	Natural numbers
<code>\Z</code>	$\mathbb{Z}$	Integers
<code>\Q</code>	$\mathbb{Q}$	Rational numbers
<code>\R</code>	$\mathbb{R}$	Real numbers
<code>\C</code>	$\mathbb{C}$	Complex numbers
<code>\H</code>	$\mathbb{H}$	Quaternions (or half-plane) <sup>6</sup>
<code>\O</code>	$\mathbb{O}$	Octonions <sup>7</sup>

<sup>5</sup>This complication arises because the minus sign lives in different places depending on the encoding. In Unicode, the minus sign is U+2212, but that is not true for other font encodings in  $\text{\TeX}$ .

<sup>6</sup>In math mode only. Outside of equations, `\H` will still behave normally. If you want to change the `\H` operator somehow, you should redefine `\mathH`, not `\H`.

<sup>7</sup>In math mode only. Outside of equations, `\O` will still behave normally. If you want to change the `\O` operator somehow, you should redefine `\mathO`, not `\O`.

<code>\P</code>	$\mathbb{P}$	Probability
<code>\E</code>	$\mathbb{E}$	Expectation

## Categories

I am not a category theorist, and serious category theorists who use this package will undoubtedly want to define more categories in their own documents using `\DeclareBoldMathText`. If I missed any common categories that should be on this list, I am happy to expand it.

<code>\Ab</code>	<b>Ab</b>	Category of abelian groups
<code>\Alg</code>	<b>Alg</b>	Category of algebras
<code>\Cat</code>	<b>Cat</b>	Category of small categories
<code>\CRing</code>	<b>CRing</b>	Category of commutative rings
<code>\Field</code>	<b>Field</b>	Category of fields
<code>\FinGrp</code>	<b>FinGrp</b>	Category of finite groups
<code>\FinVect</code>	<b>FinVect</b>	Category of finite-dimensional vector spaces
<code>\Grp</code>	<b>Grp</b>	Category of groups
<code>\Haus</code>	<b>Haus</b>	Category of Hausdorff spaces
<code>\Man</code>	<b>Man</b>	Category of manifolds
<code>\Met</code>	<b>Met</b>	Category of metric spaces
<code>\Mod</code>	<b>Mod</b>	Category of modules
<code>\Mon</code>	<b>Mon</b>	Category of monoids
<code>\Ord</code>	<b>Ord</b>	Category of preordered sets
<code>\Ring</code>	<b>Ring</b>	Category of rings
<code>\Set</code>	<b>Set</b>	Category of sets
<code>\Top</code>	<b>Top</b>	Category of topological spaces
<code>\Vect</code>	<b>Vect</b>	Category of vector spaces
<code>\cocone</code>	cocone	Cocone
<code>\colim</code>	colim	Colimit
<code>\cone</code>	cone	Cone
<code>\op</code>	$\text{op}$	Opposite category

## Jacobi Elliptic Functions

Pretty straightforward. If you load `math-operator` with `jacobi`, you won't be able to use `\sc` to change to a small-caps font. (But you shouldn't use `\sc` anyway because it's deprecated.)

<code>\cd</code>	cd
<code>\cn</code>	cn
<code>\cs</code>	cs
<code>\dc</code>	dc
<code>\dn</code>	dn
<code>\ds</code>	ds
<code>\nc</code>	nc
<code>\nd</code>	nd

<code>\ns</code>	ns
<code>\sc</code>	sc
<code>\sd</code>	sd
<code>\sn</code>	sn

## Linear Algebra

Some matrix groups and operations.

<code>\adj</code>	adj	Adjugate matrix
<code>\coker</code>	coker	Cokernel
<code>\GL</code>	GL	General linear group
<code>\nullity</code>	nullity	Nullity
<code>\Orthogonal</code>	O	Orthogonal group
<code>\proj</code>	proj	Projection (onto a vector)
<code>\rank</code>	rank	Rank
<code>\SL</code>	SL	Special linear group
<code>\SO</code>	SO	Special orthogonal group
<code>\SU</code>	SU	Special unitary group
<code>\Sp</code>	Sp	Symplectic group
<code>\spanop</code>	span	Span
<code>\tr</code>	tr	Trace
<code>\T</code>	T	Transpose
<code>\Unitary</code>	U	Unitary group

## Overlining

Loading `math-operator` with the `overbar` option tells the package to define `\overbar`. Below are two examples of this macro with `\bar` and `\overline` for comparison.

<code>\bar a</code>	$\bar{a}$	<code>\bar X</code>	$\bar{X}$
<code>\overbar a</code>	$\overbar{a}$	<code>\overbar X</code>	$\overbar{X}$
<code>\overline a</code>	$\overline{a}$	<code>\overline X</code>	$\overline{X}$

## Probability Distributions

A selection of the most common probability distributions. For the normal distribution, if you type `\Normal` without the asterisk, you will see  $\mathcal{N}$ , and if you include the asterisk after `\Normal`, then `math-operator` will write out “Normal.”

<code>\Bernoulli</code>	Bernoulli
<code>\Betaop</code>	Beta
<code>\Binomial</code>	Binomial
<code>\Boltzmann</code>	Boltzmann
<code>\Burr</code>	Burr

<code>\Categorical</code>	Categorical	
<code>\Cauchy</code>	Cauchy	
<code>\ChiSq</code>	$\chi^2$	Chi-squared
<code>\Dagum</code>	Dagum	
<code>\Exponential</code>	Exponential	
<code>\Erlang</code>	Erlang	
<code>\Gamma</code>	Gamma	
<code>\Gompertz</code>	Gompertz	
<code>\InvChiSq</code>	Inv- $\chi^2$	Inverse chi-squared
<code>\InvGamma</code>	Inv-Gamma	Inverse gamma
<code>\Kolmogorov</code>	Kolmogorov	
<code>\LogLogistic</code>	Log-Logistic	
<code>\LogNormal</code>	Log-Normal	
<code>\Logistic</code>	Logistic	
<code>\Lomax</code>	Lomax	
<code>\MaxwellBoltzmann</code>	Maxwell-Boltzmann	
<code>\Multinomial</code>	Multinomial	
<code>\NegBinomial</code>	Neg-Binomial	Negative binomial
<code>\Normal<math>\langle</math><i>optional</i><math>\rangle</math>*</code>	$\mathcal{N}$ or Normal	
<code>\Pareto</code>	Pareto	
<code>\Poisson</code>	Poisson	
<code>\Weibull</code>	Weibull	
<code>\Zipf</code>	Zipf	

## Special Functions

Common special functions from applied math.

<code>\Ai</code>	Ai	Airy function of the first kind
<code>\Bi</code>	Bi	Airy function of the second kind
<code>\Ci</code>	Ci	Cosine integral function
<code>\ci</code>	ci	Cosine integral function (variant)
<code>\Chiop</code>	Chi	Hyperbolic cosine integral function
<code>\Ei</code>	Ei	Exponential integral function
<code>\erf</code>	erf	Error function
<code>\erfinv</code>	$\text{erf}^{-1}$	Inverse error function
<code>\erfc</code>	erfc	Complementary error function
<code>\erfcinv</code>	$\text{erfc}^{-1}$	Inverse complementary error function
<code>\Li</code>	Li	Polylogarithm function
<code>\li</code>	li	Logarithmic integral function
<code>\Log</code>	Log	Logarithm (principal value)
<code>\sgn</code>	sgn	Sign function
<code>\Si</code>	Si	Sine integral function
<code>\si</code>	si	Sine integral function (variant)
<code>\Shi</code>	Shi	Hyperbolic sine integral function

## Standard Operators

Common mathematical operations. More pure mathy than the special functions.

<code>\argmax</code>	arg max	Arguments of the maxima
<code>\argmin</code>	arg min	Arguments of the minima
<code>\Aut</code>	Aut	Automorphism group
<code>\c</code>	$\complement$	Complement <sup>8</sup>
<code>\cf</code>	cf	Cofinality
<code>\cl</code>	cl	Closure
<code>\conv</code>	conv	Convex hull
<code>\corr</code>	corr	Correlation
<code>\cov</code>	cov	Covariance
<code>\curl</code>	curl	Curl
<code>\divop</code>	div	Divergence
<code>\grad</code>	grad	Gradient
<code>\Hess</code>	$\mathcal{H}$	Hessian
<code>\Hom</code>	Hom	Collection of morphisms
<code>\id</code>	id	Identity
<code>\Im</code>	Im	Imaginary part
<code>\varIm</code>	$\Im$	Imaginary part <sup>9</sup>
<code>\img</code>	img	Image
<code>\Info</code>	$\mathcal{I}$	Fisher Information
<code>\interior</code>	int	Interior
<code>\lcm</code>	lcm	Least common multiple
<code>\Proj</code>	Proj	Projective spectrum
<code>\Re</code>	Re	Real part
<code>\varRe</code>	$\Re$	Real part <sup>10</sup>
<code>\Res</code>	Res	Residue
<code>\Spec</code>	Spec	Spectrum
<code>\supp</code>	supp	Support
<code>\Var</code>	Var	Variance

## Trigonometry

All inverse, hyperbolic, and inverse hyperbolic trigonometric functions that are not in the  $\LaTeX$  kernel.

<code>\csch</code>	csch	Hyperbolic cosecant
<code>\sech</code>	sech	Hyperbolic secant
<code>\arccsc</code>	arccsc	Inverse cosecant
<code>\arcsec</code>	arcsec	Inverse secant

<sup>8</sup>In math mode only. Outside of equations, `\c` will still behave normally. If you want to change the `\c` operator somehow, you should redefine `\mathc`, not `\c`.

<sup>9</sup>In the  $\LaTeX$  kernel, `\Im` produces  $\Im$ , but I decided to change that since Im is more standard than  $\Im$ .

<sup>10</sup>In the  $\LaTeX$  kernel, `\Re` produces  $\Re$ , but I decided to change that since Re is more standard than  $\Re$ .



<code>\arccot</code>	<code>arccot</code>	Inverse cotangent
<code>\arcsinh</code>	<code>arcsinh</code>	Inverse hyperbolic sine
<code>\arccosh</code>	<code>arccosh</code>	Inverse hyperbolic cosine
<code>\arctanh</code>	<code>arctanh</code>	Inverse hyperbolic tangent
<code>\arccsch</code>	<code>arccsch</code>	Inverse hyperbolic cosecant
<code>\arcsech</code>	<code>arcsech</code>	Inverse hyperbolic secant
<code>\arccoth</code>	<code>arccoth</code>	Inverse hyperbolic tangent
<code>\arsinh</code>	<code>arsinh</code>	Inverse hyperbolic sine
<code>\arcosh</code>	<code>arcosh</code>	Inverse hyperbolic cosine
<code>\artanh</code>	<code>artanh</code>	Inverse hyperbolic tangent
<code>\arcsch</code>	<code>arcsch</code>	Inverse hyperbolic cosecant
<code>\arsech</code>	<code>arsech</code>	Inverse hyperbolic secant
<code>\arcoth</code>	<code>arcoth</code>	Inverse hyperbolic cotangent
<code>\sininv</code>	<code>sin<sup>-1</sup></code>	Inverse sine
<code>\cosinv</code>	<code>cos<sup>-1</sup></code>	Inverse cosine
<code>\taninv</code>	<code>tan<sup>-1</sup></code>	Inverse tangent
<code>\cscinv</code>	<code>csc<sup>-1</sup></code>	Inverse cosecant
<code>\secinv</code>	<code>sec<sup>-1</sup></code>	Inverse secant
<code>\cotinv</code>	<code>cot<sup>-1</sup></code>	Inverse cotangent
<code>\sinhinv</code>	<code>sinh<sup>-1</sup></code>	Inverse hyperbolic sine
<code>\coshinv</code>	<code>cosh<sup>-1</sup></code>	Inverse hyperbolic cosine
<code>\tanhinv</code>	<code>tanh<sup>-1</sup></code>	Inverse hyperbolic tangent
<code>\cschinv</code>	<code>csch<sup>-1</sup></code>	Inverse hyperbolic cosecant
<code>\sechinv</code>	<code>sech<sup>-1</sup></code>	Inverse hyperbolic secant
<code>\cothinv</code>	<code>coth<sup>-1</sup></code>	Inverse hyperbolic cotangent