Package math-operator v. 1.3a User Guide

Conrad Kosowsky
September 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.[1] Users who are looking specifically for operators in quantum mechanics should consult the linop and phfqit packages.[2]

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 and suggesting additional control sequences for the package.

[1] LaTeX3 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.

[2] 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	`blackboard`	`no-blackboard`
Category theory	`category`	`no-category`
Jacobi elliptic functions	`jacobi`	`no-jacobi`
Linear algebra	`linear`	`no-linear`
The command `\overbar`	`overbar`	`no-overbar`
Probability distributions	`probability`	`no-probability`
Special functions	`special`	`no-special`
Standard math operators	`standard`	`no-standard`
Trigonometric functions	`trigonometry`	`no-trigonometry`

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.[3] 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 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

[3] 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 TEXbook* (Addison Wesley, 1986), 170; David Salomon, *The Advanced TEXbook* (Springer, 1995), 256–258.

Table 2: Commands to Define New Operators

	Medium weight	Bold weight
Treated as `\mathord`	`\DeclareMathText`	`\DeclareBoldMathText`
Treated as `\mathop`	`\DeclareMathOperator`	`\DeclareBoldMathOperator`

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.[4] As is standard in TEX, 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, `\operatorsquared` 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

[4] 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.[5] 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 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 $\mathrm{proj}^{-1}{}_x$ instead of 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 `\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.

`\N`	$\mathbb{N}$	Natural numbers
`\Z`	$\mathbb{Z}$	Integers
`\Q`	$\mathbb{Q}$	Rational numbers
`\R`	$\mathbb{R}$	Real numbers
`\C`	$\mathbb{C}$	Complex numbers
`\H`	$\mathbb{H}$	Quaternions (or half-plane)[6]
`\O`	$\mathbb{O}$	Octonions[7]

[5] 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 TEX.

[6] 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`.

[7] 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`.

\F	$\mathbb{F}$	Arbitrary field
\P	$\mathbb{P}$	Probability
\E	$\mathbb{E}$	Expectation

Categories

In version 1.3, I doubled the number of control sequences in math-operator for category theory, so this list is fairly substantial. That being said, it by no means exhaustive, and serious category theorists who use this package will likely need to define more categories using \DeclareBoldMathText.

\Ab	**Ab**	Category of abelian groups
\Alg	**Alg**	Category of algebras
\Bialg	**Bialg**	Category of bialgebras
\Cat	**Cat**	Category of small categories
\CGH	**CGH**	Compactly generated Hausdorff spaces
\CGWH	**CGWH**	Compactly generated weak Hausdorff spaces
\Coalg	**Coalg**	Category of coalgebras
\Comod	**Comod**	Category of comodules
\CRing	**CRing**	Category of commutative rings
\Field	**Field**	Category of fields
\FinGrp	**FinGrp**	Category of finite groups
\FinHilb	**FinHilb**	Category of finite-dimensional Hilbert spaces
\FinVect	**FinVect**	Category of finite-dimensional vector spaces
\FinSet	**FinSet**	Category of finite sets
\Frm	**Frm**	Category of frames
\Grp	**Grp**	Category of groups
\Haus	**Haus**	Category of Hausdorff spaces
\HeytAlg	**HeytAlg**	Category of Heyting algebras
\Hilb	**Hilb**	Category of Hilbert spaces
\HopfAlg	**HopfAlg**	Category of Hopf algebras
\Lat	**Lat**	Category of latices
\LieAlg	**LieAlg**	Category of Lie algebras
\LieGrp	**LieGrp**	Category of Lie groups
\Loc	**Loc**	Category of locales
\Man	**Man**	Category of manifolds
\Mat	**Mat**	Category of matrices
\Met	**Met**	Category of metric spaces
\Mod	**Mod**	Category of modules
\Mon	**Mon**	Category of monoids
\Ord	**Ord**	Category of preordered sets
\Pos	**Pos**	Category of posets
\Rel	**Rel**	Category of sets and binary relations
\Ring	**Ring**	Category of rings
\Set	**Set**	Category of sets

`\Top`	**Top**	Category of topological spaces
`\Topos`	**Topos**	Category of toposes
`\Vect`	**Vect**	Category of vector spaces
`\cocone`	cocone	Cocone
`\colim`	colim	Colimit
`\cone`	cone	Cone
`\Ho`	Ho	Homotopy category
`\Hom`	Hom	Collection of morphisms
`\Ob`	Ob	Collection of objects
`\op`	$^{\mathrm{op}}$	Opposite category
`\PSh`	PSh	Presheaf category
`\Sh`	Sh	Sheaf 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. Please use `\textsc` instead.)

`\cd`	cd
`\cn`	cn
`\cs`	cs
`\dc`	dc
`\dn`	dn
`\ds`	ds
`\nc`	nc
`\nd`	nd
`\ns`	ns
`\sc`	sc
`\sd`	sd
`\sn`	sn

Linear Algebra

Some matrix groups and operations.

`\adj`	adj	Adjugate matrix
`\Cl`	Cl	Clifford algebra
`\codim`	codim	Codimension
`\coker`	coker	Cokernel
`\GL`	GL	General linear group
`\nullity`	nullity	Nullity
`\Orthogonal`	O	Orthogonal group
`\Pin`	Pin	Pin group
`\proj`	proj	Projection (onto a vector)

`\rank`	rank	Rank
`\SL`	SL	Special linear group
`\SO`	SO	Special orthogonal group
`\SU`	SU	Special unitary group
`\Sp`	Sp	Symplectic group
`\spanop`	span	Span
`\Spin`	Spin	Spin group
`\tr`	tr	Trace
`\T`	$^{\mathrm{T}}$	Transpose
`\Unitary`	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.

`\bar a`	$\bar{a}$	`\bar X`	$\bar{X}$
`\overbar a`	$\overline{a}$	`\overbar X`	$\overline{X}$
`\overline a`	$\overline{a}$	`\overline X`	$\overline{X}$

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."

`\Bernoulli`	Bernoulli	
`\Betaop`	Beta	
`\Binomial`	Binomial	
`\Boltzmann`	Boltzmann	
`\Burr`	Burr	
`\Categorical`	Categorical	
`\Cauchy`	Cauchy	
`\ChiSq`	χ^2	Chi-squared
`\Dagum`	Dagum	
`\Exponential`	Exponential	
`\Erlang`	Erlang	
`\Gammaop`	Gamma	
`\Gompertz`	Gompertz	
`\InvChiSq`	Inv-χ^2	Inverse chi-squared
`\InvGamma`	Inv-Gamma	Inverse gamma
`\Kolmogorov`	Kolmogorov	
`\LogLogistic`	Log-Logistic	
`\LogNormal`	Log-Normal	
`\Logistic`	Logistic	

`\Lomax`	Lomax	
`\MaxwellBoltzmann`	Maxwell-Boltzmann	
`\Multinomial`	Multinomial	
`\NegBinomial`	Neg-Binomial	Negative binomial
`\Normal`⟨*optional* ∗⟩	$\mathcal{N}$ or Normal	
`\Pareto`	Pareto	
`\Poisson`	Poisson	
`\Weibull`	Weibull	
`\Zipf`	Zipf	

Special Functions

Common special functions from applied math.

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

Standard Operators

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

`\argmax`	$\arg\max$	Arguments of the maxima
`\argmin`	$\arg\min$	Arguments of the minima
`\Aut`	Aut	Automorphism group
`\c`	$^{\mathrm{c}}$	Complement[8]
`\cf`	cf	Cofinality
`\cl`	cl	Closure

[8] 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`.

`\conv`	conv	Convex hull
`\corr`	corr	Correlation
`\cov`	cov	Covariance
`\curl`	curl	Curl
`\divop`	div	Divergence
`\Ext`	Ext	Ext (extension) functor
`\Gal`	Gal	Galois group
`\grad`	grad	Gradient
`\Hess`	$\mathcal{H}$	Hessian
`\id`	id	Identity
`\Im`	Im	Imaginary part
`\varIm`	$\Im$	Imaginary part[9]
`\img`	img	Image
`\Info`	$\mathcal{I}$	Fisher Information
`\interior`	int	Interior
`\lcm`	lcm	Least common multiple
`\Proj`	Proj	Projective spectrum
`\Re`	Re	Real part
`\varRe`	$\Re$	Real part[10]
`\Res`	Res	Residue
`\Spec`	Spec	Spectrum
`\st`	st	Standard part (shadow) of a hyperreal number
`\supp`	supp	Support
`\Tor`	Tor	Tor (torsion) functor
`\Var`	Var	Variance

Trigonometry

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

`\csch`	csch	Hyperbolic cosecant
`\sech`	sech	Hyperbolic secant
`\arccsc`	arccsc	Inverse cosecant
`\arcsec`	arcsec	Inverse secant
`\arccot`	arccot	Inverse cotangent
`\arcsinh`	arcsinh	Inverse hyperbolic sine
`\arccosh`	arccosh	Inverse hyperbolic cosine
`\arctanh`	arctanh	Inverse hyperbolic tangent
`\arccsch`	arccsch	Inverse hyperbolic cosecant
`\arcsech`	arcsech	Inverse hyperbolic secant
`\arccoth`	arccoth	Inverse hyperbolic tangent

[9] In the LaTeX kernel, `\Im` produces $\Im$, but I decided to change that since Im is more standard than $\Im$.
[10] In the LaTeX kernel, `\Re` produces $\Re$, but I decided to change that since Re is more standard than $\Re$.

`\arsinh`	arsinh	Inverse hyperbolic sine
`\arcosh`	arcosh	Inverse hyperbolic cosine
`\artanh`	artanh	Inverse hyperbolic tangent
`\arcsch`	arcsch	Inverse hyperbolic cosecant
`\arsech`	arsech	Inverse hyperbolic secant
`\arcoth`	arcoth	Inverse hyperbolic cotangent
`\sininv`	$\sin^{-1}$	Inverse sine
`\cosinv`	$\cos^{-1}$	Inverse cosine
`\taninv`	$\tan^{-1}$	Inverse tangent
`\cscinv`	$\csc^{-1}$	Inverse cosecant
`\secinv`	$\sec^{-1}$	Inverse secant
`\cotinv`	$\cot^{-1}$	Inverse cotangent
`\sinhinv`	$\sinh^{-1}$	Inverse hyperbolic sine
`\coshinv`	$\cosh^{-1}$	Inverse hyperbolic cosine
`\tanhinv`	$\tanh^{-1}$	Inverse hyperbolic tangent
`\cschinv`	csch^{-1}	Inverse hyperbolic cosecant
`\sechinv`	sech^{-1}	Inverse hyperbolic secant
`\cothinv`	$\coth^{-1}$	Inverse hyperbolic cotangent