

MATHSEMANTICS.STY – SEMANTIC MATH COMMANDS

Ronny Bergmann

ronny.bergmann@ntnu.no

Department of Mathematical Sciences
NTNU, Trondheim, Norway.

Roland Herzog

roland.herzog@iwr.uni-heidelberg.de

Interdisciplinary Center for Scientific Computing
Heidelberg University, Germany.

May 10, 2022, v 1.0.1

CONTENTS

1	Introduction	1
2	Package Options	2
3	Required Packages	2
4	Syntax	3
4.1	Letters	3
4.2	Syntax Helpers	4
4.3	Spacing Helpers	5
5	Abbreviations	6
5.1	English	6
5.2	German	7
6	Names	8
7	Semantic Commands	11
8	Additional Semantics by Topic	17
8.1	Manifolds: <code>mathsemantics-manifolds.sty</code>	17
8.2	Optimization: <code>mathsemantics-optimization.sty</code>	24

1 INTRODUCTION

This package aims to provide semantic commands for ease of use in mathematics to see better *what* you semantically mean which should be distinct/split from *how* it is realised in \LaTeX .

The package is a spin-off and developed in the suite of packages from the former numapde-group in Chemnitz, see the original repository at <https://gitlab.hrz.tu-chemnitz.de/numapde-public/numapde-latex>.

Throughout this documentation most commands are directly illustrated by examples, which are both displayed as code (`</>` or `</>` for math examples) and its rendered result in \LaTeX (\odot). Two examples are

`</>` `\bbR` \odot \mathbb{R}

and

`</>` `\eg` \odot e. g.

The aim is to first ease the use of some often used letters and low-level formats like bold face letters `</>` `\bbR` \odot \mathbb{R} , but also to provide high level commands that make typing mathematics easier, for example using `</>` `\abs{\frac{1}{2}}` \odot $|\frac{1}{2}|$ and `</>` `\abs[Big]{\frac{1}{2}}` \odot $\left|\frac{1}{2}\right|$. This is the main goal in [Section 4](#) about syntactical commands for mathematics. A next more support/helping section about abbreviations and names is [Section 5](#).

The first main part on general semantic commands is [Section 7](#).

While all these are loaded by default. The next part, [Section 8](#), introduces semantic commands for specific topics. These are given in separate sub-packages and can be loaded if you work in this area and want to use the commands.

The package should be loaded late, since it might overwrite a few commands, currently most prominently `\d` which is overwritten by `cleveref` in case `minted` is loaded. So for more flexibility, there is the alternative command `\dInt`.

2 PACKAGE OPTIONS

shortbb use shorter notations for the blackboard-bold math letters `\C`, `\K`, `\N`, `\Q`, `\R`, `\Z`

3 REQUIRED PACKAGES

amssymb.sty defines mathematical symbol fonts

ifthen.sty facilitates the definition of conditional commands

ifxetex.sty provides a way to check if a document is being processed by $X_{\text{e}}\LaTeX$ and company

mathtools.sty provides lots of improvements for math typesetting (includes `amsmath.sty`)

xifthen.sty extends `ifthen.sty` by adding new boolean conditions

xparse.sty provides a high-level interface to define new commands






xspace.sty adds space depending on context

4 SYNTAX

The `mathsemantics-syntax.sty` package provides mainly symbols and short commands, which can be used in semantic definitions for ease of notation. They usually are rather simple commands without too many parameters.

4.1 LETTERS


ba. . . bz	lower-case bold-face letters <code>\br</code> , <code>\bf</code>  r, f
bA. . . bZ	upper-case bold-face letters <code>\bR</code> , <code>\bF</code>  R, F
balpha. . . bomega	lower-case bold-face Greek letters <code>\balpha</code> , <code>\boldeta</code>  α, η (the latter being an exception)
bAlpha. . . bOmega	upper-case bold-face Greek letters <code>\bGamma</code> , <code>\bDelta</code>  Γ, Δ
bnull	bold-face zero <code>\bnull</code>  0
bone	bold-face one <code>\bone</code>  1
cA. . . cZ	upper-case calligraphic letters <code>\cM</code> , <code>\cN</code>  M, N
fA. . . fZ	upper-case fraktur letters <code>\fM</code> , <code>\fN</code> , <code>\fX</code>  $\mathfrak{M}, \mathfrak{N}, \mathfrak{X}$
sA. . . sZ	upper-case script letters <code>\sM</code> , <code>\sN</code> , <code>\sX</code>  $\mathcal{M}, \mathcal{N}, \mathcal{X}$
va. . . vz	lower-case letters with a vector accent <code>\va</code> , <code>\vb</code>  \vec{a}, \vec{b}
vA. . . vZ	upper-case letters with a vector accent <code>\vA</code> , <code>\vB</code>  \vec{A}, \vec{B}
valpha. . . vomega	lower-case Greek letters with a vector accent <code>\valpha</code> , <code>\vbeta</code>  $\vec{\alpha}, \vec{\beta}$


vAlpha. . . vOmega	upper-case Greek letters with a vector accent <code></> \vGamma, \vDelta</code>  $\vec{\Gamma}, \vec{\Delta}$
vnull	vector zero <code></> \vnull</code>  $\vec{0}$
vone	vector one <code></> \vone</code>  $\vec{1}$
bbA, . . . , bbZ	blackboard-bold uppercase letters <code></> \bbc, \bbk, \bbn, \bbq, \bbr, \bbs, \bbz</code>  C, K, N, Q, R, S, Z use the package option <code>shortbb</code> to introduce <code></> \c, \k, \n, \q, \r, \z</code>  C, K, N, Q, R, Z if not already defined elsewhere (i. e. they are not redefined, only <i>provided</i>).


4.2 SYNTAX HELPERS


enclspacing provides spacing after the opening and before the closing delimiters for `\enclose`. This is by default set to be empty.

enclose is a command which encloses some content in scaled delimiters. It is meant as a helper to facilitate the definition of other commands. Its syntax is `\enclose[#1]{#2}{#3}{#4}`. The first (optional) argument is used to scale the delimiters to the standard amsmath sizes.¹ The second and fourth arguments specify the opening and closing delimiters, respectively. The third argument is the content to be enclosed.

`</> \enclose{[]}{\dfrac{1}{2}}{[]}`  $\left[\frac{1}{2}\right]$

`</> \enclose[Big][{\dfrac{1}{2}}]`  $\left[\frac{1}{2}\right]$

`</> \enclose[auto]{[]}{\dfrac{1}{2}}{[]}`  $\left[\frac{1}{2}\right]$

`</> \enclose[none]{[]}{\dfrac{1}{2}}{[]}`  $\frac{1}{2}$

Note 1. none is merely meant for testing when having arguments in brackets whether it is useful to omit them. You can also deactivate the absolute value vertical lines this way, so *use this option with care*.


Note 2. This command should normally be used only in the definition of other commands. For instance, `\abs` is using it internally. See `\paren` for the


¹big, Big, bigg, Bigg or auto, which uses left and right as well as none to easily deactivate brackets.

nicer command to use

enclspacingSet provides spacing before and after the center delimiter `\encloseSet`. This is by default set to `\,`.


encloseSet is a command which encloses some content in scaled delimiters. It is meant as a helper to facilitate the definition of other commands. Its syntax is `\encloseSet[#1]{#2}{#3}{#4}{#5}{#6}`. The first (optional) argument is used to scale the delimiters including the center one to the standard amsmath sizes.¹ The second and sixth arguments specify the opening and closing delimiters, respectively. The fourth argument specifies the center delimiter and The third and fifth argument are the content to be enclosed.

`</> \encloseSet[big]{\{x\in\mathbb{R}\}{|}{x>5}\}`  $\{x \in \mathbb{R} \mid x > 5\}$


`</> \encloseSet[auto]{\{x\in\mathbb{R}\}{|}{x>\dfrac{1}{2}}\}`  $\left\{x \in \mathbb{R} \mid x > \frac{1}{2}\right\}$

Note. This command should normally be used only in the definition of other commands. For instance, `\setDef` is using it internally.

paren is an alternative to `\enclose`, with a different ordering of arguments. Its syntax is `\paren[#1]{#2}{#3}{#4}`, which is simply mapped to `\enclose[#1]{#2}{#4}{#3}`.


`</> \paren[Big][\{ \}]{\dfrac{1}{2}}`  $\left[\frac{1}{2}\right]$


`</> \paren[Big][\dfrac{1}{2}]`  $\left[\frac{1}{2}\right]$


`</> \paren[auto][\{ \}]{\dfrac{1}{2}}`  $\left[\frac{1}{2}\right]$

4.3 SPACING HELPERS


clap complements the standard \LaTeX commands `\llap` and `\rlap`. These commands horizontally smash their arguments.

`</> Let us \llap{smash} something.`  ~~Let us smash something.~~


`</> Let us \clap{smash} something.`  ~~Let us smash something.~~

`</> Let us \rlap{smash} something.`  ~~Let us smash something.~~


mathllap corresponds to `\llap` in math mode.

`</> \sum_{\mathllap{1\le i\le j\le n}} X_{ij}`  $\sum_{1\le i\le j\le n} X_{ij}$


mathclap corresponds to `\clap` in math mode.


`</> \sum_{\mathclap{1\le i\le j\le n}} X_{ij}`  $\sum_{1\le i\le j\le n} X_{ij}$


mathrlap corresponds to `\rlap` in math mode.

`</> \sum_{\mathrlap{1\le i\le j\le n}} X_{ij}`  $\sum_{1\le i\le j\le n} X_{ij}$

mrep stands for *math replace* and it typesets an argument while reserving the space for another. Its syntax is `\mrep[#1]{#2}{#3}`. The first (optional) argument is one of {l, c, r} and it is used to define the alignment. c is the default.

`</> \mrep[l]{1}{-1}-1`  1 - 1


`</> \mrep[c]{1}{-1}-1`  1 - 1


`</> \mrep[r]{1}{-1}-1`  1 - 1

5 ABBREVIATIONS

Abbreviations and their spacing might be difficult to generate


abrv write the letters provided as an abbreviation


`</> \abrv{ie}`  i. e.


`</> \abrv{wlog}`  w. l. o. g.









And we introduce a few of them specifically also a few that are not following this scheme.

5.1 ENGLISH














ala almost all `</> \ala`  a. a.

ale almost everywhere `</> \ale`  a. e.

cf conferatur (compare (to)) `</> \cf`  cf.

eg	exempli gratia (for example) <code></> \eg</code>  e. g.
Eg	Exempli gratia (for example, at beginning of a sentence) <code></> \Eg</code>  E. g.
etc	et cetera (and so on) <code></> \etc</code>  etc.
ie	id est (id est) <code></> \ie</code>  i. e.
iid	independent and identically distributed <code></> \iid</code>  i. i. d.
spd	symmetric positive definite <code></> \spd</code>  s. p. d.
suth	such that or subject to <code></> \suth</code>  s. t.
wrt	with respect to <code></> \wrt</code>  w. r. t.

5.2 GERMAN



bspw	beispielsweise (for example) <code></> \bspw</code>  bspw.
bzgl	bezüglich (with regard to) <code></> \bzgl</code>  bzgl.
bzw	beziehungsweise (respectively) <code></> \bzw</code>  bzw.
Dah	Das heißt (That is, beginning of phrase) <code></> \Dah</code>  D. h.
dah	das heißt (that is) <code></> \dah</code>  d. h.
evtl	eventuell (possibly) <code></> \evtl</code>  evtl.
fs	fast sicher <code></> \fs</code>  f. s.
fue	fast überall <code></> \fue</code>  f. ü.
IA	Im Allgemeinen (beginning of phrase) <code></> \IA</code>  I. A.
iA	im Allgemeinen <code></> \iA</code>  i. A.
idR	in der Regel <code></> \idR</code>  i. d. R.
IdR	In der Regel (beginning of phrase) <code></> \IdR</code>  I. d. R.
iW	im Wesentlichen <code></> \iW</code>  i. W.




IW	Im Wesentlichen (beginning of phrase) <code></></code> <code>\IW</code>  I. W.
mE	meines Erachtens <code></></code> <code>\mE</code>  m. E.
oBdA	ohne Beschränkung der Allgemeinheit <code></></code> <code>\oBdA</code>  o. B. d. A.
OBdA	ohne Beschränkung der Allgemeinheit (beginning of phrase) <code></></code> <code>\OBdA</code>  O. b. d. A.
oge	oben genannt <code></></code> <code>\oge</code>  o. g.
oae	oder ähnliche <code></></code> <code>\oae</code>  o. ä.
sio	siehe oben <code></></code> <code>\sio</code>  s. o.
ua	unter anderem <code></></code> <code>\ua</code>  u. a.
Ua	Unter anderem (beginning of phrase) <code></></code> <code>\Ua</code>  U. a.
ug	unten genannt <code></></code> <code>\ug</code>  u. g.
usw	und so weiter (and so on)  usw.
uU	unter Umständen <code></></code> <code>\uU</code>  u. U.
UnU	Unter Umständen (beginning of phrase) <code></></code> <code>\UnU</code>  U. U.
vgl	vergleiche (compare) <code></></code> <code>\vgl</code>  vgl.
zB	zum Beispiel <code></></code> <code>\zB</code>  z. B.
ZB	Zum Beispiel (beginning of phrase) <code></></code> <code>\ZB</code>  Z. B.
zHd	zu Händen <code></></code> <code>\zHd</code>  z. Hd.

6 NAMES

adimat	 ADIMAT
ampl	 AMPL
BibTeX	 BIBTEX













BibLaTeX	👁 BIBLATEX
cg	👁 CG
cpp	👁 C++
cppmat	👁 CPPMAT
dolfin	👁 DOLFIN
dolfinplot	👁 DOLFIN-PLOT
dolfinadjoint	👁 DOLFIN-ADJOINT
doxygen	👁 DOXYGEN
femorph	👁 FEMORPH
fenics	👁 FENICS
ffc	👁 FFC
fmg	👁 FMG
fortran	👁 FORTRAN
gitlab	👁 GITLAB
gmres	👁 GMRES
gmsb	👁 GMSH
ipopt	👁 IPOPT
libsvm	👁 LIBSVM
liblinear	👁 LIBLINEAR
macmpec	👁 MACMPEC
manifoldsjl	👁 MANIFOLDS.JL
manopt	👁 MANOPT
manoptjl	👁 MANOPT.JL
mathematica	👁 MATHEMATICA

matlab	 MATLAB
maple	 MAPLE
maxima	 MAXIMA
metis	 METIS
minres	 MINRES
mshr	 MSHR
mvirt	 MVIRT
numpy	 NUMPY
paraview	 PARAVIEW
pdflatex	 PDF LA T _E X
perl	 PERL
petsc	 PETSc
pymat	 PYMAT
python	 PYTHON
scikit	 SciKIT
scikitlearn	 SciKIT-LEARN
scipy	 SciPy
sphinx	 SPHINX
subgmres	 SUBGMRES
subminres	 SUBMINRES
superlu	 SUPERLU
svmlight	 SVM ^{LIGHT}
tritetmesh	 TriTETMESH
ufl	 UFL
















uqlab	 UQLAB
viper	 VIPER
xml	 XML















7 SEMANTIC COMMANDS

Build upon Syntax from [Section 4](#) this part provides semantic mathematical commands.

abs	absolute value. Its syntax is <code>\abs[#1]{#2}</code> . The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the argument.
	<code>\abs{a}</code>  $ a $
	<code>\abs[Big]{\dfrac{1}{2}}</code>  $\left \frac{1}{2}\right $
	<code>\abs[auto]{\dfrac{1}{2}}</code>  $\left \frac{1}{2}\right $
aff	affine hull <code>\aff</code>  aff
arcosh	area hyperbolic cosine <code>\arcosh</code>  arcosh
arcoth	area hyperbolic cotangens <code>\arcoth</code>  arcoth
argmax	maximizer of a function <code>\argmax_{x \in \mathbb{R}} f(x)</code>  $\arg \max_{x \in \mathbb{R}} f(x)$
Argmax	set of maximizers of a function <code>\Argmax_{x \in \mathbb{R}} f(x)</code>  $\text{Arg} \max_{x \in \mathbb{R}} f(x)$
argmin	minimizer of a function <code>\argmin_{x \in \mathbb{R}} f(x)</code>  $\arg \min_{x \in \mathbb{R}} f(x)$
Argmin	set of minimizers of a function <code>\Argmin_{x \in \mathbb{R}} f(x)</code>  $\text{Arg} \min_{x \in \mathbb{R}} f(x)$
arsinh	area hyperbolic cotangens <code>\arsinh</code>  arsinh
artanh	area hyperbolic tangens <code>\artanh</code>  artanh

bdiv	bold (meaning: vector) divergence of a matrix-valued function <code>\bdiv</code> div
ceil	integer larger or equal to input. Its syntax is <code>\ceil[#1]{#2}</code> . The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the argument. <code>\ceil{a}</code> $[a]$ <code>\ceil[Big]{\dfrac{1}{2}}</code> $\left[\frac{1}{2}\right]$
clconv	closure of the convex hull of a set <code>\clconv M</code> $\overline{\text{conv } M}$
closure	closure of a set <code>\closure M</code> $\text{cl } M$
cofac	cofactor matrix <code>\cofac(A)</code> $\text{cof}(A)$
compactly	compact embedding of topological spaces <code>\compactly</code> \hookrightarrow
cone	conic hull <code>\cone</code> cone
conv	convex hull of a set <code>\conv M</code> $\text{conv } M$
corresponds	binary operator for correspondence <code>A\corresponds B</code> $A \hat{=} B$
cov	covariance <code>\cov</code> Cov
curl	the curl operator <code>\curl</code> curl
d, dInt	integral symbol with prepended space, as in <code>\int_{\mathbb{R}} \exp(-x^2) \, \text{d } x</code> $\int_{\mathbb{R}} \exp(-x^2) \, dx$ Since <code>\text{d}</code> is often overridden, <code>\text{dInt}</code> is the safe alternative
dev	deviator of a matrix <code>\dev A</code> $\text{dev } A$
diag	diagonal matrix composed of entries in a vector, or diagonal of a matrix <code>\diag(a)</code> $\text{diag}(a)$ <code>\diag(A)</code> $\text{diag}(A)$
diam	diameter <code>\diam(M)</code> $\text{diam}(M)$

distOp	the mathematical operator denoting the distance
	<code>\distOp</code>  dist
dist	distance from a point to a set. Its syntax is <code>\dist[#1][#2]{#3}</code> or <code>\dist[#1]{#2}</code> . The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes. ¹ The second argument denotes the set. The third argument denotes the point; it can be omitted. The command <code>\distOp</code> is used to typeset the operator.
	<code>\dist[Big]{\cC}{\dfrac{x}{2}}</code>  $\text{dist}_C\left(\frac{x}{2}\right)$
	<code>\dist{\cC}</code>  dist_C
	<code>\dist</code>  dist
div	divergence <code>\div</code>  div
Div	(row-wise) divergence <code>\Div</code>  Div
dom	domain <code>\dom</code>  dom
dotcup	distinct union <code>\dotcup</code>  \cup
dprod	double contraction of matrices $A : B = \sum_{i,j} A_{ij}B_{ij} = \text{trace}(A^T B)$
	<code>A \dprod B</code>  $A : B$
dual	duality pairing. Its syntax is <code>\dual[#1][#2]{#3}</code> . The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the first factor. The third argument denotes the second factor.
	<code>\dual{x^*}{x}</code>  $\langle x^*, x \rangle$
	<code>\dual[Big]{x^*}{\dfrac{1}{2}}</code>  $\left\langle x^*, \frac{1}{2} \right\rangle$
e	Euler's number <code>\e</code>  e
embed	embedding of topological spaces <code>\embed</code>  \hookrightarrow
embeds	synonym of <code>\embed</code> <code>\embeds</code>  \hookrightarrow
epi	epigraph <code>\epi</code>  epi

eR	extended real line $\langle /> \backslash eR = \backslash bbR \ \backslash cup \ \{\pm \infty\}$  $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$
essinf	essential infimum $\langle /> \displaystyle \backslash essinf_{x \in \backslash bbR} f(x)$  $\operatorname{ess\,inf}_{x \in \mathbb{R}} f(x)$
esssup	essential supremum $\langle /> \displaystyle \backslash esssup_{x \in \backslash bbR} f(x)$  $\operatorname{ess\,sup}_{x \in \mathbb{R}} f(x)$
file	typesets a file name (using <code>noLinkurl</code>) $\langle /> \backslash file\{test.txt\}$  <code>test.txt</code>
floor	integer less or equal to input. Its syntax is <code>\floor[#1]{#2}</code> . The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the argument. $\langle /> \backslash floor\{a\}$  $\lfloor a \rfloor$ $\langle /> \backslash floor[Big]\{\backslash dfrac{1}{2}\}$  $\left\lfloor \frac{1}{2} \right\rfloor$
grad	gradient (of a function) $\langle /> \backslash grad F$  $\operatorname{grad} F$
Graph	graph of a function $\langle /> \backslash Graph$  Graph
id	identity operator $\langle /> \backslash id$  id
image	image of a function $\langle /> \backslash image$  image
inj	injectivity (radius) $\langle /> \backslash inj$  inj
inner	inner product. Its syntax is <code>\inner[#1]{#2}{#3}</code> . The first (optional) argument is used to scale the parentheses enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the first factor. The third argument denotes the second factor. $\langle /> \backslash inner\{a\}\{b\}$  (a, b) $\langle /> \backslash inner[Big]\{a\}\{\backslash dfrac{b}{2}\}$  $\left(a, \frac{b}{2}\right)$
interior	$\langle /> \backslash interior$  int

jump jump of a quantity, e. g., across a finite element facet. Its syntax is `\jump[#1]{#2}`. The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes.¹ The second argument denotes the argument.

`</> \jump{a}` $\llbracket a \rrbracket$

`</> \jump[Big]{\dfrac{1}{2}}` $\left\| \frac{1}{2} \right\|$

laplace the Laplace operator `</> \laplace u` Δu

lin linear hull of a set of vectors `</> \lin\{v_1, v_2\}` $\operatorname{lin}\{v_1, v_2\}$

norm norm of a vector. Its syntax is `\norm[#1]{#2}`. The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes.¹ The second argument denotes the argument.

`</> \norm{a}` $\|a\|$

`</> \norm[Big]{\dfrac{c}{2}}` $\left\| \frac{c}{2} \right\|$

`</> \norm[auto]{\dfrac{c}{2}}` $\left\| \frac{c}{2} \right\|$

projOp the mathematical operator denoting the projection `</> \projOp` proj

`</> \projOp` proj

proj projection onto a set. Its syntax is `\proj[#1]{#2}(#3)` or `\proj[#1]{#2}`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ The second argument denotes the set and can also be left out. The third argument denotes the point; it can be omitted. The command `\projOp` is used to typeset the operator.











`</> \proj` proj

`</> \proj(x)` $\operatorname{proj}(x)$

`</> \proj{\mathbb{C}}` $\operatorname{proj}_{\mathbb{C}}$

`</> \proj{\mathbb{C}}(x)` $\operatorname{proj}_{\mathbb{C}}(x)$

`</> \proj[Big]{\dfrac{x}{2}}` $\operatorname{proj}\left(\frac{x}{2}\right)$

	<code></> \proj[Big]{\cC}(\dfrac{x}{2})</code>  $\text{proj}_C\left(\frac{x}{2}\right)$
proxOp	the mathematical operator denoting the proximal map
	<code></> \proxOp</code>  prox
prox	the proximal operator of a function. Its syntax is <code>\prox[#1]{#2}(#3)</code> or <code>\prox[#1]{#2}</code> . The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes. ¹ The second argument denotes the set. The third argument denotes the point; it can be omitted. The command <code>\proxOp</code> is used to typeset the operator.
	<code></> \prox</code>  prox
	<code></> \prox{\lambda F}</code>  $\text{prox}_{\lambda F}$
	<code></> \prox{\lambda F}(x)</code>  $\text{prox}_{\lambda F}(x)$
	<code></> \prox[auto]{\lambda F}(\dfrac{x}{2})</code>  $\text{prox}_{\lambda F}\left(\frac{x}{2}\right)$
rank	rank (of a matrix) <code></> \rank</code>  rank
range	range of some operator <code></> \range</code>  range
restr	restriction/evaluation. Its syntax is <code>\restr[#1]{#2}{#3}</code> . The first (optional) argument is used to scale the delimiters enclosing the arguments to the standard amsmath sizes. ¹ The second argument denotes the argument to be restricted/evaluated. The third argument denotes the restriction set/evaluation point.
	<code></> \restr[auto]{\frac{d}{d t}}(f \circ \gamma)(t){t=0}</code>  $\left. \frac{d}{dt}(f \circ \gamma)(t) \right _{t=0}$
ri	relative interior <code></> \ri</code>  ri
setDef	define a set, where <code>\setMid</code> serves as the center divider. Its syntax is <code>\setDef[#1]{#2}{#3}</code> . The first (optional) argument is used to scale the parentheses enclosing the argument and the center divider to the standard amsmath sizes. ¹ The second argument denotes the left part of the definition, naming the potential elements of the set being defined. The third argument denotes the condition to include the elements in the set.
	<code></> \setDef{x \in \bbR}{x > 5}</code>  $\{x \in \mathbb{R} \mid x > 5\}$

	<code>\setDef[Big]{x\in\bbR}{x>\dfrac{1}{2}}</code>	$\left\{x \in \mathbb{R} \mid x > \frac{1}{2}\right\}$
setMid	divider within <code>\setDef</code> (set definitions). This defaults to <code>\setMid</code> .	
sgn	sign <code>\sgn</code> sgn	
Sgn	sign (set valued) <code>\Sgn</code> Sgn	
supp	support (of a function) <code>\supp F</code> $\text{supp } F$	
sym	symmetric part (of a matrix) <code>\sym A</code> $\text{sym } A$	
trace	trace (of a matrix) <code>\trace A</code> $\text{trace } A$	
transp	transpose of a vector or matrix.	
	<code>A^\transp</code> A^T	
transposeSymbol	symbol to use for the transpose	
	<code>\transposeSymbol</code> T	
var	variance <code>\var</code> Var	
weakly	weak convergence of a sequence <code>\weakly</code> \rightharpoonup	
weaklystar	weak star convergence of a sequence <code>\weaklystar</code> $\overset{*}{\rightharpoonup}$	

8 ADDITIONAL SEMANTICS BY TOPIC


While semantic commands might be suitable for all mathematical topics, the following subsections collect commands which are most useful in one particular mathematical area and hence might clutter the general semantic file. Any semantic topic files should always build on `mathsemantics-semantic.sty`.

8.1 MANIFOLDS: `mathsemantics-manifolds.sty`

The semantic file `mathsemantics-manifolds.sty` collects definitions and notations for Riemannian manifolds.

bitangentSpace the bi tangent space. Its syntax is `\bitangent{#1}[#2]`. The first argument denotes the base point. The second (optional) argument denotes the manifold, which defaults to \mathcal{M} .

`</>` `\bitangentSpace{p}`  $\mathcal{T}_p^{**}\mathcal{M}$


`</>` `\bitangentSpace{q}[\backslash\mathcal{N}]`  $\mathcal{T}_q^{**}\mathcal{N}$

bitangentSpaceSymbol the symbol used within `\bitangentSpace`.

`</>` `\bitangentSpaceSymbol`  \mathcal{T}^{**}

cotangentSpace the cotangent space. Its syntax is `\cotangentSpace{#1}[#2]`. The first argument denotes the base point. The second (optional) argument denotes the manifold, which defaults to \mathcal{M} .

`</>` `\cotangentSpace{p}`  $\mathcal{T}_p^*\mathcal{M}$

`</>` `\cotangentSpace{q}[\backslash\mathcal{N}]`  $\mathcal{T}_q^*\mathcal{N}$

cotangentBundle the cotangent bundle. Its syntax is `\cotangentBundle[#1]`. The (optional) argument denotes the manifold, which defaults to \mathcal{M} .


`</>` `\cotangentBundle`  $\mathcal{T}^*\mathcal{M}$

`</>` `\cotangentBundle[\backslash\mathcal{N}]`  $\mathcal{T}^*\mathcal{N}$


cotangentSpaceSymbol the symbol used within `\cotangent`.

`</>` `\cotangentSpaceSymbol`  \mathcal{T}^*

covariantDerivative is the covariant derivative. Its syntax is `\covariantDerivative{#1}[#2]`. The first argument is the vector (or vector field) determining the direction of differentiation. The second (optional) argument denotes the tensor field being differentiated.

`</>` `\covariantDerivative{X}{Y}`  $D_X Y$

covariantDerivativeSymbol the symbol used for the covariant derivative `\covariantDerivative`.

`</>` `\covariantDerivativeSymbol`  D

exponential the exponential map. Its syntax is `\exponential[#1][#2](#3)`. The first argument can be used to scale the third. The second argument denotes the base point and is mandatory. The third argument denotes the tangent vector,

which is optional, but if provided, the argument is put in brackets. The first following example illustrates the case, where no brackets are put. Note that the space is mandatory.

`</> \exponential{p}X` $\exp_p X$

`</> \exponential{p}(X)` $\exp_p(X)$

`</> \exponential[Big]{p}(\frac{X}{2})` $\exp_p\left(\frac{X}{2}\right)$

expOp

the symbol used within the `\exponential`.

`</> \expOp` \exp

geodesic

a geodesic. Its syntax is `\geodesic#1-;#2;[#3]-#4"-#5"(#6)-`. The first argument can be used to use a different symbol (locally) for the geodesic. The second (optional) argument is used to modify the style of the geodesic (symbol, long, arc or plain, where the last is the default). The third (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ It is ignored when the sixth argument is not given. The fourth argument denotes the initial point (at $t = 0$). The fifth argument denotes either the final point (at $t = 1$) for types `l` and `a`, or the initial tangent vector for type `p`. The sixth (optional) argument denotes the evaluation point. The command `\geodesicSymbol` is used to typeset the geodesic symbol default (i.e. globally)

`</> \geodesic<s>` γ

`</> \geodesic<s>(t)` $\gamma(t)$

`</> \geodesic<l>{p}{q}` $\gamma(\cdot; p, q)$

`</> \geodesic<l>{p}{q}(t)` $\gamma(t; p, q)$

`</> \geodesic<a>{p}{q}` $\widehat{\gamma}_{p,q}$

`</> \geodesic<a>[Big]{p}{q}(\frac{t}{2})` $\widehat{\gamma}_{p,q}\left(\frac{t}{2}\right)$

`</> \geodesic<p>{p}{X}` $\gamma_{p,X}$

`</> \geodesic<p>{p}{X}(t)` $\gamma_{p,X}(t)$

`</> \geodesic<p>[Big]{p}{X}(\frac{t}{2})` $\gamma_{p,X}\left(\frac{t}{2}\right)$

	$\backslash\text{geodesic}[\text{big}]{p}\{X\}((1-t)t)$ $\gamma_{p,X}((1-t)t)$
	$\backslash\text{geodesic} \dot{\gamma} {p}\{X\}(t)$ $\dot{\gamma}_{p,X}(t)$
geodesicSymbol	symbol to use for the geodesic in <code>\geodesic</code>
	$\backslash\text{geodesicSymbol}$ γ
inverseRetract	use an inverse retraction, the arguments are similar to <code>\logarithm</code> but use the <code>\retractionSymbol</code>
	$\backslash\text{inverseRetract}\{p\}q$ $\text{retr}_p^{-1}q$
	$\backslash\text{inverseRetract}\{p\}(q)$ $\text{retr}_p^{-1}(q)$
	$\backslash\text{inverseRetract}[\text{Big}]\{p\}(q)$ $\text{retr}_p^{-1}(q)$
logarithm	the logarithmic map. Its syntax is <code>\logarithm[#1]{#2}(<#3)</code> . The first argument can be used to scale the third. The second argument denotes the base point and is mandatory. The third argument denotes another point, which is optional, but if provided, the argument is put in brackets. The first following example illustrates the case, where no brackets are put. Note that the space is mandatory.
	$\backslash\text{logarithm}\{p\}q$ $\log_p q$
	$\backslash\text{logarithm}\{p\}(q)$ $\log_p (q)$
	$\backslash\text{logarithm}[\text{Big}]\{p\}(q)$ $\log_p (q)$
logOp	the symbol used within the <code>\logarithm</code> .
	$\backslash\text{logOp}$ \log
parallelTransport	the parallel transport. Its syntax is <code>\parallelTransport[#1]{#2}{#3}(<#4)<#5</code> . The first (optional) argument is used to scale the parentheses enclosing the argument #4. ¹ The second argument is the start point of parallel transport on a manifold. The third argument is the end point of parallel transport on a manifold. The fourth (optional) argument is the tangent vector that is transported. Putting it in brackets enables the scaling by the first argument. The fifth (optional) argument specifies an exponent, for example to parallel transport along a curve c

`\parallelTransport{p}{q}X` $P_{q \leftarrow p}X$

`\parallelTransport{p}{q}(X)` $P_{q \leftarrow p}(X)$

`\parallelTransport[big]{p}{q}(X)` $P_{q \leftarrow p}(X)$

`\parallelTransport{p}{q}(X)[c]` $P_{q \leftarrow p}^c(X)$

`\parallelTransport{p}{q}[c]` $P_{q \leftarrow p}^c$

parallelTransportDir similar to `\parallelTransport`, but the third argument is a direction to transport into. This can be rewritten to the classical notation applying an exponential map from the base point (#2) to th direction (#3). The fifth (optional) argument specifies an exponent, for example to parallel transport along a curve c

`\parallelTransportDir{p}{Y}X` $P_{p,Y}X$

`\parallelTransportDir{p}{Y}(X)` $P_{p,Y}(X)$

`\parallelTransportDir[big]{p}{Y}(X)` $P_{p,Y}(X)$

`\parallelTransportDir{p}{Y}(X)[c]` $P_{p,Y}^c(X)$

`\parallelTransportDir{p}{Y}[c]` $P_{p,Y}^c$

parallelTransportSymbol the symbol to use within `\parallelTransport` and `\parallelTransportDir`

`\parallelTransportSymbol` P

retract

a retraction.

Its syntax is `\retract[#1]{#2}{#3}`. The first argument can be used to scale the third. The second argument denotes the base point. The third argument denotes the tangent vector, which is optional, but if provided, the argument is put in brackets. The first following example illustrates the case, where no brackets are put. Note that the space is mandatory.

`\retract{p}X` $\text{retr}_p X$

`\retract{p}(X)` $\text{retr}_p(X)$

`\retract[Big]{p}(\frac{X}{2})` $\text{retr}_p\left(\frac{X}{2}\right)$


retractionSymbol symbol to use for a retraction and an inverse retraction, see `\retract` and


`\inverseRetract`.


`</>` `\retractionSymbol`  `retr`

riemannian

the Riemannian metric (family of inner products on the tangent spaces). Its syntax is `\riemannian[#1]{#2}{#3}[#4]`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard `amsmath` sizes.¹ The second argument denotes the first factor. The third argument denotes the second factor. The fourth (optional) argument denotes the base point of the tangent space.

`</>` `\riemannian{X_1}{X_2}`  (X_1, X_2)

`</>` `\riemannian{Y_1}{Y_2}[q]`  $(Y_1, Y_2)_q$


`</>` `\riemannian[Big]{\dfrac{1}{2}X_1}{X_2}[p]`  $\left(\frac{1}{2}X_1, X_2\right)_p$

riemanniannorm

the norm induced by the Riemannian metric. Its syntax is `\riemanniannorm[#1]{#2}[#3]`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard `amsmath` sizes.¹ The second argument denotes the argument. The third (optional) argument denotes the base point of the tangent space.


`</>` `\riemanniannorm{X}`  $\|X\|$

`</>` `\riemanniannorm{Y}[p]`  $\|Y\|_p$

`</>` `\riemanniannorm[Big]{\dfrac{1}{2}X}[p]`  $\left\|\frac{1}{2}X\right\|_p$

secondCovariantDerivative is the second-order covariant derivative.

Its syntax is `\secondCovariantDerivative{#1}{#2}[#3]`. The first argument is the vector (or vector field) determining the first direction of differentiation. The second argument is the vector (or vector field) determining the second direction of differentiation. The third (optional) argument denotes the tensor field being differentiated.

`</>` `\secondCovariantDerivative{X}{Y}{T}`  $D_{X,Y}^2 T$

secondCovariantDerivativeSymbol the symbol used for the second covariant derivative.

This is used within `\secondCovariantDerivative`.

`</>` `\secondCovariantDerivativeSymbol`  D^2

tangentSpace

the tangent space. Its syntax is `\tangentSpace{#1}[#2]`. The first argument

denotes the base point. The second (optional) argument denotes the manifold, which defaults to \mathcal{M} .

`\tangentSpace{p}` $\mathcal{T}_p\mathcal{M}$

`\tangentSpace{q}[\cN]` $\mathcal{T}_q\mathcal{N}$

tangentBundle the tangent bundle. Its syntax is `\tangentBundle[#1]`. The (optional) argument denotes the manifold, which defaults to \mathcal{M} .

`\tangentBundle` $\mathcal{T}\mathcal{M}$

`\tangentBundle[\cN]` $\mathcal{T}\mathcal{N}$

tangentSpaceSymbol the symbol used within `\tangent`.

`\tangentSpaceSymbol` \mathcal{T}

tensorBundle the tensor bundle. Its syntax is `\tensorBundle{#1}{#2}{#3}`. The first argument denotes the number r of elements of the cotangent space the tensors accept. The second argument denotes the number s of elements of the tangent space the tensors accept. The third (optional) argument denotes the manifold, which defaults to \mathcal{M} .

`\tensorBundle{r}{s}` $\mathcal{T}^{(r,s)}\mathcal{M}$

`\tensorBundle{r}{s}[\cN]` $\mathcal{T}^{(r,s)}\mathcal{N}$

tensorSpace a tensor space over a vector space V . Its syntax is `\tensorSpace{#1}{#2}{#3}`. The first argument denotes the number r of elements of the dual space V^* the tensors accept. The second argument denotes the number s of elements of the space V the tensors accept. The third (optional) argument denotes the vector space, which defaults to empty.

`\tensorSpace{r}{s}` $\mathcal{T}^{(r,s)}()$

`\tensorSpace{r}{s}[V]` $\mathcal{T}^{(r,s)}(V)$

tensorSpaceSymbol the symbol used within `\tensorSpace` and `\tensorBundle`.

`\tensorSpaceSymbol` \mathcal{T}

vectorTransport a vector transport.
Its syntax is `\vectorTransport[#1]{#2}{#3}({#4})[#5]`. The first (optional) argument is used to scale the parentheses enclosing the argument $\#4$.¹ The

second argument is the start point of vector transport on a manifold. The third argument is the end point of vector transport on a manifold. The fourth (optional) argument is the tangent vector that is transported. Putting it in brackets enables the scaling by the first argument. Finally a retraction symbol can be added in the exponent to distinguish vector transports as #5.

`</> \vectorTransport{p}{q}X` $T_{q \leftarrow p}X$

`</> \vectorTransport{p}{q}(X)` $T_{q \leftarrow p}(X)$

`</> \vectorTransport[big]{p}{q}(X)` $T_{q \leftarrow p}(X)$

`</> \vectorTransport{p}{q}(X)[\retractionSymbol]` $T_{q \leftarrow p}^{\text{retr}}(X)$

vectorTransportDir similar to `\vectorTransport`, but the third argument is a direction to transport into. This can be rewritten to the classical notation applying an retraction from the base point (#2) to th direction (#3).

`</> \vectorTransportDir{p}{Y}X` $T_{p,Y}X$

`</> \vectorTransportDir{p}{Y}(X)` $T_{p,Y}(X)$

`</> \vectorTransportDir[big]{p}{Y}(X)` $T_{p,Y}(X)$

`</> \vectorTransportDir{p}{Y}(X)[\retractionSymbol]` $T_{p,Y}^{\text{retr}}(X)$

vectorTransportSymbol the symbol to use within `\vectorTransport` and `\vectorTransportDir`

`</> \vectorTransportSymbol` T

8.2 OPTIMIZATION: mathsemantics-optimization.sty

The semantic file `mathsemantics-optimization.sty` collects definitions and notations related to optimization.

linearizingcone the linearizing cone. Its syntax is `\linearizingcone[#1]{#2}{#3}`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ The second argument denotes the set. The third argument denotes the base point.

`</> \linearizingcone{A}{x}` $\mathcal{T}_A^{\text{lin}}(x)$

`</> \linearizingcone{A}{x^2}` $\mathcal{T}_A^{\text{lin}}(x^2)$

`</> \linearizingcone[big]{A}{x^2}` $\mathcal{T}_A^{\text{lin}}(x^2)$

normalcone

the normal cone. Its syntax is `\normalcone[#1]{#2}{#3}`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ The second argument denotes the set. The third argument denotes the base point.

`</> \normalcone{A}{x}` $\mathcal{N}_A(x)$

`</> \normalcone{A}{x^2}` $\mathcal{N}_A(x^2)$

`</> \normalcone[big]{A}{x^2}` $\mathcal{N}_A(x^2)$

polarcone

the polar cone of a set `</> \polarcone{A}` A°

radialcone

the radial cone. Its syntax is `\radialcone[#1]{#2}{#3}`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ The second argument denotes the set. The third argument denotes the base point.

`</> \radialcone{A}{x}` $\mathcal{K}_A(x)$

`</> \radialcone{A}{x^2}` $\mathcal{K}_A(x^2)$

`</> \radialcone[big]{A}{x^2}` $\mathcal{K}_A(x^2)$

tangentcone

the tangent cone. Its syntax is `\tangentcone[#1]{#2}{#3}`. The first (optional) argument is used to scale the parentheses enclosing the argument to the standard amsmath sizes.¹ The second argument denotes the set. The third argument denotes the base point.

`</> \tangentcone{A}{x}` $\mathcal{T}_A(x)$

`</> \tangentcone{A}{x^2}` $\mathcal{T}_A(x^2)$

`</> \tangentcone[big]{A}{x^2}` $\mathcal{T}_A(x^2)$