The `ShorTeX` package

Trevor Campbell, Jonathan Huggins, and Jeff Negrea

Updated April 10, 2023

Abstract

The purpose of the ShorTeX (meta)package is to make the process of typesetting typical mathematical documents in LaTeX more efficient, and the resulting code easier to read. It achieves this by (1) providing an extensive, internally consistent, and easy to learn set of macro shorthands, (2) incorporating a set of packages that are dedicated to reducing manual coding effort, and (3) incorporating a collection of very common / standard boilerplate packages.

1 Usage and package options

Include ShorTeX by adding `\usepackage{shortex}` to the preamble of your document. ShorTeX will include and configure many common packages for you (e.g., `graphicx, subcaption, hyperref, algorithm, algpseudocode, amsmath`, among others), so you do not need to explicitly include and set these up yourself. If you are writing a document that must use a specific style file (e.g., for a conference or journal) that itself includes some of these packages, we recommend editing those style files to remove the package imports.

The ShorTeX package has a few options:

`manualnumbering` Do not include `autonum.sty`. This disables automatic equation numbering.

`blackhypersetup` Switch hyperlinks, citations, references, etc. to be typeset in black font. The default is dark blue.

You must compile your document 4 times when using ShorTeX to ensure that equation numbers and references update properly.

2 Packages included in ShorTeX

2.1 Internal packages

The following packages provide features that are used internally in ShorTeX for macro definitions etc.

xifthen, xstring, xspace, xargs asdf

2.2 Typical packages

We include a set of typical packages to avoid the problem of having to remember precisely what order to import each one, which options, compatibility, etc

It is worth noting that we did not include a standard bibliography package in ShorTeX (e.g., `natbib.sty`). This is because...

mathrsfs,dsfont,amsmath,amssymb,amsthm,bm,bbm,amsfonts,mathtools,thmtools asdf

hyperref asdf

color asdf

algorithm, algpseudocode asdf

graphicx asdf

2.3 Improvement packages

cleveref Typically to use a reference in LaTeX, you have to write the name of the type of reference yourself. For example, if you want to reference a figure, you would have to write something like:

```
In Figure \ref{fig:first}, you can see...
```

Or for multiple figures, you might use:

```
Figures \ref{fig:first}, \ref{fig:second},
and \ref{fig:third} show that...
```

The `cleveref` package simplifies this process significantly. Use the `\cref` command to automatically typeset the names of the objects you're referencing (including properly handling multiple references). The above two examples become

```
In \cref{fig:first}, you can see...
```

and

```
\cref{fig:first,fig:second,fig:third} show that...
```

This works for many different reference types (Figure, Algorithm, Equation, Table, etc), and can be extended if needed. See the `cleveref` documentation at https://ctan.org/pkg/cleveref?lang=en and the homepage at https://www.dr-qubit.org/cleveref.html for more information.

autonum Typically when you typeset equations, you have to choose between `$...$`, `$$...$$`, `\begin{align}...\end{align}`, `\begin{aligned}...\end{aligned}`, `\begin{equation}...\end{equation}`, not to mention starred versions of those environments and `\nonumber`/`\notag` commands, depending on whether/where you want equation numbers, display or in-text math, etc. This leads to verbose, inconsistent code.

The `autonum` package automatically decides which equations to provide numbers based on *which equations you reference*. So when using ShorTeX, you only need two commands for math mode: single dollar signs `$...$` for inline math, and `align` environments (redefined in ShorTeX to be `\[...\]`) for display math.[1]

For example, if you create the following display math,

```
\[
   a+b = c \label{eq:the_equation}
\]
```

then if you use the command `\cref{eq:the_equation}` somewhere in the document, that equation will automatically be assigned a number. If not, it won't get a number. See the `autonum` package documentation https://ctan.org/pkg/autonum?lang=en for more information.

nicefrac asdf

crossreftools asdf

multirow asdf

wrapfig asdf

caption,subcaption asdf

microtype asdf

booktabs asdf

import,subfiles asdf

url asdf

[1] Note that there are minor differences between how `align` and `equation` display equations. But after 10+ years of using LaTeX, I have not ever encountered a case where it mattered much. That being said, ShorTeX does not *disable* any functionality, so you can use the usual environments anywhere you feel it is necessary.

3 Shortened commands and macros

3.1 Environments

LaTeX documents often includes a lot of verbose code related to creating environments (`\begin{blah}...\end{blah}`). ShorTeX provides a set of shortened macros for common environments. Note that all theorem-like environments (theorem, lemma, proposition, etc.) are numbered by default; unnumbered versions can be obtained by appending a `u`. For example, `\bthmu...\ethmu` creates an unnumbered theorem environment, while `\blemu...\elemu` creates an unnumbered lemma environment.

Environment	Syntax
abstract	`\babs...\eabs`
algorithm	`\balg...\ealg`
algorithmic	`\balgc...\ealgc`
table	`\btab...\etab`
subtable	`\bsubtab...\esubtab`
tabular	`\btabr...\etabr`
figure	`\bfig...\efig`
figure*	`\bfigs...\efigs`
subfigure	`\bsubfig...\esubfig`
center	`\bcent...\ecent`
align	`\[...\]`
inline math	`$...$`

Note: These are numbered theorem-like environments. For unnumbered, append a `u`*: e.g.,* `bthmu...ethmu`.

theorem	`\bthm...\ethm`
lemma	`\blem...\elem`
proposition	`\bprop...\eprop`
corollary	`\bcor...\ecor`
conjecture	`\bconj...\econj`
definition	`\bdef...\edef`
assumption	`\bassump...\eassump`
example	`\bexa...\eexa`
remark	`\brmk...\ermk`
fact	`\bfact...\efact`
exercise	`\bexer...\eexer`
proof	`\bprf...\eprf`
proofof	`\bprfof{\cref{proof_label}}...\eprfof`

3.2 Delimiters

Mathematics in LaTeX often includes quite a few delimiters (parentheses, brackets, curly brackets, etc.). A very common usage of these involves the `\left...` and `\right...` commands for automatic sizing. ShorTeX defines the shortened `\lt...` and `\rt...` to replace these.

Description	Example	Text style	Display style
Parentheses	`\lt(\frac{x}{y}\rt)`	$\left(\frac{x}{y}\right)$	$\left(\dfrac{x}{y}\right)$
Curly brackets	`\lt\{\frac{x}{y}\rt\}`	$\left\{\frac{x}{y}\right\}$	$\left\{\dfrac{x}{y}\right\}$
Square brackets	`\lt[frac{x}{y}\rt]`	$\left[\frac{x}{y}\right]$	$\left[\dfrac{x}{y}\right]$
Pipes	`\lt|frac{x}{y}\rt|`	$\left\lvert\frac{x}{y}\right\rvert$	$\left\lvert\dfrac{x}{y}\right\rvert$
Double Pipes	`\lt\\|frac{x}{y}\rt\\|`	$\left\lVert\frac{x}{y}\right\rVert$	$\left\lVert\dfrac{x}{y}\right\rVert$
Angles	`\lt<frac{x}{y}\rt>`	$\left\langle\frac{x}{y}\right\rangle$	$\left\langle\dfrac{x}{y}\right\rangle$

You can specify a bracket size using $*$ for `\left` and `\right` or one of the standard size choices (`\big`, `\Big`, `\bigg`, `\Bigg`).

Description	Example	Text style	Display style
Round brackets (i.e., parentheses)	`\rbra{\frac{x}{y}}`	$(\frac{x}{y})$	$(\dfrac{x}{y})$
Curly brackets	`\cbra*{\frac{x}{y}}`	$\left\{\frac{x}{y}\right\}$	$\left\{\dfrac{x}{y}\right\}$
Square brackets	`\sbra[\bigg]{\frac{x}{y}}`	$\left[\frac{x}{y}\right]$	$\left[\dfrac{x}{y}\right]$

Many other bracket-like, semantic commands are also available:

Description	Example	Text style	Display style
Absolute value	`\abs{\frac{x}{y}}`	$\lvert\frac{x}{y}\rvert$	$\left\lvert\frac{x}{y}\right\rvert$
Set	`\set{\frac{x}{y}, \frac{y}{z}}`	$\{\frac{x}{y}, \frac{y}{z}\}$	$\left\{\frac{x}{y}, \frac{y}{z}\right\}$
Floor	`\floor{\frac{x}{y}}`	$\lfloor\frac{x}{y}\rfloor$	$\left\lfloor\frac{x}{y}\right\rfloor$
Ceiling	`\ceil{\frac{x}{y}}`	$\lceil\frac{x}{y}\rceil$	$\left\lceil\frac{x}{y}\right\rceil$
Norm	`\norm{\frac{x}{y}}`	$\lVert\frac{x}{y}\rVert$	$\left\lVert\frac{x}{y}\right\rVert$
Inner product	`\inner{\frac{x}{y}}{\frac{y}{z}}`	$\langle\frac{x}{y}, \frac{y}{z}\rangle$	$\left\langle\frac{x}{y}, \frac{y}{z}\right\rangle$
Cardinality	`\card{\whA}`	$\lvert\widehat{A}\rvert$	$\lvert\widehat{A}\rvert$

The norm and inner product commands also have versions with a subscript argument:

Description	Example	Text style	Display style
Norm with subscript	`\normsub*{\frac{x}{y}}{2}`	$\left\lVert\frac{x}{y}\right\rVert_2$	$\left\lVert\frac{x}{y}\right\rVert_2$
Inner product with subscript	`\innersub*{\frac{x}{y}}{z}{2}`	$\left\langle\frac{x}{y}, z\right\rangle_2$	$\left\langle\frac{x}{y}, z\right\rangle_2$

3.3 L_p Spaces and Operators

Description	Example	Text style	Display style
L_p space	`\Lp{2}`	L_2	L_2
L_p space for specified measure	`\Lpmeas{2}{\hmu}`	$L_2(\hat{\mu})$	$L_2(\hat{\mu})$
	`\Lpmeas[\Big]{2}{\hmu}`	$L_2\Big(\hat{\mu}\Big)$	$L_2\Big(\hat{\mu}\Big)$
L_p norm	`\Lpnorm{\hGamma}{2}`	$\|\hat{\Gamma}\|_{L_2}$	$\|\hat{\Gamma}\|_{L_2}$
	`\Lpnorm*{\hGamma}{2}`	$\left\|\hat{\Gamma}\right\|_{L_2}$	$\left\|\hat{\Gamma}\right\|_{L_2}$
	`\Lpnorm*{\Gamma}{2}`	$\|\Gamma\|_{L_2}$	$\|\Gamma\|_{L_2}$
	`\left\Vert{\hGamma}\right\Vert_{2}`	$\left\|\hat{\Gamma}\right\|_2$	$\left\|\hat{\Gamma}\right\|_2$
	`\left\Vert{\Gamma}\right\Vert_{2}`	$\|\Gamma\|_2$	$\|\Gamma\|_2$
L_p norm for specified measure	`\Lpmeasnorm{\hGamma}{2}{\hmu}`	$\|\hat{\Gamma}\|_{L_2(\hat{\mu})}$	$\|\hat{\Gamma}\|_{L_2(\hat{\mu})}$
	`\Lpmeasnorm[\Big]{\hGamma}{2}{\hmu}`	$\left\|\hat{\Gamma}\right\|_{L_2(\hat{\mu})}$	$\left\|\hat{\Gamma}\right\|_{L_2(\hat{\mu})}$
L_p inner product	`\Lpinner{\hGamma}{\Gamma}{2}`	$\langle\hat{\Gamma},\Gamma\rangle_{L_2}$	$\langle\hat{\Gamma},\Gamma\rangle_{L_2}$
	`\Lpinner*{\hGamma}{\Gamma}{2}`	$\left\langle\hat{\Gamma},\Gamma\right\rangle_{L_2}$	$\left\langle\hat{\Gamma},\Gamma\right\rangle_{L_2}$
L_p inner product for specified measure	`\Lpmeasinner{\hGamma}{\Gamma}{2}{\hmu}`	$\langle\hat{\Gamma},\Gamma\rangle_{L_2(\hat{\mu})}$	$\langle\hat{\Gamma},\Gamma\rangle_{L_2(\hat{\mu})}$
	`\Lpmeasinner[\big]{\hGamma}{\Gamma}{2}{\hmu}`	$\left\langle\hat{\Gamma},\Gamma\right\rangle_{L_2(\hat{\mu})}$	$\left\langle\hat{\Gamma},\Gamma\right\rangle_{L_2(\hat{\mu})}$

3.4 annotation commands

`\barA`	$\bar{A}$
`\bara`	$\bar{a}$
`\bA`	$\bar{A}$
`\bB`	$\bar{B}$
`\balpha`	$\bar{\alpha}$
`\bGamma`	$\bar{\Gamma}$
`\mcA`	$\mathcal{A}$
`\hmcA`	$\hat{\mathcal{A}}$
`\mfA`	$\mathfrak{A}$
`\mfa`	$\mathfrak{a}$
`\bmfA`	$\mathfrak{A}$
`\bmfa`	$\mathfrak{a}$
`\hA`	$\hat{A}$
`\ha`	$\hat{a}$
`\halpha`	$\hat{\alpha}$
`\hGamma`	$\hat{\Gamma}$
`\bhA`	$\hat{\mathbf{A}}$
`\bha`	$\hat{\mathbf{a}}$
`\bhalpha`	$\hat{\alpha}$
`\bhGamma`	$\hat{\mathbf{\Gamma}}$
`\whA`	$\widehat{A}$
`\wha`	$\widehat{a}$
`\tdA`	$\tilde{A}$
`\tda`	$\tilde{a}$
`\tdalpha`	$\tilde{\alpha}$
`\tdGamma`	$\tilde{\Gamma}$
`\btdA`	$\tilde{\mathbf{A}}$
`\btda`	$\tilde{\mathbf{a}}$
`\btdalpha`	$\tilde{\alpha}$
`\btdGamma`	$\tilde{\mathbf{\Gamma}}$
`\biA`	$\boldsymbol{A}$
`\bia`	$\boldsymbol{a}$
`\bhiA`	$\hat{\boldsymbol{A}}$
`\bhia`	$\hat{\boldsymbol{a}}$

3.5 new approach to annotation commands

Out of the box we can get all sorts of cool fonts in math mode using `\f[<fontcodestring>]A`. For the time being I only implemented 2 fonts and 1 accent. These can be expanded once everything else is set. We see the effect of single font codes:

`\f[b]A`	$\mathbf{A}$
`\f[k]A`	$\mathbb{A}$
`\f[h]A`	$\widehat{A}$

And multiple font codes:

`\f[bh]A` $\widehat{\mathbf{A}}$
`\f[hb]A` $\widehat{\mathbf{A}}$
`\f[hk]A` $\widehat{\mathbb{A}}$
`\f[kh]A` $\mathbb{A}$

Note that these are expanded in the reverse of the order they appear: the font code furthest to the right is applied first. This natches the order that the corresponding commands would appear in TeX code.

We can also avoid typing `[]` for some combinations of font codes we sue frequently. To do this, use `\parsefontstylesstrings{{<fcstr1>},{<fcstr2>},...}{<alphabet>}` as demonstrated below. For "ease of use" we have implemented `\upperCaseRomanLetters` and `\lowerCaseRomanLetters` . .

```
\parsefontstylesstrings{{hb},{hk}}{ABCDEFG}                    ...
\parsefontstylesstrings{{hb},{hk}}{\lowerCaseRomanLetters}     ...
\fhbA                                                          Â
\fhkB                                                          B̂
\fhbx                                                          x̂
\fhby                                                          ŷ
\fhkz                                                          F̂
```

Since `\mathbb<lowercaseletter>` is defined to give weird characters, our macros do the same.

4 Example Document

TODO: a full example in basic latex versus shortex