Package **innerscript** v. 1.4 Implementation

Conrad Kosowsky

April 2025

`kosowsky.latex@gmail.com`

For easy, off-the-shelf use, type the following in your
document preamble and compile using LuaLATEX:

`\usepackage{innerscript}`

Overview

The **innerscript** package optionally modifies four aspects of TEX's automatic math formatting to improve typesetting: (1) it adds extra space around relation and operation symbols in superscripts and subscripts; (2) it removes extra space around `\left`-`\right` delimiter pairs; (3) it adds extra space after right delimiters in certain situations; and (4) it forces `\left` and `\right` delimiters to completely cover their contents. Using LuaLATEX is required.

This file documents the code for the **innerscript** package. It is not a user guide! If you are looking for instructions on how to load **innerscript** and what it can do for your document, see `innerscript-user-guide.pdf`, which is included with the **innerscript** installation and is available on CTAN. Section 1 begins with the package declaration and option processing. Sections 2 and 3 contain the code to adjust spacing between symbols in `\scriptstyle` and `\scriptscriptstyle` respectively, and Section 4 contains the code for adjusting the spacing around `\mathinner` subformulas. Section 5 handles spacing between `\mathclose` and `\mathord` atoms, and section 6 fills out the resizable delimiter heights. Version history appears at the end of the document.

1 Setup

We begin the implementation by declaring the package and defining a general informational macro. The first 60 lines of `innerscript.sty` are comments.

```
61 \NeedsTeXFormat{LaTeX2e}
62 \ProvidesPackage{innerscript}[2025/04/07 v. 1.4 Package innerscript]
63 \def\IS@info#1{\wlog{Package innerscript Info: #1}}
```

The conditionals will encode package option information.

```
64 \newif\ifIS@script
65 \newif\ifIS@scriptscript
66 \newif\ifIS@legacyscript
```

Acknowledgements: Thanks to Sheldon Axler and Clea F. Rees for pointing a bug in a previous version of **innerscript**.

```
67 \newif\ifIS@legacyscriptscript
68 \newif\ifIS@inner
69 \newif\ifIS@close
70 \newif\ifIS@cover
```

And set default options.

```
71 \IS@scripttrue
72 \IS@scriptscripttrue
73 \IS@legacyscriptfalse
74 \IS@legacyscriptscriptfalse
75 \IS@innertrue
76 \IS@closetrue
77 \IS@covertrue
```

Now define and process the package options.

```
78 \DeclareOption{script}              {\IS@scripttrue}
79 \DeclareOption{scriptscript}        {\IS@scriptscripttrue}
80 \DeclareOption{inner}              {\IS@innertrue}
81 \DeclareOption{close}              {\IS@closetrue}
82 \DeclareOption{cover}              {\IS@covertrue}
83 \DeclareOption{legacy-script}      {\IS@scripttrue
84   \IS@legacyscripttrue}
85 \DeclareOption{legacy-scriptscript}{\IS@scriptscripttrue
86   \IS@legacyscriptscripttrue}
87 \DeclareOption{no-script}          {\IS@scriptfalse}
88 \DeclareOption{no-scriptscript}    {\IS@scriptscriptfalse}
89 \DeclareOption{no-inner}          {\IS@innerfalse}
90 \DeclareOption{no-close}          {\IS@closefalse}
91 \DeclareOption{no-cover}          {\IS@coverfalse}
92 \ProcessOptions*
```

Check whether `\Umathordordspacing` is defined. If not, issue an error and `\endinput`. We normally can't print multiple spaces at once, so within a group, we change the space catcode to 12.

```
93 \ifx\Umathordordspacing\@undefined
94 \bgroup
95 \catcode`\ =12\relax
96 \def\IS@LuaTeXError{\GenericError{}%
97 {\MessageBreak\MessageBreak
98 Package innerscript error:%
99 \MessageBreak\MessageBreak
100 ******************\MessageBreak
101 *                *\MessageBreak
102 *   CANNOT LOAD   *\MessageBreak
103 *   INNERSCRIPT    *\MessageBreak
104 *  LuaTeX Needed  *\MessageBreak
105 *                *\MessageBreak
106 ******************\MessageBreak\@gobbletwo}%
107 {See the innerscript package documentation for explanation.}%
```

```
108 {I need LuaTeX to make the innerscript package work. It\MessageBreak
109 looks like the current engine is something else, so I\MessageBreak
110 can't load the package file. To use innerscript, please\MessageBreak
111 typeset with LuaLaTeX. To continue without innerscript,\MessageBreak
112 press return.}}%
113 \expandafter\egroup
114 \IS@LuaTeXError
115   \AtEndOfPackage{\IS@info{Failed to load \on@line!}}
116   \expandafter\endinput % we \endinput with a balanced conditional
117 \fi
```

If the user requested a legacy option for super or subscripts, issue a warning. We store the warning in `\@tempa`, and if both `\ifIS@legacyscript` and `\ifIS@legacyscriptscript` and false, we turn `\@tempa` into `\relax`.

```
118 \def\@tempa{%
119   \PackageWarningNoLine{innerscript}
120   {I'm ignoring your request for\MessageBreak
121   legacy spacing\on@line\MessageBreak
122   and using the default space\MessageBreak
123   adjustments instead since this\MessageBreak
124   package option is deprecated}}
125 \ifIS@legacyscript
126 \else
127   \ifIS@legacyscriptscript
128   \else
129     \let\@tempa\relax
130   \fi
131 \fi
132 \@tempa
```

2   Superscripts and Subscripts

First we set the spacing for `\scriptstyle` and `\scriptscriptstyle` atoms. Each space adjustment has the form

$$\texttt{\textbackslash Umath}\langle \mathit{classes}\rangle\texttt{spacing}\langle \mathit{style}\rangle\texttt{\textbackslash muexpr}\langle \mathit{skip}\rangle*\langle \mathit{factor}\rangle\texttt{\textbackslash relax}.$$

The ⟨*classes*⟩ are a pair of choices from `ord`, `op`, `bin`, `rel`, `open`, `close`, `punct`, and `inner`, and the ⟨*style*⟩ is either `\scriptstyle`, `\scriptscriptstyle`, or the cramped version of each style. The ⟨*skip*⟩ is the amount of muglue that typically appears between the ⟨*classes*⟩ while in `\textstyle` or `\displaystyle`. The package takes that muglue and scales it by 0.6 for superscripts and subscripts and by 0.4 for second-order superscripts and subscripts. Table 1 shows the usual amounts of space between consecutive atom types. To actually code it, we define `\@tempa` to expand to various space-adjustment commands depending on package options.

```
133 \ifIS@script        % change \scriptstyle
134   \IS@info{Adjusting space for \string\scriptstyle.}
```

Table 1: Space Inserted between Atoms

Consecutive Atom Types	Default space added
\mathord\mathop	\thinmuskip
\mathord\mathbin	\medmuskip
\mathord\mathrel	\thickmuskip
\mathord\mathinner	\thinmuskip
\mathop\mathord	\thinmuskip
\mathop\mathop	\thinmuskip
\mathop\mathrel	\thickmuskip
\mathop\mathinner	\thickmuskip
\mathbin\mathord	\medmuskip
\mathbin\mathop	\medmuskip
\mathbin\mathopen	\medmuskip
\mathbin\mathinner	\medmuskip
\mathrel\mathord	\thickmuskip
\mathrel\mathop	\thickmuskip
\mathrel\mathopen	\thickmuskip
\mathrel\mathinner	\thickmuskip
\mathclose\mathop	\thinmuskip
\mathclose\mathbin	\medmuskip
\mathclose\mathrel	\thickmuskip
\mathclose\mathinner	\thinmuskip
\mathpunct\mathord	\thinmuskip
\mathpunct\mathop	\thinmuskip
\mathpunct\mathrel	\thinmuskip
\mathpunct\mathopen	\thinmuskip
\mathpunct\mathclose	\thinmuskip
\mathpunct\mathpunct	\thinmuskip
\mathpunct\mathinner	\thinmuskip
\mathinner\mathord	\thinmuskip
\mathinner\mathop	\thinmuskip
\mathinner\mathbin	\medmuskip
\mathinner\mathrel	\thickmuskip
\mathinner\mathopen	\thinmuskip
\mathinner\mathpunct	\thinmuskip
\mathinner\mathinner	\thinmuskip

```
135   \IS@info{Adjusting space for \string\crampedscriptstyle.}
136   \ifIS@scriptscript % change \scriptscriptstyle
137     \IS@info{Adjusting space for \string\scriptscriptstyle.}
138     \IS@info{Adjusting space for \string\crampedscriptscriptstyle.}
```

In this branch, `\@tempa` expands to space-adjustment commands for all four styles of superscripts and subscripts. The `#1` argument is a `\Umath` spacing primitive, and `#2` is a muglue register.

```
139     \def\@tempa#1#2{%
140       #1\scriptstyle              \muexpr #2 * 6 / 10\relax
141       #1\crampedscriptstyle       \muexpr #2 * 6 / 10\relax
142       #1\scriptscriptstyle        \muexpr #2 * 4 / 10\relax
143       #1\crampedscriptscriptstyle\muexpr #2 * 4 / 10\relax}
144   \else              % do not change \scriptscriptstyle
145     \IS@info{No space changes for \string\scriptscriptstyle.}
146     \IS@info{No space changes for \string\crampedscriptscriptstyle.}
```

Now change just first-order superscripts and subscripts.

```
147     \def\@tempa#1#2{%
148       #1\scriptstyle              \muexpr #2 * 6 / 10\relax
149       #1\crampedscriptstyle       \muexpr #2 * 6 / 10\relax}
150   \fi
151 \else                     % do not change \scriptstyle
152   \IS@info{No space changes for \string\scriptstyle.}
153   \IS@info{No space changes for \string\crampedscriptstyle.}
154   \ifIS@scriptscript % change \scriptscriptstyle
155     \IS@info{Adjusting space for \string\scriptscriptstyle.}
156     \IS@info{Adjusting space for \string\crampedscriptscriptstyle.}
```

In this case, change just second-order superscripts and subscripts.

```
157     \def\@tempa#1#2{%
158       #1\scriptscriptstyle        \muexpr #2 * 4 / 10\relax
159       #1\crampedscriptscriptstyle\muexpr #2 * 4 / 10\relax}
160   \else              % do not change \scriptscriptstyle
161     \IS@info{No space changes for \string\scriptscriptstyle.}
162     \IS@info{No space changes for \string\crampedscriptscriptstyle.}
```

And here change nothing.

```
163     \let\@tempa\@gobbletwo
164   \fi
165 \fi
```

Now actually make the space changes as applicable.

```
166   \@tempa{\Umathordopspacing}     {\thinmuskip}
167   \@tempa{\Umathordbinspacing}    {\medmuskip}
168   \@tempa{\Umathordrelspacing}    {\thickmuskip}
169   \@tempa{\Umathordinnerspacing}  {\thinmuskip}
170   \@tempa{\Umathopordspacing}     {\thinmuskip}
171   \@tempa{\Umathopopspacing}      {\thinmuskip}
172   \@tempa{\Umathoprelspacing}     {\thickmuskip}
173   \@tempa{\Umathopinnerspacing}   {\thickmuskip}
174   \@tempa{\Umathbinordspacing}    {\medmuskip}
175   \@tempa{\Umathbinopspacing}     {\medmuskip}
176   \@tempa{\Umathbinopenspacing}   {\medmuskip}
177   \@tempa{\Umathbininnerspacing}  {\medmuskip}
```

```
178    \@tempa{\Umathrelordspacing}      {\thickmuskip}
179    \@tempa{\Umathrelopspacing}       {\thickmuskip}
180    \@tempa{\Umathrelopenspacing}     {\thickmuskip}
181    \@tempa{\Umathrelinnerspacing}    {\thickmuskip}
182    \@tempa{\Umathcloseopspacing}     {\thinmuskip}
183    \@tempa{\Umathclosebinspacing}    {\medmuskip}
184    \@tempa{\Umathcloserelspacing}    {\thickmuskip}
185    \@tempa{\Umathcloseinnerspacing}{\thinmuskip}
186    \@tempa{\Umathpunctordspacing}    {\thinmuskip}
187    \@tempa{\Umathpunctopspacing}     {\thinmuskip}
188    \@tempa{\Umathpunctrelspacing}    {\thinmuskip}
189    \@tempa{\Umathpunctopenspacing}   {\thinmuskip}
190    \@tempa{\Umathpunctclosespacing}{\thinmuskip}
191    \@tempa{\Umathpunctpunctspacing}{\thinmuskip}
192    \@tempa{\Umathpunctinnerspacing}{\thinmuskip}
193    \@tempa{\Umathinnerordspacing}    {\thinmuskip}
194    \@tempa{\Umathinneropspacing}     {\thinmuskip}
195    \@tempa{\Umathinnerbinspacing}    {\medmuskip}
196    \@tempa{\Umathinnerrelspacing}    {\thickmuskip}
197    \@tempa{\Umathinneropenspacing}   {\thinmuskip}
198    \@tempa{\Umathinnerpunctspacing}{\thinmuskip}
199    \@tempa{\Umathinnerinnerspacing}{\thinmuskip}
```

3   Mathclose Spacing

In `\textstyle` and `\displaystyle`, we add half a `\thinmuskip` between `\mathclose` and `\mathord` atoms. As with the `script` and `scriptscript` options, we scale the spacing by 0.6 and 0.4 for `\scriptstyle` and `\scriptscriptstyle` respectively. Even though `close` comes after `inner` in the options list in the user guide, it is important to put it before `inner` in the implementation. We should set all interatom spacing before changing the spacing around `\mathinner` subformulas because the `inner` spacing changes reference values of other `\Umath` spacing primitives.

```
200 \ifIS@close
201   \IS@info{Adding space after closing grouping symbols.}
202   \Umathcloseordspacing\displaystyle        \muexpr\thinmuskip / 2\relax
203   \Umathcloseordspacing\textstyle            \muexpr\thinmuskip / 2\relax
204   \Umathcloseordspacing\crampeddisplaystyle \muexpr\thinmuskip / 2\relax
205   \Umathcloseordspacing\crampedtextstyle     \muexpr\thinmuskip / 2\relax
206   \ifIS@script
207     \Umathcloseordspacing\scriptstyle        \muexpr\thinmuskip * 3 / 10\relax
208     \Umathcloseordspacing\crampedscriptstyle\muexpr\thinmuskip * 3 / 10\relax
209   \fi
210   \ifIS@scriptscript
211     \Umathcloseordspacing\scriptscriptstyle \muexpr\thinmuskip / 5\relax
212     \Umathcloseordspacing\crampedscriptscriptstyle
213       \muexpr\thinmuskip / 5\relax
```

```
214   \fi
215 \else
216   \IS@info{No changes to space after delimiters.}
217 \fi
```

4   Mathinner Subformulas

Now set the spacing for `\mathinner` subformulas. Prior to version 1.4, `innerscript` spaced `\mathinner` subformulas under the `inner` option like `\mathord` atoms, but as of version 1.4, `innerscript` treats these subformulas like `\mathopen` on the left and `\mathclose` on the right. Every space adjustment in this section has the form

$$\text{\texttt{\textbackslash Umath}}\langle \textit{classes}\rangle\texttt{spacing}\langle \textit{style}\rangle\texttt{=\textbackslash the\textbackslash Umath}\langle \textit{other classes}\rangle\texttt{spacing}\langle \textit{style}\rangle\texttt{\textbackslash relax}.$$

In most cases, the ⟨*classes*⟩ are a pair of `inner` and one of `ord`, `op`, `bin`, `rel`, `open`, `close`, or `punct`, and the ⟨*other classes*⟩ are the same except with `inner` replaced with `open` or `close` depending on whether it appears on the left or the right. We can also have ⟨*classes*⟩ be `innerinner`, in which case ⟨*other classes*⟩ is `closeopen`. The ⟨*style*⟩ is one of the eight math style commands: `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`, and the cramped versions. This section comes after `script` and `close` in the implementation because the spacing here depends on the spacing of other math classes.

```
218 \ifIS@inner
219   \IS@info{Adjusting space around \string\mathinner\space subformulas.}
```

First, we loop through the math classes and let `\@i` be a math class.

```
220   \@tfor\@i:=\displaystyle\crampeddisplaystyle
221     \textstyle\crampedtextstyle
222     \scriptstyle\crampedscriptstyle
223     \scriptscriptstyle\crampedscriptscriptstyle
224     \do{%
```

For each style, first set `\Umathinnerinnerspacing`, and then loop through the math classes. We simultaneously set the spacing for atom pairs where `inner` appears either first or second. Apparently the `\Umath` spacing primitives do not expand the following control sequences in looking for a math style control sequence, so we need many `\expandafter`s here.

```
225       \expandafter\Umathinnerinnerspacing\@i=\expandafter
226         \the\expandafter\Umathcloseopenspacing\@i\relax
227       \@for\@j:=ord,op,bin,rel,open,close,punct\do{%
228         \csname Umath\@j innerspacing\expandafter\endcsname\@i=\the
229           \csname Umath\@j openspacing\expandafter\endcsname\@i\relax
230         \csname Umathinner\@j spacing\expandafter\endcsname\@i=\the
231           \csname Umathclose\@j spacing\expandafter\endcsname\@i\relax}}
232 \else
233   \IS@info{No changes to space for \string\mathinner\space subformulas.}
234 \fi
```

5   Delimiter Heights

Finally, filling out the delimiter heights is easy. We set `\delimiterfactor` to 1000.

```
235 \ifIS@cover
236   \IS@info{Setting delimiters to full height.}
237   \delimiterfactor\@m
238 \else
239   \IS@info{No changes to delimiter heights.}
240 \fi
241 \wlog{}
```

And that's a wrap!

Version History

1.0 . . . . February 2021
  —initial release

1.1 . . . . February 2021
  —bug fix for `\mathinner` spacing
  —added `\IS@skip` muglue register

1.2 . . . . November 2023
  —redesigned spacing for options `script`
  and `scriptscript`
  —no more `\IS@skip`—options `script`
  and `scriptscript` now use `\the`
  —added option `close`
  —added option `cover`
  —separated implementation and user
  guide
  —added `\scalemu`

1.3 . . . . August 2024
  —bug fix involving `\cramedisplaystyle`
  typo

1.4 . . . . April 2025
  —refactored package code
  —`legacy-` package options for
  superscripts and subscripts now same as
  default options
  —use `\muexpr` instead of `\scalemu`
  —`inner` option changed to look more like
  `\mathopen` and `\mathclose` atoms

Index