

# Intelligent brackets

## The ibrackets package\*

Antoine Missier  
antoine.missier@ac-toulouse.fr

December 19, 2022

## 1 Introduction

Open intervals are usually represented with parenthesis  $(0, +\infty)$  but sometimes we find also brackets  $]0, +\infty[$ , for example in French mathematics. In that case the space around them is often unsuitable, e.g.  $x \in ]0, +\infty[$ . This small package redefines brackets symbols `[` and `]` for mathematical mode to get correct spacing:  $x \in ]0, +\infty[$ .

Originally implemented in the mismath package [1] and also in frenchmath [2] since version 2.1, our previous redefinitions produce however incorrect spacing when the left boundary of the interval begins with a sign - or +, which was then interpreted as a binary operation. Thus blank spaces surrounding the sign would have been too large. This problem was pointed out by Jean-François Burnol, and an easy solution, that has been documented, consisted to nest the operator or the left boundary within a pair of braces, e.g.  $\$x \in ]\{-\}\infty, 0]\$$ , or use `\left` and `\right` or even `\mathopen{[]}`.

Inspired by the icomma package [3] of Walter Schmidt, we now provide an improved bracket definition that works correctly without these pair of brackets.

Let us also mention other approaches e.g. `\DeclarePairedDelimiters`, a macro from the mathtools package [4], or the interval package [5] with his `\interval` macro. Nevertheless our solution is lighter.

## 2 Usage

You just have to type intervals in an easy way:  $\$x \in ]0, \pi[ \cup ]2\pi, 3\pi[\$$  produce

$$x \in ]0, \pi[ \cup ]2\pi, 3\pi[ \quad \text{with ibrackets,}$$

instead of

$$x \in ]0, \pi[\cup]2\pi, 3\pi[ \quad \text{without ibrackets.}$$

---

\*This document corresponds to ibrackets v1.0, dated 2022/12/19.

Generally [ and ] symbols are not defined anymore as delimiters, but as ordinary characters. Thereby a line break could occur between the two brackets, but it is always possible to transform them into delimiters with `\left` and `\right`.

The problem of a sign following the first bracket is solved with this package, so the example in the introduction is simply obtained with `$x \in ]-\infty, 0]$` which gives  $x \in ]-\infty, 0]$ .

*However, you don't have to leave a space between the first bracket and the sign:* e.g. `$x \in ] -\infty, 0]$` yields  $x \in ] - \infty, 0]$  with bad spacing around the minus sign. Contrariwise, when you want to write algebra on intervals then you must leave a blank space between the second bracket and the +/- operations, e.g. `$[a, b] + [c, d]$` yields  $[a, b] + [c, d]$  but `$[a, b]+ [c, d]$` yields  $[a, b]+[c, d]$ . To summarize the new behavior of a bracket: it is an ordinary character, but an open delimiter when it is immediately followed by a + or - character.

### 3 Implementation

At `\begin{document}`, we memorize the `\mathcode` of the original brackets, in the `\math...bracket` macros, and we make the brackets in math mode active:

```
1 \AtBeginDocument{%
2   \mathchardef\mathopenbracket\mathcode'[%
3   \mathcode'["8000
4   \mathchardef\mathclosebracket\mathcode']%
5   \mathcode']="8000
```

The active brackets check the next input character. If this is a - or a +, the active brackets return `\mathopen` with the saved `\math...bracket` so that no space will be added after the bracket; otherwise, `\mathord\math...bracket` is returned :

```
6 }
7 {\catcode'[\active
8   \gdef[\futurelet\@next\sm@rtopenbracket]
9 }
10 \def\sm@rtopenbracket{%
11   \ifx\@next- \mathopen \else
12   \ifx\@next+ \mathopen \else
13     \mathord\fi\fi \mathopenbracket}
14
15 {\catcode']=\active
16   \gdef[\futurelet\@next\sm@rtcclosebracket]
17 }
18 \def\sm@rtcclosebracket{%
19   \ifx\@next- \mathopen \else
20   \ifx\@next+ \mathopen \else
21     \mathord\fi\fi \mathclosebracket}
```

We could have use the internal  $\TeX$  command `\@ifnextchar` to skip blank spaces after the bracket, and look if there is a + or - after, but then it would become tricky when you really want to follow an interval with an operation plus or minus.

## References

- [1] *mismath* – *Miscellaneous mathematical macros*. Antoine Missier, CTAN, v2.0 2022/11/11.
- [2] *L'extension frenchmath*. Antoine Missier, CTAN, v2.2 2022/12/15.
- [3] *The icomma package for  $\text{\LaTeX}$  2 $\epsilon$* . Walter Schmidt, CTAN, v2.0 2002/03/10.
- [4] *The mathtool package*. Morten Høgholm, Lars Madsen, CTAN, v1.21 2018/01/08.
- [5] *The interval package*. Lars Madsen, CTAN, v0.4 2019/03/06.