

# The intervalx package<sup>\*</sup>

Valentin Dao<sup>†</sup>

Released 2025-11-16

## Abstract

The purpose of this package is to extend `interval`'s<sup>1</sup> functionlities by improving the main macro and adding a few new ones. Although the implementation is has been modernised through the use of `expl3`, the approach is very similar to that used by Lars Madsen. As such, most keys will have the same name. Even though `intervalx` is presented as an extension of `interval`, new macros for composing inequalities have nevertheless been implemented.

## Contents

<b>1</b>	<b>Package options</b>	<b>2</b>
<b>2</b>	<b>Typeset an interval</b>	<b>2</b>
<b>3</b>	<b>Interval product, union, and intersection</b>	<b>4</b>
3.1	Product . . . . .	4
3.2	Union . . . . .	5
3.3	Intersection . . . . .	5
<b>4</b>	<b>Inequalities</b>	<b>6</b>
	<b>Changelog</b>	<b>6</b>

---

<sup>\*</sup>This file describes v1.0.0

<sup>†</sup>E-mail: [vdao.texdev@gmail.com](mailto:vdao.texdev@gmail.com)

<sup>1</sup>See on CTAN.

## License

© 2025 Valentin Dao, published under the L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) 1.3c

## Fonts

*Mercury Text G4 Roman*

© 1996, 2007 Hoefler & Co.

*Whitney-Medium*

© 1996, 2009 Hoefler & Frere-Jones.

## Repository

⌚ See GitHub repository.

# 1 Package options

Here are the options that can be declared using \usepackage:

<u>soft fences</u> <booléen>	(default: false)
This key replaces open brackets with parentheses.	
<u>smart fences</u> <booléen>	(default: true)
This key automatically adapts the direction of the brackets to the presence of $+\infty$ and $-\infty$	
<u>separator</u> <token>	(default: ,)
Controls the character inserted between the interval endpoints.	

## 2 Typeset an interval

---

```
\interval *[(keys)]{(list of endpoints)}
```

---

To typeset an interval, simply use the macro in math mode, entering the endpoints in the form of a list:

L <small>A</small> T <small>E</small> Xcode	Result	
1 \begin{equation} 2   \interval{a, b} 3 \end{equation}	[a, b]	(1)

---

```
open
```

---

```
open right
```

---

```
open left
```

---

To change the direction of the brackets, use the option provided here.

L <small>A</small> T <small>E</small> Xcode	Result	
1 \begin{gather} 2   \interval[open right]{a, b} \\ 3   \interval[open left]{a, b} \\ 4   \interval[open]{a, b} 5 \end{gather}	[a, b[	(2)
	]a, b[	(3)
	]a, b[	(4)

The direction of the brackets also adapts itself to  $-\infty$  and  $+\infty$ , provided that the corresponding option remains enabled.

L <small>A</small> T <small>E</small> Xcode	Result	
1 \begin{equation} 2   \interval[-\infty, +\infty} 3 \end{equation}	]-\infty, +\infty[	(5)

---

```
scaled big, Big, bigg, Bigg
```

---

It is also possible to adjust the brackets/parentheses' size through the `scaled` key, which has the same usage as in the `interval` package.

---

```
LATEXcode
```

---

<pre> 1 \begin{equation} 2   \interval[scaled=bigg]{\frac 3     {1}{2}, \frac{3}{2}} \end{equation} </pre>	$\left[ \frac{1}{2}, \frac{3}{2} \right] \quad (6)$
------------------------------------------------------------------------------------------------------------	-----------------------------------------------------

The starred variant typesets integer intervals by using the `stmaryrd` package.<sup>2</sup> All the keys described above are compatible with these symbols.

L <sup>A</sup> T <sub>E</sub> X code	Result	
<pre> 1 \begin{equation} 2   \interval*[2, 10] \end{equation} </pre>	$[2, 10] \quad (7)$	

---

```

\ointerval *{<open, keys>}{<endpoints>}
\linterval *{<open left, keys>}{<endpoints>}
\rinterval *{<open right, keys>}{<endpoints>}

```

---

Finally, the package provides short hands through these three macros:

L <sup>A</sup> T <sub>E</sub> X code	Result	
<pre> 1 \begin{gather} 2   \ointerval[scaled=bigg]{\frac 3     {1}{2}, \frac{9}{2}} \\ 4   \linterval{3, 8} \\ 5   \rinterval{3, 8} \end{gather} </pre>	$\left[ \frac{1}{2}, \frac{9}{2} \right] \quad (8)$	
	$]3, 8] \quad (9)$	
	$[3, 8[ \quad (10)$	

---

### 3 Interval product, union, and intersection

`intervalx` also makes it easier to typeset product, union, and intersection relations between intervals.

---

<sup>2</sup>See on CTAN

## 3.1 Product

---

\xinterval [ $\langle \text{keys} \rangle$ ] { $\langle *-\text{list} \rangle$ }

---

For the product, the main argument takes the same form as with \interval, but the different intervals are delimited by an asterisk.

**Mnemonic:** The “x” at the beginning of the macro name evokes the product symbol  $\times$ , while the asterisk is a way of denoting it in programming.

L <small>A</small> T <small>E</small> Xcode	Result
1   \begin{equation} 2    \xinterval{2, 10 * 1, 15 * -3, 3    19} \end{equation}	$[2, 10] \times [1, 15] \times [-3, 19]$ (11)

To combine this macro with the keys of \interval, the latter must be specified as a list delimited by semicolons (the comma being already used to separate the different keys). Here is an example to clarify this point.

L <small>A</small> T <small>E</small> Xcode	Result
1   \begin{equation} 2    \xinterval[open right, scaled= Big; open left, scaled=Big]{2, 10 * 1, 15} \end{equation}	$[2, 10] \times [1, 15]$ (12)

## 3.2 Union

---

\uinterval [ $\langle \text{keys} \rangle$ ] { $\langle /-\text{list} \rangle$ }

---

Similarly, interval union is typeset using the vertical bar as a delimiter.

**Mnemonic:** The “u” at the beginning of the macro name evokes the union symbol  $\cup$ , while the vertical bar is a way in programming to denote the logical *or*.

L <small>A</small> T <small>E</small> Xcode	Result

<pre> 1  \begin{equation} 2    \uinterval{2, 10   1, 15} 3  \end{equation} </pre>	$[2, 10] \cup [1, 15]$	(13)
-----------------------------------------------------------------------------------	------------------------	------

### 3.3 Intersection

---

\ninterval [ $\langle keys \rangle$ ] { $\langle \&-list \rangle$ }

---

Again, interval intersection is typeset using the ampersand as a delimiter. The use of keys is also identical to \xinterval and \uinterval.

**Mnemonic:** The “n” evokes the intersection symbol  $\cap$ , with the ampersand denoting the logical *and* in programming.

\LaTeX code	Result	
<pre> 1  \begin{equation} 2    \ninterval{3, 20 &amp; \pi, e^3} 3  \end{equation} </pre>	$[3, 20] \cap [\pi, e^3]$	(14)

---

## 4 Inequalities

---

\ineq \* [ $\langle keys \rangle$ ] { $\langle list \ of \ bounds \rangle$ } [ $\langle variable \rangle$ ]

---

The composition of inequalities is quite similar to that of intervals, with a few differences. The keys open right, open left and open are also available to make it easy to write strict or non-strict inequalities. In addition to the bounds, you can also specify the variable, which defaults to  $x$ . Finally, the starred variant of the macro uses the alternative symbols from amssymb for non-strict inequalities.

\LaTeX code	Result	
-------------	--------	--

---

1	\begin{gather}			
2	\ineq{a, b}\\"			$a \leq x \leq b$ (15)
3	\ineq*[a, b][y]\\"			$a \leq y \leq b$ (16)
4	\ineq[open right]{a, b}\\"			$a \leq x < b$ (17)
5	\ineq[open left]{a, b}\\"			$a < x \leq b$ (18)
6	\ineq[open]{a, b}			$a < x < b$ (19)
7	\end{gather}			

---

## Changelog

**1.0.0** (2025-11-16) — Initial version

## Index

<b>I</b>	<b>R</b>
\ineq . . . . . 6	\interval . . . . . 4
\interval . . . . . 2, 4, 5	
<b>L</b>	<b>S</b>
\linterval . . . . . 4	scaled . . . . . 3
	separator . . . . . 2
	smart_fences . . . . . 2
<b>N</b>	soft_fences . . . . . 2
\ninterval . . . . . 5	
<b>O</b>	<b>U</b>
\ointerval . . . . . 4	\uinterval . . . . . 5
open . . . . . 3	\usepackage . . . . . 2
open_left . . . . . 3	
open_right . . . . . 3	<b>X</b>
	\xinterval . . . . . 4, 5