

# VarTable

VarTable is a package to make variation table, in a simple way

This package is build on top of Cetz

(version : 0.1.0)

## Contents

1 - Introduction .....	1
2 - tabvar function .....	3
2.1 - general description .....	3
tabvar .....	3
Parameters .....	3
variable .....	3
label .....	3
domain .....	4
contents .....	4
table-style .....	4
nocadre .....	4
arrow-mark .....	4
arrow-style .....	4
line-0 .....	5
line-style .....	5
hatching-style .....	5
first-column-width .....	5
first-line-height .....	5
element-distance .....	5
values .....	6
add .....	6
2.2 - The contents parameter .....	7
2.2.1 - Sign rows format .....	7
2.2.1.1 - A classical sign array .....	7
2.2.1.2 - Custom separation bar .....	8
2.2.1.2.1 style of bar .....	8
2.2.1.2.2 type of bar .....	8
2.2.1.3 - Same sign for more than one value of the variable .....	9
2.2.2 - Variation table .....	10
2.2.2.1 - A classical variation array .....	10
2.2.2.2 - Undefined values .....	10
2.2.2.3 - Skip a value .....	12
3 - More complex examples .....	13
3.1 - <u>Γ</u> function on [0; +∞] .....	13
3.2 - A Rational function .....	14
3.3 <u>Hyperbolic</u> function .....	15
3.3 A weird table for a simple polynomial function .....	16

## 1 - Introduction

This package is designed to simplify the creation of variation tables for functions. To do this, it gives

you a typst function, whose parameters are described in detail in this documentation.

**A word of warning:** it's quite normal that during the array creation process, The elements, such as the lines between the various elements, aren't created as they should be. For example, the line between the labels and the rest, which doesn't go all the way to the end.

If you encounter any bugs, please report them on my [GitHub](#).

## 2 - tabvar function

### 2.1 - general description

#### tabvar

Retourne un tableau de variation

#### Parameters

```
tabvar(  
    variable: content,  
    label: array,  
    domain: array,  
    contents: array,  
    table-style: style,  
    nocadre: bool,  
    arrow-mark: mark,  
    arrow-style: style,  
    line-0: bool,  
    line-style: style,  
    hatching-style: tiling,  
    first-column-width: length,  
    first-line-height: length,  
    element-distance: length,  
    values: array,  
    add: content  
)
```

#### variable content

variable est la variable qui contient la variable du tableau (comme  $x$  ou  $t$ )

**Exemple:** si la variable de la fonction est  $t$ , alors :

```
variable : $ t $
```

Default: \$ x \$

#### label array

label est un array qui contient des array de longueur 2, une pour chaque ligne du tableau, dont le premier élément est le titre de la ligne et le second est le type de la ligne: signe (s) ou variation (v)

**Exemple:** pour le tableau de variation de la fonction  $f$ , vous devriez écrire:

```
label : (  
    ([Signe de $f$], "s"), // la première ligne est un tableau de signe  
    ([Variation de $f$], "v") // la seconde ligne est un tableau de variation  
)
```

Default: ()

**domain** array

les valeurs prises par la variable

par exemple, si votre fonction change de signe ou atteind un extremum pour  $x \in \{0, 1, 2, 3\}$  vous devriez écrire :

```
domain: ($0$, $1$, $2$, $3$)
```

Default: ()

**contents** array

le contenu de la table

voir 2.2 pour plus de détaille

Default: ((),)

**table-style** style**Optionelle**

Le style de la table

le type style est définis par Cetz, ainsi je vous recommande de vous référer au [manuelle de Cetz](#).

**Attention:** Si vous ne mettez pas le paramètre de style: mark à none, alors toute les lignes du tableau auront une tête en flèche

Default: (stroke: 1pt + black, mark: (symbol: none))

**nocadre** bool

Pour cacher le cadre externe du tableau

Default: false

**arrow-mark** mark

Le style de la tête de flèche.

**N.B.** le type mark est définis par Cetz

Default: (end: "straight")

**arrow-style** style**Optionelle :**

Le style des flèches.

**Attention:** le paramètre mark est supplémenté par le paramètre arrow-mark

Default: (stroke: black + 1pt)

**line-0** `bool`

**Optionnelle**

si vous voulez changer la bar par défaut dans les tableaux de signe, pour une bar avec un zéro en centre

Default: `false`

**line-style** `style`

**Optionnelle**

Si vous voulez le style de toutes les bars de séparation entre les signes

Default: `(stroke: black + 1pt)`

**hatching-style** `tiling`

**Optionnelle**

le style des hachures s'il y a des zones hachurées

Default: `tiling(size: (30pt, 30pt)) [`

```
#place(line(start: (0%, 100%), end: (100%, 0%), stroke: 2pt))
#place(line(start: (-100%, 100%), end: (100%, -100%), stroke: 2pt))
#place(line(start: (0%, 200%), end: (200%, 0%), stroke: 2pt))
```

`]`

**first-column-width** `length`

**Optionnelle**

change la largeur de la première colonne

Default: `none`

**first-line-height** `length`

change la hauteur de la première ligne (celle du domaine et de la variable)

Default: `none`

**element-distance** `length`

**Optionnelle**

change la distance entre deux éléments

Default: `none`

**values** array

pour ajouter des valeurs entre deux valeurs prè-définis

Default: ((),)

**add** content

Pour ajouter plus d'éléments via Cetz

Default: ()

## 2.2 - The contents parameter

The contents parameter must be an array with one element per line (per label)

Each element is itself an array with one element per column, with a different format for either sign or variation rows

### 2.2.1 - Sign rows format

Should contain as much element as the domain less one (one per interval) + one optional end bar style element

Each element is either of these form (can be mixed on a same line):

() – Empty : extend previous cell

body – Simple body such as `$+$` or `$-$`

(body, bar\_style) – to specify an optional style for the previous bar, with one of "`|`" (simple bar), "`||`" (double bar) or "`0`" (bar with a zero)

NB: the `line-0` parameter change the default bar style to "`|`"

The optional last element is "`||`"

#### 2.2.1.1 - A classical sign array

A sign array must contain contents like `$+$` or `$-$`, but you can put anything else.

##### Example:

A normal sign table:

```
#tabvar(
  init: (
    variable: $t$,
    label: ([sign], "Sign"),),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (($+$, $-$, $+$),),
)
```

$t$	2	4	6	8
sign		+	-	+

More complex usage:

```
#tabvar(
  init: (
    variable: $t$,
    label: ([sign], "Sign"),),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      "Hello world !",
      $-$,
      $3 / 2 $)
  ),
)
```

$t$	2	4	6	8
sign	Hello world!		-	$\frac{3}{2}$

**Note:** on the second example the table is squeezed with the scale function)

### 2.2.1.2 - Custom separation bar

#### 2.2.1.2.1 style of bar

you can modify the style of the bars (note that this modifies all the default ones, not the others, see 2.2.1.2.2).

the style of the bar is a dictionary, of the type “style” defined by Cetx as said earlier.

To make it simple if you want to modify only the stroke of the bar, you just have to put `stroke: your_stroke` in brackets.

For more complex applications, please refer to the Cetx manual.

#### Example

```
#tabvar(
  line-style: (
    stroke: (paint: red, dash: "dashed")
  ),
  init: (
    variable: $t$,
    label: ([sign], "Sign")),
  ),
  domain: ($2$, $4$, $6$,),
  contents: (
    ($+$, $-$),
  ),
)
```

$t$	2	4	6
sign	+		-

#### 2.2.1.2.2 type of bar

For all signs except the first one, instead of putting the sign directly, you can put a couple, whose first element defines the previous bar’s type.

There are 3 different types of bar:

- “|” : a simple bar
- “0” : a bar with a 0 on the center
- “||” : a double bar, like for the undefined values

**NB :** the marks-line parameter has no effect on these bars

#### Example :

```
#tabvar(
  init: (
    variable: $t$,
    label: ([sign], "Sign")),
  ),
  domain: ($2$, $4$, $6$, $8$, $10$),
  contents: (
    (
      $+$,
      ("|", $-$),
      ("0", $-$),
      ("||", $+$)
    ),
  ),
)
```

$t$	2	4	6	8	10
sign	+	-	0	-	+

**Note :** The `line-0` parameter is to default lines to “0” type or “|” type

If you want a double line at the start of the table, you can use a double bar "||" on the very first sign. If you want it at the end, you can add this element "||" at the end of sign array

**Example:**

```
#tabvar(
    init: (
        variable: $t$,
        label: ([sign], "Sign")),
    ),
    domain: ($2$, $4$, $6$, $8$),
    contents: (
        (
            ("||", $+$),
            $-$,
            "||"
        ),
    ),
)
```

$t$	2	4	6
sign	+	-	

### 2.2.1.3 - Same sign for more than one value of the variable

For this, it is pretty easy, instead of putting the sign directly, you can put an empty couple

**Example:**

```
#tabvar(
    line-0: true,
    init: (
        variable: $t$,
        label: (
            ([sign], "Sign"),
        ),
    ),
    domain: ($2$, $4$, $6$, $8$),
    contents: (
        ($+$, (), $-$),
    ),
)
```

$t$	2	4	6	8
sign		+	0	-

## 2.2.2 - Variation table

Should contains as much elements as the domain

Each element is in ether of these forms :

- () to extend the previous arrow
- (position, body) with position being one of top, center or bottom
- (pos1, pos2, "||", body1, body2) to put in 2 value separated by an undefined value (double bar)
- (pos, "||", body) shorthand for (pos, pos, "||", body, body) (see previous format)

### 2.2.2.1 - A classical variation array

A variation array must contain couple with in first position, the element position, and in second position, whatever you want as long as it's of the contents type.

The position can be top, center or bottom, but no other type of alignment

**Example :**

```
#tabvar(  
  
    init: (  
        variable: $t$,  
        label: ([variation], "Variation"),),  
    ),  
    domain: ($2$, $4$, $6$, $8$),  
    contents: (  
        (  
            (top, $3$),  
            (bottom, $0$),  
            (center, $1$),  
            (top, $4$),  
        ),  
    ),  
)
```

$t$	2	4	6	8
variation	3	0	1	4

### 2.2.2.2 - Undefined values

If your function is not defined on some values like  $f(x) = \frac{1}{x}$  for  $x = 0$ , you certainly want to put a double line meaning that the function is undefined on this value, and you can !

\* For each values of domain except the start and the end.

The array of one value should look like (pos1, pos2, "||", contents1, contents2) where :

- pos1 and 2 is top, center, bottom and pos1 is for the placement of contents1 similary for pos2
- "||" is to precise the value is undefined
- contents1 and 2 is type of contents and contents1 one is for before the double bar and contents2 for after

**Example :**

```
#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      (top, $3$),
      (bottom, top, "||", $0$, $2$),
      (bottom, $1$),
    ),
  ),
)
```

<i>t</i>	2	4	6
variation	3	2	1

The diagram shows three arrows originating from the word 'variation' and pointing to the values 3, 2, and 1 respectively, which are aligned with the t values 2, 4, and 6.

If **pos1** and **pos2** is same, you can just fill in one instead of two,

In the same way if **contents1** and **contents2** is same, you can also enter just one

### Example:

Instead of (top, top, "||", \$0\$, \$0\$) you can use (top, "||", \$0\$)

```
#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      (top, $3$),
      (bottom, "||", $0$, $1$),
      (top, center, "||", $2$),
      (top, "||", $3$),
      (bottom, $1$),
    ),
  ),
)
```

<i>t</i>	2	4	6	8	9
variation	3	1	2	3	1

The diagram shows five arrows originating from the word 'variation' and pointing to the values 3, 1, 2, 3, and 1 respectively, which are aligned with the t values 2, 4, 6, 8, and 9.

\* For the first and the end values

It a basic array but with "||" this parameter at the array's center

For example (top, "||", \$3\$)

```
#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      (top, "||", $3$),
      (bottom, $1$),
      (top, "||", $2$),
    ),
  ),
)
```

<i>t</i>	2	4	6
variation	3	1	2

The diagram shows three arrows originating from the word 'variation' and pointing to the values 3, 1, and 2 in the 't' column. The arrow to 3 is diagonal down-right, the arrow to 1 is vertical down, and the arrow to 2 is diagonal up-right.

### 2.2.2.3 - Skip a value

When you want to use several functions in the same table, you will probably want to skip some values, to do this, as with sign arrays, you must create an empty array

**Example:**

```
#tabvar(
  arrow-mark: (end: ">", stroke: red),
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
  ),
  domain: ($2$, $4$, $6$),
  contents: (
    (
      (top, "||", $3$),
      (),
      (bottom, $2$),
    ),
  ),
)
```

<i>t</i>	2	4	6
variation	3		2

The diagram shows two arrows originating from the word 'variation' and pointing to the values 3 and 2 in the 't' column. The arrow to 3 is diagonal down-right, and the arrow to 2 is diagonal up-right. There is no arrow for the value 4.

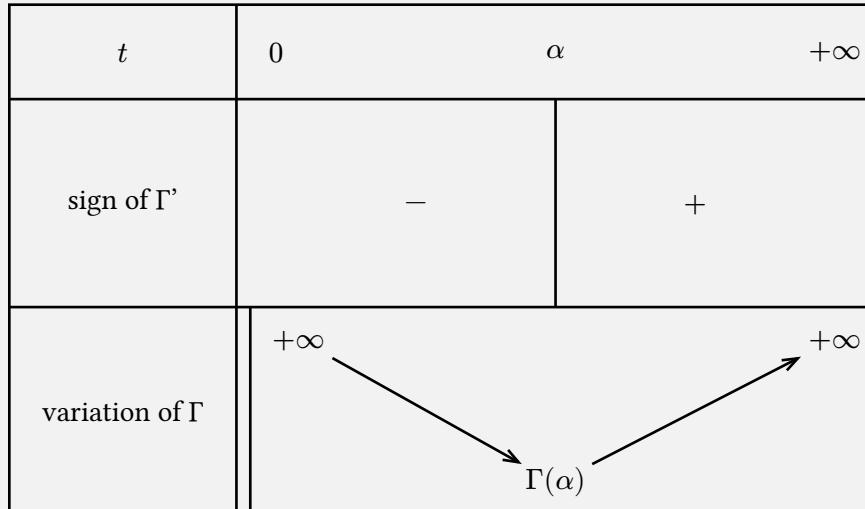
### 3 - More complex examples

Here is a little bundle of what you can do with the package.

#### 3.1 - $\Gamma$ function on $[0; +\infty]$

Where it takes a minimum on  $[0; +\infty[$  for  $x = \alpha$

```
#tabvar(  
  init: (  
    variable: $t$,  
    label: ([sign of #sym.Gamma], "Sign"),  
    ([variation of #sym.Gamma], "v"),  
  ),  
,  
  domain: ($0$, $alpha $, $+oo $),  
  contents: (  
    ($-$, $+$),  
    (  
      (top, "||", $+oo$),  
      (bottom, $Gamma(alpha)$),  
      (top, $+oo$),  
    ),  
  ),  
)
```



## 3.2 - A Rational function

$$\text{Take } f(x) = \frac{4x^2+12x+29}{4(x^2+3x+2)}$$

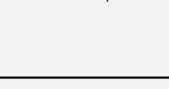
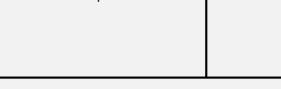
So we have  $f'(x) = \frac{-2x-3}{16(x^2+3x+2)^2}$

And finally, we get:

**Code:**

```
#tabvar(
    init: (
        variable: $t$,
        label: (
            ([sign of $f$], "Sign"),
            ([variation of $f$], "Variation"),
        ),
    ),
    domain: ($ -oo $, $ -2 $, $ -3 / 2 $, $ -1 $, $ +oo $),
    contents: (
        ($+$, ("||", $+$), $-$, ("||", -$)),
        (
            (bottom, $1$),
            (top, bottom, "||", $+oo$, $-oo$),
            (top, $-20$),
            (bottom, top, "||", $-oo$, $+oo$),
            (bottom, $1$),
        ),
    ),
),
)
```

#### - Result:

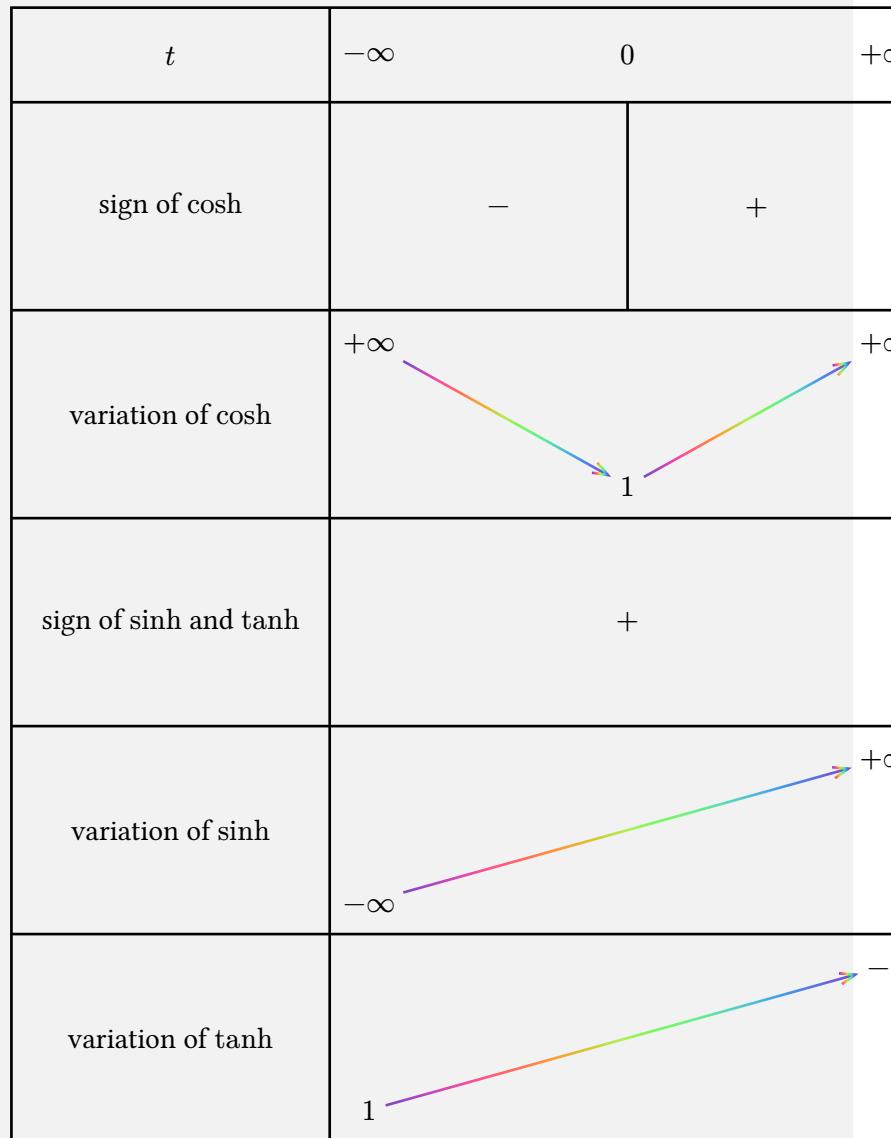
$t$	$-\infty$	$-2$	$-\frac{3}{2}$	$-1$	$+\infty$
sign of $f'$	+	+	-	-	
variation of $f$					

### 3.3 Hyperbolic function

Code:

```
#tabvar(
  arrow: "| -harpoon",
  stroke-arrow:
gradient.linear(..color.map.rainbow),
  marks-line: "...",
  init: (
    variable: $t$,
    label: (
      ([sign of $cosh$], "Sign"),
      ([variation of $cosh$], "Variation"),
      ([sign of $sinh$ and $tanh$], "Sign"),
      ([variation of $sinh$], "Variation"),
      ([variation of $tanh$], "Variation"),
    ),
  ),
  domain: ($ -oo $, $ 0 $, $ +oo $),
  contents: (
    ($-$, $+$),
    (
      (top, $+oo$),
      (bottom, $1$),
      (top, $+oo$),
    ),
    ($+$, ()),
    (
      (bottom, $-oo$),
      (),
      (top, $+oo$),
    ),
    (
      (bottom, $1$),
      (),
      (top, $-1$),
    ),
  ),
)
```

Result:



### 3.3 A weird table for a simple polynomial function

Take  $g(t) = t^2 - t^3$

So, we have  $g'(t) = 2t - 3t^2$

And has local extrema for  $x = 0$  and  $x = \frac{2}{3}$

Code:

```
#tabvar(
  stroke: 5pt + red,
  arrow: "X-*-<>",
  arrow-style: (stroke :purple + 1.4pt),
  marks-line: "<- ->",
  init: (
    variable: $t$,
    label: (
      ([sign of $g'$], "Sign"),
      ([variation of $g$], "Variation"),
    ),
  ),
  domain: ($ -oo $, $ 0 $, $ 2 / 3 $, $ +oo $),
  contents: (
    ($-$, ("|", $+$), $-$),
    (
      (top, $+oo$),
      (bottom, $0$),
      (center, $ 4 / 27 $),
      (bottom, $-oo$),
    ),
  ),
)
)
```

Result:

$t$	$-\infty$	$0$	$\frac{2}{3}$	$+\infty$
sign of $g'$	—	+	0	—
variation of $g$	$+\infty$	0	$\frac{4}{27}$	$-\infty$