

L'extension `tblvar`*

Antoine Missier
`antoine.missier@ac-toulouse.fr`

1^{er} juillet 2019

Table des matières

1	Introduction	1
2	Utilisation	2
2.1	Tableaux de variations simples	2
2.2	Tableaux de signes avec barres de séparation	4
2.3	Doubles barres et discontinuités	5
2.4	Valeurs remarquables	9
2.5	Zones interdites	12
3	Le code	18
3.1	Extensions requises et options	18
3.2	Les paramètres généraux	18
3.3	Les commandes graphiques <code>PSTricks/TikZ</code>	19
3.4	Longueurs et compteurs internes	20
3.5	Les environnements <code>tblvar</code> et <code>tblvar*</code>	21
3.6	La commande <code>\variations</code>	22
3.7	Les commandes de positionnement	23
3.8	Barres, discontinuités et valeurs remarquables	24
3.9	Zones interdites	26

1 Introduction

Cette extension permet de construire des tableaux de variation (et de signes) de manière simple et intuitive. Bien que plusieurs autres extensions soient déjà dédiées à cette tâche¹, nous avons voulu produire des tableaux de manière automatisée, y compris pour des tableaux complexes, mais avec de nombreuses possibilités

*Ce document correspond à `tblvar` v1.2, dernière modification le 01/07/2019.

1. Mentionnons `tabvar` de Daniel Flipo et `variations` de Christian Obrecht ou, plus complexes, `tablor` de Guillaume Connan et `tkz-tab` de Alain Matthes.

de réglages et d'ajustements personnels et une haute qualité graphique, en particulier pour le dessin des flèches, en exploitant les fonctionnalités des extensions graphiques PSTricks (plus exactement `pst-node`) ou TikZ pour définir un *graphe* de *nœuds* reliés par des flèches, venant se « superposer » au tableau lui-même.

Le parti pris ici est d'utiliser la même syntaxe que les environnements `array` (ou `tabular`) en laissant à L^AT_EX le soin de faire la composition du tableau et à l'extension graphique (PSTricks ou TikZ) celui de réaliser le dessin des flèches, automatisé et sans intervention de l'utilisateur. On a simplement besoin de préciser, à l'intérieur d'une commande `\variations`, ce qui est en haut et ce qui est en bas.

L'extension possède deux options utilisées pour le tracé automatique des flèches : `pstricks` (par défaut) ou `tikz` (`\usepackage[tikz]{tblvar}`). Pour l'option `pstricks`, il faut compiler avec L^AT_EX + `dvips` + `ps2pdf` ; pour l'option `tikz`, il faut compiler DEUX fois avec pdfL^AT_EX (la première fois les flèches ne sont pas correctement dessinées). Si on veut compiler avec l'option `pstricks` après avoir compilé avec `tikz`, il faut supprimer le fichier `.aux`, sinon la compilation plante.

Cette documentation donne une galerie d'exemples et décrit les commandes fournies.

2 Utilisation

2.1 Tableaux de variations simples

Un tableau de variation (ou de signes) se définit par un environnement `tblvar` qui *doit être en mode mathématiques* (comme `array`).

`\haut` Voici un premier tableau tout simple avec les commandes de positionnement
`\bas` naïves `\haut`, `\bas` et `\mil` (milieu).
`\mil`

x	-5	-1	2	3	5
$f'(x)$	+	0	-	0	-
$f(x)$	0	↗ 2	↘ 1	↗ 4	↘ -3

```

\[
\begin{tblvar}{4}
\hline
x & -5 & -1 & 2 & 3 & 5 \\
\hline
f'(x) & + & 0 & - & 0 & - \\
\hline
\variations{ \mil{f(x)} & \bas{0} & \haut{2} & \bas{1} & \haut{4} & \bas{-3} }
\hline
\end{tblvar}
\]
```

`tablvar` L'argument obligatoire de l'environnement `tablvar` correspond au nombre d'intervalles du tableau c'est-à-dire aussi au nombre de flèches (ici 4). Les délimiteurs `&` correspondent aux changements de colonnes comme pour l'environnement `array`. Outre la 1ère colonne de légendes, il y a deux types de colonnes : des colonnes de *valeurs*, de largeur variable en fonction de leur contenu et centrées, et des colonnes *intervalles* de largeur fixe (2.5em par défaut).

`tablvar[⟨largeur⟩]` L'environnement `tablvar` possède un paramètre optionnel pour modifier la largeur des colonnes intervalles. Voici un exemple avec un seul intervalle utilisant l'argument optionnel `4em` pour allonger la largeur de l'intervalle qui sinon apparaîtrait très étroit.

```
\[
\begin{tablvar}[4em]{1}
  \hline
  x & -5 && +\infty \\
  \hline
  \variations{\mil{f(x)} & \haut{2} &&
    \bas{-1} }
  \hline
\end{tablvar}
\]
```

x	-5	$+\infty$
$f(x)$	2	-1

`\pos` Les commandes `\haut` et `\bas` ne sont en fait que des alias d'une commande plus générale qui est `\pos[⟨opt⟩]{⟨ligne⟩}{⟨valeur⟩}`. Celle-ci place la *⟨valeur⟩* sur la *⟨ligne⟩* indiquée en créant un nœud pour les flèches. La ligne des x et les lignes contenant dérivée ou tableau de signe ont pour indice 0. La partie `variations` contient par défaut 3 lignes numérotées 1, 2, 3 *du haut vers le bas* (dans le sens de l'écriture et de la construction du tableau). Le paramètre optionnel `⟨opt⟩`² permet d'ajuster le positionnement des flèches : `c` (centered, par défaut), `b` (bottom) ou `t` (top). Comparer le positionnement des flèches dans les deux exemples suivants.

`\variations` C'est la commande `\variations` qui, une fois les nœuds définis, construit ensuite automatiquement les flèches.

`\pos*` La commande `\mil`, utilisée pour la première colonne contenant $f(x)$, est un alias reposant sur la commande plus générale `\pos*{⟨ligne⟩}{⟨valeur⟩}` qui se comporte comme `\pos` mais ne crée pas de nœud pour les flèches.

```
\begin{tablvar}{2}
  \hline
  x & 0 & & \frac{1}{2} & & 1 \\
  \hline
  f'(x) & + & & 0 & & - \\
  \hline
  \variations{\mil{f(x)} & \bas{2} &&
    \haut{f(\frac{1}{2})} && \bas{0}
  }
  \hline
\end{tablvar}
```

x	0	$\frac{1}{2}$	1
$f'(x)$	+	0	-
$f(x)$	2	$f(\frac{1}{2})$	0

2. Ce paramètre optionnel n'est pas implémenté et est sans effet pour l'option `tikz`.

partie variations. Ici $f(x) = -4x^3 + 3x^2 + 18x - 3$ et $f'(x) = 6(x+1)(-2x+3)$.

x	$-\infty$	-1	$\frac{3}{2}$	$+\infty$
$x+1$	$-$	0	$+$	$+$
$-2x+3$	$+$	$+$	0	$-$
Signe de $f'(x)$	$-$	0	0	$-$
Variations de f	$+\infty$	-14	$\frac{69}{4}$	$-\infty$

Les bornes des intervalles ont été placées dans des `\makebox[⟨largeur⟩]{\$ \$}`, sur la ligne des x , pour que les colonnes soient plus équilibrées (ceci est affaire de goût).

```
\begin{tblvar}{3}
  \hline
  x &  $-\infty$  &  $\frac{3}{2}$  &  $+\infty$  \\
  \hline
  x+1 & - & 0 & + \\
  \hline
  -2x+3 & + & + & 0 \\
  \hline
  \mbox{Signe de }f'(x) & - & 0 & 0 \\
  \hline
  \variations{ \mbox{Variations de }f} & \haut{+\infty} & \bas{-14} & \haut{\frac{69}{4}} \\
  \hline
\end{tblvar}
```

2.3 Doubles barres et discontinuités

`\bb` Une double barre s'obtient avec la commande `\bb`. Pour ne pas tracer de flèche entre deux nœuds (en particulier pour éviter de traverser une double barre), on place la commande `\discont` entre les deux.

x	$-\infty$	0	$+\infty$
$-\frac{1}{x}$	$+$	$-$	
$\frac{1}{x^2}$	0	$+\infty$	0

La commande `\dfrac` provient de l'extension `amsmath` ou peut être redéfinie par `\newcommand*\dfrac[2]{\displaystyle\frac{#1}{#2}}`.

```
\begin{tblvar}{2}
\hline
x &  $-\infty$  &  $0$  &  $+\infty$  \\
\hline
 $-\frac{1}{x}$  &  $+$  &  $\infty$  &  $-$  \\
\hline
\variations{\mil{\dfrac{1}{x^2}} & \bas{0} & \haut{+\infty} \bb
\discont \haut{+\infty} & \bas{0}
}
\hline
\end{tblvar}
```

La syntaxe ci-dessus pose problème lorsque les limites à gauche et à droite d'une valeur interdite se trouvent sur des lignes différentes, ou que leur contenu n'a pas la même largeur, car alors la double barre ne serait plus au centre de sa colonne et se trouverait décalée d'une ligne à l'autre.

Une première solution est d'interrompre la double barre pour placer les limites qui resteront alors centrées au milieu de la colonne (mais il faut aimer).

x	$-\infty$	-1	0	1	$+\infty$
$\frac{1}{1-x^2}$	0	$+\infty$	1	$+\infty$	0
		$ $		$ $	
		$-\infty$		$-\infty$	

```
\begin{tblvar}[3em]{4}
\hline
x &  $-\infty$  &  $-1$  &  $0$  &  $1$  &  $+\infty$  \\
\hline
\variations{\mil{\dfrac{1}{1-x^2}} & \haut{0} &
\bas{-\infty}\mil{\bb}\discont\haut{+\infty} & \bas{1} &
\haut{+\infty}\mil{\bb}\discont\bas{-\infty} & \haut{0}
}
\hline
\end{tblvar}
```

`\bblim` La commande `\bblim` résout le problème en gardant, sur chaque ligne, la double barre centrée dans sa colonne, avec des limites à gauche et à droite quelconques et quelles que soient les lignes où celles-ci sont positionnées. Elle prend 4 paramètres qui sont respectivement *ligne* et *limite* à gauche puis à droite de la double

barre. Elle trace la double barre, place les limites et gère la discontinuité.

x	$-\infty$	-1	0	1	$+\infty$
$\frac{1}{1-x^2}$	0	$+\infty$	1	$+\infty$	0

```

\begin{tblvar}[2em]{4}
\hline
x & -\infty & -1 & 0 & 1 & +\infty \\
\hline
\variations{\mil{\dfrac{1}{1-x^2}} & \haut{0} & & & & & \\
& \bblim{3}{-\infty}{1}{+\infty} & \bas{1} & & & & \\
& \bblim{1}{+\infty}{3}{-\infty} & & & & \haut{0} & \\
}
\hline
\end{tblvar}

```

Un autre exemple avec des limites de largeur assez différentes (placées ici sur la même ligne).

x	0	$\frac{\pi}{6}$	$\frac{\pi}{2}$	π
$f'(x)$	$+$	0	$-$	$-$
$f(x)$	$f(0)$	$f(\frac{\pi}{6})$	$-\infty$	$f(\pi)$

La colonne de valeurs du $\frac{\pi}{2}$ étant très large, la position des signes « - » a été légèrement décalée avec la commande $\text{\LaTeX} \text{\hspace}$ qui produit un espacement horizontal positif ou négatif.

```

\begin{tblvar}{3}
\hline
x & 0 & & \frac{\pi}{6} & & \frac{\pi}{2} & & \pi \\
\hline
f'(x) & + & 0 & \hspace{1em} - & \bb & \hspace{-1em} - & & \\
\hline
\variations{\mil{f(x)} & \pos{3}{f(0)} & & \pos[t]{1}{f(\frac{\pi}{6})} & & & & \\
& & \bblim{3}{-\infty}{3}{0} & & \pos[t]{1}{f(\pi)} & & & \\
}
\hline
\end{tblvar}

```

tblvar* L'environnement **tblvar*** sert à gérer correctement le positionnement des

doubles barres lorsqu'elles se trouvent aux extrémités. La différence avec `tblvar` est que les colonnes de valeurs des extrémités ne sont plus centrées mais alignées à gauche pour le 1ère et à droite pour la dernière.

x	0	1	$+\infty$
$\ln x - x$		-1	
	$-\infty$	↗ ↘	$-\infty$

x	0	1	$+\infty$
$\ln x - x$		-1	
	$-\infty$	↗ ↘	$-\infty$

On observera l'utilisation de `\pos*` dans le second tableau, pour tracer une double barre, sauf sur la ligne 3 où on place $-\infty$.

```
\begin{tblvar*}{2}
  \hline
  x & 0 & 1 & +\infty \\
  \hline
  \variations{ \mil{\ln x -x} & \bb \pos{3}{-\infty} & \pos{1}{-1} & \pos{3}{-\infty} }
  \hline
\end{tblvar*}

\begin{tblvar*}{2}
  \hline
  x & 0 & 1 & +\infty \\
  \hline
  \variations{ \mil{\ln x -x} & \pos*{1}{\bb} \pos*{2}{\bb} & \pos{3}{-\infty} & \pos{1}{-1} & \pos{3}{-\infty} }
  \hline
\end{tblvar*}
```

Si par contre, à l'autre extrémité du tableau, les valeurs ont des largeurs assez différentes, leur alignement non centré peut être corrigé en utilisant `\hspace`. Ci-dessous, de l'espace a été ajouté à droite du 1.

```
\begin{tblvar*}[3em]{1}
  \hline
  x & 0 & +\infty \\
  \hline
  \variations{ \mil{1-\dfrac{1}{x}} & \bb & \bas{-\infty} & \haut{1\hspace{0.5em}} }
  \hline
\end{tblvar*}
```

x	0	$+\infty$
$1 - \frac{1}{x}$		1
	$-\infty$	↗

`\tblvarinit` Enfin, pour personnaliser totalement la définition des colonnes, on peut revenir à l'environnement `array` plutôt que d'utiliser `tblvar` (il faut dans ce cas appeler `\tblvarinit` juste avant `\begin{array}`). Ce qui est essentiel dans cette extension est la macro `\variations` avec les commandes de positionnement et non

l'environnement `tblvar` qui n'est qu'un `array` dans lequel on a fixé les définitions de colonnes (et initialisé des variables).

2.4 Valeurs remarquables

Nous appellerons *valeur remarquable*, un valeur supplémentaire que l'on place dans un tableau de variation et qui ne correspond pas à un extremum.

`\vr` Une première approche est de laisser les flèches de variations passer à travers ces valeurs remarquables. Pour chaque valeur remarquable, on place une commande `\vr` sur la ligne des x et une commande `\vr` dans la partie variations. Les valeurs remarquables seront automatiquement reliées par des pointillés (tracés réalisés dans la commande `\variations`).

Voici un exemple avec le logarithme népérien où la valeur remarquable est placée dans la colonne intervalle.

```
\begin{tblvar*}[5em]{1}
  \hline
  x & 0 & \vr{1} & & +\infty \\
  \hline
  \variations{ \mil{\ln x} &
    \bb \pos{3}{-\infty} &
    \vr{0} & \pos{1}{+\infty} }
  \hline
\end{tblvar*}
```

x	0	1	$+\infty$
$\ln x$		⋮	$+\infty$
		0	
	$-\infty$		

Un autre exemple avec deux valeurs remarquables, placées cette fois dans des colonnes valeurs.

x	$-\infty$	0	1	$+\infty$
$\exp x$		⋮	⋮	$+\infty$
		1	e	
	0			

```
\renewcommand{\tblvarstretch}{1.2}
\begin{tblvar}[1.5em]{3}
  \hline
  x & -\infty & \vr{0} & \vr{1} & & +\infty \\
  \hline
  \variations[4]{ \pos*{2}{\vdecal{-1.5ex}{\exp x}} & \pos{4}{0} &
    \vr{3}{1} & \vr{\mathrm{e}} & \pos{1}{+\infty}
  }
  \hline
\end{tblvar}
```

`\variations[nblignes]` Ci-dessus, la partie variations a été composée sur 4 lignes grâce au paramètre optionnel de la commande `\variations`. Les lignes sont alors numérotées de 1 (haut) à 4 (bas).

- `\vr[⟨ligne⟩]` La commande `\vr` possède elle aussi un argument optionnel qui est la ligne sur laquelle placer la valeur remarquable, lorsqu'il s'agit de la partie variations (2 par défaut).
- `\tblvarstretch` À cause des 4 lignes de variations, la hauteur de lignes du tableau a été réduite en modifiant le facteur `\tblvarstretch` (fixé à 1.4 par défaut), modification locale car placée à l'intérieur de l'environnement `math`³.
- `\vdecal` Enfin la légende `exp x` a été placée en ligne 2 mais décalée vers le bas, pour qu'elle se trouve centrée verticalement sachant qu'il y a 4 lignes de variations. Ceci s'obtient grâce à la commande `\vdecal{⟨decal⟩}{⟨contenu⟩}` où la valeur du décalage peut être positive (vers le haut) ou négative (vers le bas).

On peut préférer que les flèches de variations s'interrompent sur les valeurs remarquables qui sont alors des nœuds. Voici un tableau qui présente simultanément les deux manières de traiter les valeurs remarquables. La taille des flèches s'ajuste automatiquement. Bien entendu, on peut aussi ne pas tracer les pointillés mais placer quand même une valeur remarquable sur la flèche avec `\mil{0}`.

x	0	α_1	$\frac{\pi}{6}$	α_2	$\frac{\pi}{2}$
$f'(x)$	+	⋮	+	0	-
$f(x)$	$f(0)$	⋮	$f(\frac{\pi}{6})$	0	$-\infty$

Si on choisit l'option `[t]` pour le max en ligne 1, il est alors nécessaire de choisir l'option `[b]` en ligne 3 pour conserver un positionnement correct du 0 sur la flèche.

```

\begin{tblvar*}{4}
  \hline
  x & 0 & \vr{\alpha_1} & \frac{\pi}{6} & \alpha_2 & \frac{\pi}{2} \\
  \hline
  f'(x) & + & & + & 0 & - \\
  \hline
  \variations{\mil{f(x)} & \pos[b]{3}{f(0)} & \vr{0} & & & \\
  \pos[t]{1}{f(\frac{\pi}{6})} & & \pos{2}{0} & & \pos[b]{3}{-\infty} \\
  }
  \hline
\end{tblvar*}

```

La commande `\vr` peut également servir à représenter des discontinuités particulières, voir l'exemple ci-dessous avec la fonction définie par

$$f(x) = \begin{cases} \frac{\sin x}{x} & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases} .$$

3. On peut utiliser ce paramètre pour agrandir la hauteur des lignes mais l'effet sera global pour tout le tableau. Si l'on souhaite agrandir spécifiquement une ligne on peut utiliser la commande `\vstrut` de l'extension `spacingtricks`. Malheureusement un ajustement automatique comme proposé dans l'extension `arraycols` ne fonctionne pas, à cause des doubles barres.

```

\begin{tblvar}{2}
\hline
x &  $-\pi$  &  $0$  &  $\pi$  \\
\hline
\variations{\mil{f(x)} & \bas{0} & & \\
\haut{ 1 \hspace{0.2em} } \discont & & & \\
\vr[3]{0} \haut{ \hspace{0.2em} 1 } & & & \\
& & & \bas{0} \\
}
\hline
\end{tblvar}

```

x	$-\pi$	0	π
$f(x)$	0	1	0

Voici un dernier exemple, traité de deux manières différentes, avec des valeurs remarquables dans deux tableaux conjoints. Avec des nœuds c'est assez simple.

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
$\cos x$	1	0	-1	0	1
$\sin x$	0	1	0	-1	0

```

\begin{tblvar}{4}
\hline
x & 0 &  $\frac{\pi}{2}$  &  $\pi$  &  $\frac{3\pi}{2}$  &  $2\pi$  \\
\hline
\variations{ \mil{\cos x} & \haut{1} & \pos{2}{0} & \bas{-1} & & \\
\pos{2}{0} & & \haut{1} & & & } \\
\hline
\variations{ \mil{\sin x} & \bas{0} & \haut{1} & \pos{2}{0} & & \\
\bas{-1} & & \haut{0} & & & } \\
\hline
\end{tblvar}

```

`\noeud` Par contre, pour traiter ici les valeurs remarquables avec des pointillés, la
`\vrconnect` commande `\vr` ne permet pas connecter les valeurs sur x avec celles du sinus dans
le 2^e tableau, mais on peut le faire « à la main » en appelant les macros `\noeud`
et `\vrconnect`.

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
$\cos x$	1	0	-1	0	1
$\sin x$	0	1	0	-1	0

```

\begin{tblvar}{4}
\hline
x & 0 & \noeud{X1}{\frac{\pi}{2}} & \noeud{X2}{\pi} & \noeud{X3}{\frac{3\pi}{2}} & 2\pi \\
\hline
\variations{ \mil{\cos x} & \haut{1} & \mil{0} & \bas{-1} & \mil{0} & \haut{1} }
\hline
\variations{ \mil{\sin x} & \bas{0} & \haut{\noeud{Y1}{1}} & \mil{\noeud{Y2}{0}} & \bas{\noeud{Y3}{-1}} & \haut{0} }
\hline
\end{tblvar}
\vrconnect{X1}{Y1}
\vrconnect{X2}{Y2}
\vrconnect{X3}{Y3}

```

2.5 Zones interdites

Nous abordons pour finir le tracé de zones interdites où la fonction n'est pas définie. On peut dessiner ces zones interdites en couleur ou en hachures.

`\ZIc` Pour une zone interdite en couleur, on place la commande `\ZIc` dans les intervalles que l'on veut colorer (signes ou variations). Un exemple avec la fonction f définie par $f(x) = \frac{1}{\sqrt{x^2-1}}$. La zone grisée arrive toujours au contact des doubles barres.

x	$-\infty$	-1	1	$+\infty$
$-\frac{x}{\sqrt{x^2-1}^3}$	+			-
$\frac{1}{\sqrt{x^2-1}}$	0			0

`\zbox` Plusieurs « interventions » manuelles sont nécessaires. D'abord sur la ligne des x ,

il faut permettre aux bornes de déborder sur la colonne de la zone interdite. Pour cela on les place dans une `\zbox`. La commande `\zbox[⟨pos⟩]{⟨contenu⟩}` affiche son contenu mais considère la largeur comme nulle. Le 1^{er} paramètre (optionnel) est le positionnement dans la boîte : `c` (par défaut), `l` (left) ou `r` (right). Ici `zbox` est aussi nécessaire pour annuler la largeur des boîtes contenant les $+\infty$ sinon la double barre ne serait plus au centre de sa colonne, en ligne 1, ce qui produirait un décalage avec les autres lignes.

La deuxième ligne avec la racine cubique a été agrandie grâce à la commande `\strut`. Enfin, les signes ont été décalés avec `\hspace`, pour améliorer leur centrage par rapport aux flèches.

```
\begin{tblvar}[4em]{3}
\hline
x & -\infty & \zbox{-1} & \zbox{1} & +\infty \\
\hline
-\frac{x}{\sqrt{x^2-1}} & \hspace{-1.5em} + & \bb & \ZIc & \bb & \hspace{1.5em} - & \\
\hline
\variations{\mil{\dfrac{1}{\sqrt{x^2-1}}} & \bas{0} & & & & & \\
\zbox[r]{\haut{+\infty}} & \bb & \ZIc & & & & \\
\bb & \zbox[l]{\haut{+\infty}} & & & & & \\
}
\hline
\end{tblvar}
```

Un autre exemple avec $f(x) = \sqrt{2x^2 - 1}$ où les doubles barres n'apparaissent que sur la dérivée.

x	$-\infty$	$-\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$+\infty$
$\frac{2x}{\sqrt{2x^2-1}}$		-		+
$\sqrt{2x^2-1}$	$+\infty$			$+\infty$
		↘ 0	0 ↗	

```
\begin{tblvar}[3em]{3}
\hline
x & -\infty & \zbox{-\frac{\sqrt{2}}{2}} & \zbox{\frac{\sqrt{2}}{2}} & +\infty \\
\hline
\frac{2x}{\sqrt{2x^2-1}} & - & \bb & \ZIc & \bb & + & \\
\hline
\variations{\mil{\sqrt{2x^2-1}} & \haut{+\infty} & \zbox[r]{\bas{0}} & \ZIc & \zbox[l]{\bas{0}} & \haut{+\infty} \\
}
\hline
\end{tblvar}
```

`\Zih` Voyons à présent la réalisation de tableaux avec zones interdites hachurées. On place la commande `\Zih` dans les intervalles que l'on veut hachurer.

x	$-\infty$	$-\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$+\infty$
$f'(x)$	-			+
$f(x)$	$+\infty$			$+\infty$

Pour les mêmes raisons que précédemment, les `\zbox` sont toujours nécessaires sur la ligne des x , mais plus pour les 0.

```
\begin{tblvar}{3}
\hline
x & -\infty & \zbox{-\frac{\sqrt{2}}{2}} & \zbox{\frac{\sqrt{2}}{2}} & +\infty \\
\hline
f'(x) & - & \bb & \Zih & \bb & + & \\
\hline
\variations{\mil{f(x)} & \haut{+\infty} & \bas{0} & \Zih & \bas{0} & \haut{+\infty} }
\hline
\end{tblvar}
```

`\Zih[hauteur]` Les hachures s'ajustent automatiquement lorsque toutes les lignes ont une hauteur standard. Si par contre des lignes sont plus hautes à cause de leur contenu, on doit ajouter de la hauteur avec `\Zih[hauteur]` (réglage manuel).

x	$-\infty$	-1	1	$+\infty$
$f'(x)$	+			-
$\sqrt{\frac{x-1}{x+1}}$	1			1

On remarquera ici l'utilisation de `\barre` dans la partie variations qui oblige à placer le 0 dans une `\zbox` sans quoi la barre serait décalée après le 0.

```
\begin{tblvar}[3.5em]{3}
\hline
x & -\infty & \zbox{-1} & \zbox{1} & +\infty \\
\hline
f'(x) & \hspace{-1em} + & \bb & \Zih & \bb & \hspace{1em} - & \\
\hline
\variations{ \mil{\sqrt{\dfrac{x-1}{x+1}}} }
\hline
\end{tblvar}
```

```

& \bas{1} && \zbox[r]{\haut{+\infty}} \bb & \ZIh[2ex]
& \zbox{\bas{0}} \barre && \haut{1}
}
\hline
\end{tablvar}

```

Voici un exemple avec deux zones interdites. On considère la fonction f telle que $f(x) = \sqrt{(x^2 - 1)(x^2 - 4)}$.

x	$-\infty$	-2	-1	0	1	2	$+\infty$
$2x$		-		- 0 +		+	
$2x^2 - 5$		+		-		-	+
$f'(x)$		-		+ 0 -		+	
$f(x)$	$+\infty$						$+\infty$

La commande `\ZIh` ne doit pas être placée deux fois dans une même ligne de *signe* car, sauf sur la dernière ligne des variations où elle trace les hachures, son effet est simplement de cumuler la hauteur de la ligne. Et comme toutes les lignes de signes ont le même numéro 0, la commande `\ZIh`, dans la partie signes, ne sait pas sur quelle ligne elle se trouve et cumule la hauteur de ligne à chaque appel.

```

\begin{tablvar}{6}
\hline
x & -\infty && \zbox{-2} && \zbox{-1} && 0 && \zbox{1} && \zbox{2}
&& +\infty \\
\hline
2x & &-& & & - 0 + & & & & + & & \\
\hline
2x^2-5 & &+& & & - & & & & - & & + \\
\hline
f'(x) & &-& & & + 0 - & & & & + & & \\
\hline
\variations{\mil{f(x)} & \haut{+\infty} && \zbox{\bas{0}} \barre
& \ZIh[0.4ex] & \zbox{\bas{0}} \barre && \haut{2} && \zbox{\bas{0}}
\barre & \ZIh[0.4ex] & \zbox{\bas{0}} \barre && \haut{+\infty}
}
\hline
\end{tablvar}

```

`\ZIh*` Lorsqu'on ne fait qu'un tableau de signes (sans variations), il faut utiliser la commande `\ZIh*` pour déclencher le tracé des hachures. Celle-ci doit être placée sur la dernière ligne de chaque bloc de hachures. Sur les lignes précédentes, le `\ZIh` peut être placé n'importe où (une seule fois par ligne).

x	$-\infty$	-2	-1	0	1	2	$+\infty$
$2x$		-		- 0 +			+
$2x^2 - 5$		+		-	-		+
$f'(x)$		-		+ 0 -			+

```

\begin{tblvar}{6}
\hline
x & -\infty & \zbox{-2} & \zbox{-1} & 0 & \zbox{1} & \zbox{2} & +\infty \\
\hline
2x & & \bar{} & \bar{} & \bar{0} & \bar{} & \bar{} & \\
\hline
2x^2-5 & & \bar{} & \bar{} & \bar{} & \bar{} & \bar{} & \\
\hline
f'(x) & & \bar{} & \bar{} & \bar{0} & \bar{} & \bar{} & \\
\hline
\end{tblvar}

```

Si on veut tracer des rectangles de hachures sur des lignes non contiguës, il faut un appel à `\ZIh*` pour chaque rectangle.

x	$-\infty$	-1	1	2	$+\infty$
$\sqrt{x^2 - 1}$		+		0 +	+
$x - 2$		-	-	- 0 +	
$(x - 2)\sqrt{x^2 - 1}$		-		0 - 0 +	

```

\begin{tblvar}{4}
\hline
x & -\infty & \zbox{-1}\phantom{0} & \zbox{1}{1}\phantom{0} & 2 & +\infty \\
\hline
\sqrt{x^2-1} & & \bar{} & \bar{} & \bar{0} & \bar{} & \bar{} & \\
\hline
x-2 & & \bar{} & \bar{} & \bar{} & \bar{} & \bar{0} & \\
\hline
(x-2)\sqrt{x^2-1} & & \bar{} & \bar{} & \bar{} & \bar{} & \bar{} & \\
\hline
\end{tblvar}

```

Ci-dessus, la commande \LaTeX `` sur la ligne des x permet de créer une boîte fantôme de la largeur du 0, pour éviter que la largeur de la colonne valeur

ne soit complètement nulle produisant un chevauchement inélégant des 0 sur les hachures.

`\ZIinit` L'appel à `\ZIinit` sur la dernière ligne permet de ne pas conserver la hauteur cumulée lors du précédent appel à `\ZIh*` (en fait inutile ici car la 1ère zone hachurée ne fait qu'une seule ligne de haut).

`\hachure` On ne peut pas dessiner des zones interdites hachurées sur plusieurs colonnes contiguës avec `\ZIh`, mais on peut alors utiliser la primitive graphique `\hachure` pour définir manuellement le rectangle à hachurer.

x	$-\infty$	-2	$-\sqrt{\frac{5}{2}}$	-1	0	1	$\sqrt{\frac{5}{2}}$	2	$+\infty$	
$2x$		-	-	-	- 0 +	+	+	+		
$2x^2 - 5$		+	+	0 -	-	-	- 0 +	+		
$f'(x)$		-			+	0 -			+	
$f(x)$	$+\infty$			0	2	0			0	$+\infty$

Si l'on ne dispose pas de ligne de signe pour placer la commande `\hachure`, on peut la placer sur la ligne des x ou dans la partie variations mais il faut alors mettre celle-ci dans un `\pos*` sinon elle sera répétée pour chaque ligne des variations.

```
\begin{tblvar}{8}
  \hline
  x & -\infty & \zbox{-2} & \zbox{-\sqrt{\frac{5}{2}}} & \zbox{-1} & 0 & 1 & \zbox{\sqrt{\frac{5}{2}}} & 2 & +\infty \\
  \hline
  2x & & \bar{-} & \bar{-} & \bar{-} & \bar{-} 0 \bar{+} & \bar{+} & \bar{+} & \bar{+} & \\
  \hline
  2x^2-5 & & \bar{+} & \bar{+} & \bar{0} \bar{-} & \bar{-} & \bar{-} & \bar{-} 0 \bar{+} & \bar{+} & \\
  \hline
  f'(x) & & \bar{-} & \hachure{-4em,-12.8ex}{4em,2.6ex} & & \bar{+} & \bar{0} \bar{-} & \hachure{-4em,-12.8ex}{4em,2.6ex} & & \bar{+} \\
  \hline
  \variations{\mil{f(x)} & \haut{+\infty} & \bas{0} & \discont & & & & \bas{0} & \haut{2} & \bas{0} & \discont & & \pos*{3}{\hachure{-4em,14.3ex}{4em,-1.2ex}} & \bas{0} & \haut{+\infty} \\
  } \\
  \hline
\end{tblvar}
```

3 Le code

3.1 Extensions requises et options

```
1 \RequirePackage{array}
2 \RequirePackage{ifthen}
3 \RequirePackage{multido}
4 \RequirePackage{colortbl} % pour \ZIC
5
6 \newboolean{tikz}
7 \DeclareOption{tikz}{\setboolean{tikz}{true}}
8 \DeclareOption{pstricks}{\setboolean{tikz}{false}} % valeur par défaut
9 \ProcessOptions \relax
10
11 \ifthenelse{\boolean{tikz}}{
12   \RequirePackage{tikz}
13   \usetikzlibrary{patterns}
14   \newlength{\tikznode@below}
15   }{
16   \RequirePackage{pst-node}
17 }
```

3.2 Les paramètres généraux

Dans cette section sont présentés tous les paramètres que l'on peut redéfinir pour modifier l'aspect des tableaux, mais qui n'ont en principe pas besoin d'être touchés.

- `\intervalwidth` Définit la largeur par défaut des colonnes « intervalles », valeur prédéfinie à 2.5 em.
- ```
18 \newlength{\intervalwidth}
19 \setlength{\intervalwidth}{2.5em} % largeur des "intervalles"
```
- `\bordercolsep` Définit la largeur de l'espace extérieur des première et dernière colonnes de valeurs. Sa valeur par défaut a été réduite par rapport à la longueur  $\LaTeX$  standard `\arraycolsep` afin que les valeurs des extrémités (souvent des  $\infty$ ) soient plus proches des lignes verticales de début et de fin.
- ```
20 \newlength{\bordercolsep} % largeur de l'espace inter-colonne
21 \setlength{\bordercolsep}{2pt}
```
- `\innercolsep` Définit la largeur entre les colonnes valeurs et les colonnes intervalles du tableau. Cette largeur a été ajustée pour que les zones interdites grisées avec `\cellcolor` arrivent au contact de la double barre.
- ```
22 \newlength{\innercolsep}
23 \setlength{\innercolsep}{4pt}
```
- `maxdiscount` La gestion de discontinuités permettant de ne pas relier les flèches l'une à l'autre se fait grâce à la commande `\discont`. Le compteur `maxdiscount` est fixé à 3 par défaut ; il faut l'augmenter si l'on veut produire un tableau de variation avec plus de 3 discontinuités.

```

24 \newcounter{maxdiscont}
25 \setcounter{maxdiscont}{3} % nb max de discontinuités

\tablvarstretch Permet de régler la valeur relative de l'espace vertical des lignes du tableau. La
valeur par défaut est 1.4 (1 correspondant à la valeur standard d'un environnement
array). Placée dans un environnement math, elle sera locale à cet environnement.
26 \newcommand{\tablvarstretch}{1.4}

\tvbarrewidth Définit l'épaisseur des barres de séparation verticales coupant les 0 d'un tableau
de signe : 0.5 pt par défaut.
27 \newlength{\tvbarrewidth}
28 \setlength{\tvbarrewidth}{0.5pt}

tvbarrecolor Définit la couleur des mêmes barres de séparation verticales.
29 \definecolor{tvbarrecolor}{gray}{0.7}

```

### 3.3 Les commandes graphiques PSTricks/TikZ

Nous présentons ici les commandes graphiques permettant le dessin des flèches, les pointillés des valeurs remarquables, les hachures des zones interdites. Celles-ci sont définies différemment s'il s'agit de l'option `tikz` ou `pstricks` (plus précisément liées à l'extension `pst-node`).

```

\fleche La commande \fleche{<noeud1>}{<noeud2>} possède deux paramètres qui sont
les noms des nœuds à relier. La création des nœuds est obtenue avec la commande
\noeud appelée par \pos et le tracé des flèches réalisé automatiquement par la
commande \variations. Pour modifier l'aspect des flèches on peut redéfinir la
commande.
30 \newcommand*{\fleche}[2]{
31 \ifthenelse{\boolean{tikz}}{
32 \tikz[remember picture,overlay]{\draw[->,>=stealth,
33 line width=0.6pt] (#1) -- (#2);}
34 }{
35 \ncline[arrowsize=2pt 2,arrowinset=0.4,nodesep=3pt,
36 linewidth=0.6pt]{->}{#1}{#2}
37 }
38 }

\vrconnect La commande \vrconnect{<noeud1>}{<noeud2>} relie les nœuds définis par \vr
(valeurs remarquables) et le tracé est réalisé automatiquement par la commande
\variations. Par défaut : lignes en pointillés d'épaisseur 1 pt.
39 \newcommand*{\vrconnect}[2]{
40 \ifthenelse{\boolean{tikz}}{
41 \tikz[remember picture,overlay]{\draw[dotted,line width=1pt]
42 (#1) -- (#2);}
43 }{
44 \ncline[nodesep=5pt,linestyle=dotted,linewidth=1pt]{-}{#1}{#2}
45 }
46 }

```

`\noeud` `\noeud[⟨pos⟩]{⟨noeud⟩}{⟨valeur⟩}` définit les nœuds des flèches et valeurs remarquables ; le 1<sup>er</sup> paramètre optionnel correspond à l’option `t` (top), `b` (bottom) ou `c` (centered, par défaut) permettant d’ajuster la manière dont la flèche arrive sur le nœud (sans effet pour `tikz`) ; le 2<sup>e</sup> paramètre est le nom du nœud (qui est donné automatiquement par les commandes de positionnement) ; le 3<sup>e</sup> paramètre est la valeur affichée dans le tableau.

```

47 \newcommand*{\noeud}[3][c]{
48 \ifthenelse{\boolean{tikz}}{
49 \tikz[remember picture,baseline]{
50 \node[anchor=base,inner sep=0,outer sep=4] (#2) { $\#3$ };
51 } % l'option de placement (#1) n'est pas implémentée pour tikz
52 }{
53 \rnode[#1]{#2}{#3}
54 }
55 }

```

`\hachure` Définition des hachures pour les zones interdites. La macro prend deux arguments qui sont des paires de longueurs, par exemple `\hachure{-3em,12ex}{3em,-1ex}`, représentant les extrémités du rectangle à hachurer par rapport à la position courante où la macro est appelée.

```

56 \newcommand*{\hachure}[2]{
57 \ifthenelse{\boolean{tikz}}{
58 \tikz[remember picture,overlay]{\fill[pattern=north east lines]
59 (#1) rectangle (#2);}
60 }{
61 \psframe[linestyle=none,fillstyle=vlines,hatchwidth=0.2pt,
62 hatchsep=3pt](#1)(#2)
63 }
64 }

```

### 3.4 Longueurs et compteurs internes

```

65 \newcounter{ligne} % numéro de ligne
66 \newcounter{noeud} % numéro du nœud
67 \newcounter{numvr} % numéro de la valeur remarquable
68 \newcounter{numdiscont} % numéro de la discontinuité

```

Un compteur est créé pour chaque discontinuité : `discont1`, `discont2`, etc. Le compteur `discont⟨i⟩` contient le numéro du nœud précédant la *i*-ème discontinuité. La flèche partant de ce nœud ne sera pas tracée. Il faut un compteur de plus que le nombre de discontinuités.

```

69 \AtBeginDocument{% car maxdiscont a pu être modifié dans le préambule
70 \stepcounter{maxdiscont}
71 % il faut un compteur de plus que le nb de discontinuités
72 \multido{\I=1+1}{\themaxdiscont}{\newcounter{discont\I}}
73 }

```

Longueurs et compteurs pour la commande `\ZIh`.

```

74 \newlength{\ZIheight}

```

```

75 \newlength{\ZIdepth}
76 \newlength{\ZIwidth}
77 \newcounter{ZI} % numéro de la ZI
78 \newcounter{ZIstar} % numéro de ZI pour la commande \ZI*
79 \newcounter{ZIvarlignes} % dernière ligne des variations

```

### 3.5 Les environnements `tablvar` et `tablvar*`

Grâce à l'extension `array` nous pouvons définir un nouveau type de colonnes pour les intervalles : `i`.

```

80 \newcolumntype{i}[1]{>\centering\arraybackslash $}p{#1}<{ $}
81 % nouveau type de colonne i pour les intervalles

```

`\tablvarinit` Cette commande d'initialisation des compteurs est appelée au début de chaque environnement `tablvar` ou `tablvar*`. Les redéfinitions de `\extrarowheight` et `\arraystretch` seront locales dans l'environnement `tablvar`.

```

82 \newcommand{\tablvarinit}{
83 \setlength{\extrarowheight}{0pt} % paramètre de l'extension array
84 \renewcommand{\arraystretch}{\tablvarstretch}
85 \setcounter{ligne}{0}
86 \setcounter{numvr}{0}
87 }

```

`\ZIinit` Initialisation des longueurs et compteurs pour la commande `\ZIh`. Prend un argument (optionnel) qui est la largeur des colonnes intervalles (car il faut la sauvegarder dans `\ZIwidth`) et qui est passé automatiquement par `tablvar`.

```

88 \newcommand*{\ZIinit}[1][\intervalwidth]{
89 \setlength{\ZIheight}{0pt}
90 \setlength{\ZIdepth}{0pt}
91 \setlength{\ZIwidth}{#1}
92 \setcounter{ZI}{0}
93 \setcounter{ZIstar}{0}
94 \setcounter{ZIvarlignes}{3}
95 % doit être non nul pour \ZIh si tableau de signe seul
96 }

```

`tablvar` L'environnement `tablvar` : `\begin{tablvar}[(width)]{(nbintervals)}`. Le 1<sup>er</sup> paramètre (optionnel) permet de régler la largeur des colonnes intervalles (définie par `\intervalwidth` par défaut), le 2<sup>e</sup> paramètre (obligatoire) est le nombre d'intervalles. Cet environnement n'est rien d'autre qu'un `array` dans lequel on a réglé les options de colonages. Les colonnes de valeurs sont de type `c` et les colonnes d'intervalles de type `i`<sup>4</sup>.

```

97 \newenvironment{tablvar}[2][\intervalwidth]{
98 \tablvarinit
99 \ZIinit[#1]

```

---

4. L'utilisation d'un type de colonne permettant un ajustement automatique de la hauteur tel que fourni par l'extension `cellspace` de Josselin Noirel ne fonctionne pas ici : les double-barres ne sont pas correctement dessinées.

```

100 \begin{array}{l}
101 |c|@{\hspace{\bordercolsep}}%
102 *{#2}{c@{\hspace{\innercolsep}}}%
103 i{#1}@{\hspace{\innercolsep}}}%
104 }%
105 c@{\hspace{\bordercolsep}}|}%
106 }
107 }\end{array}}

```

**tblvar\*** L'environnement `tblvar*` est une variante de `tblvar` (même syntaxe) où la première et la dernière colonnes de valeurs sont alignées respectivement à gauche (l) et à droite (r). Ceci est utile lorsqu'il y a une double barre à l'une des extrémités.

```

108 \newcounter{nb@intervals}
109 \newenvironment{tblvar*}[2][\intervalwidth]{
110 % environnement tblvar* , type l et r pour les extrémités
111 \tblvarinit
112 \ZIinit[#1]
113 \setcounter{nb@intervals}{#2}
114 \addtocounter{nb@intervals}{-1}
115 \begin{array}{l}
116 |c|@{\hspace{\bordercolsep}}%
117 l@{\hspace{\innercolsep}}%
118 i{#1}@{\hspace{\innercolsep}}}%
119 *{\value{nb@intervals}}{
120 c@{\hspace{\innercolsep}}%
121 i{#1}@{\hspace{\innercolsep}}}%
122 }%
123 r@{\hspace{\bordercolsep}}|}%
124 }
125 }\end{array}}

```

### 3.6 La commande `\variations`

C'est ici qu'est la magie !

**\varloop** La commande `\varloop{<iter>}{<code>}` répète `<code>` (`<iter>`−1) fois (car la dernière ligne de variations doit subir un traitement particulier). Nous avons créé une variante de `\multido` (du package `multido`) car `\multido`, `\Multido` ou `\whiledo` plantent sur `\\` ou `\@arraycr` et la commande `\variations` a besoin d'utiliser une boucle dans un tableau. `\varloop` n'est autre qu'un `\ifthenelse` récursif.

```

126 \newcounter{loop@counter}
127 \newcommand{\varloop}[2]{%
128 \setcounter{loop@counter}{#1}
129 \addtocounter{loop@counter}{-1}% on boucle 1 fois de moins que #1
130 \ifthenelse{\value{loop@counter}=0}{}{%
131 #2 \varloop{\value{loop@counter}}{#2}%
132 }
133 }

```

`\variations` `\variations[ $\langle nblignes \rangle$ ]{ $\langle code \rangle$ }` où  $\langle nblignes \rangle$  est le nombre de lignes pour les variations (3 par défaut);  $\langle code \rangle$  contient les séparateurs de colonnes et la composition des variations grâce aux commandes de positionnement.

Le principe est que l'on parcourt 3 fois (par défaut) le contenu de `\variations`; à chaque itération, le compteur `ligne` est incrémenté, le compteur `noeud` est remis à 0 puis incrémenté à chaque commande `\pos`, mais le contenu de `\pos` n'est affiché et le nœud n'est effectivement créé que si la valeur du compteur `ligne` correspond à l'argument de ligne de `\pos`.

Les flèches et pointillés sont dessinés à la fin, quand tous les nœuds sont créés, mais il faut les tracer avant le `\final`, sinon la compilation plante!?

```

134 \newcommand*\variations[2][3]{% #1=nblignes (3 par défaut)
135 % (ré)initialisation des compteurs
136 \setcounter{ligne}{0} % nécessaire pour 2 parties variations
137 \setcounter{numdiscont}{0}
138 \multido{\I=1+1}{\themaxdiscont}{\setcounter{discont\I}{0}}
139 \setcounter{ZIvarlignes}{#1} % nécessaire pour \ZIh
140 % boucle : on exécute le code #2 un nb de fois égal à (#1)-1
141 \varloop{#1}{%
142 \setcounter{noeud}{0}\setcounter{numvr}{0}\setcounter{ZI}{0}
143 % à chaque boucle on réinitialise certains compteurs
144 \stepcounter{ligne} % le numéro de ligne est incrémenté
145 #2 % les nœuds sont fabriqués par le code #2 (avec \pos et \vr)
146 \\ % retour ligne
147 }
148 % dernière itération -> flèches tracées AVANT \\ sinon bug !?
149 \setcounter{noeud}{0}\setcounter{numvr}{0}\setcounter{ZI}{0}
150 \stepcounter{ligne} #2
151 % tracé des flèches
152 \addtocounter{noeud}{-1} % 1 flèche de moins que le nb de nœuds
153 \setcounter{numdiscont}{1}
154 \multido{\Ix=1+1,\Iy=2+1}{\thenoeud}{
155 \ifthenelse{\value{discont\thenumdiscont}=\Ix}{
156 % on saute les discontinuités
157 \stepcounter{numdiscont}}{
158 % sinon on trace la flèche N1->N2 puis N2->N3, etc.
159 \fleche{N\Ix}{N\Iy}
160 }
161 }
162 % tracé des pointillés pour les valeurs remarquables
163 \multido{\Ix=1+1}{\thenumvr}{\vrconnect{X\Ix}{Y\Ix}}
164 \\ % dernier retour ligne du tableau
165 }

```

### 3.7 Les commandes de positionnement

`\pos` `\pos[ $\langle opt \rangle$ ]{ $\langle ligne \rangle$ }{ $\langle valeur \rangle$ }` sert à positionner les valeurs dans la partie variations.  $\langle ligne \rangle$  désigne la ligne où il faut placer  $\langle valeur \rangle$  et produire le nœud (numéroté avec le compteur `noeud` et défini en appelant la commande `\noeud`).

Les lignes de variations sont numérotées *du haut vers le bas* (et les lignes de signes portent toutes le numéro 0). Le 1<sup>er</sup> argument optionnel, *c* (centered, par défaut), *t* (top) ou *b* (bottom), est utilisé pour le positionnement des flèches.

`\pos*` Dans la version étoilée, `\pos*{<ligne>}{<valeur>}`, la différence est qu’aucun nœud n’est créé. Ceci est utile en particulier pour la colonne de base, contenant les légendes.

```
166 \newcommand*{\@pos}[3][c]{
167 \stepcounter{noeud}
168 \ifthenelse{\theligne=#2}{
169 \noeud[#1]{N\thenoeud}#{3}
170 }{} % si ligne != #2, on ne fait rien
171 }
172 \newcommand*{\@@pos}[2]{\ifthenelse{\theligne=#1}{#2}{} }
173 \newcommand*{\pos}{\ifstar{\@pos}{\@pos}}
```

Voici quelques alias utiles qui peuvent être utilisés à la place des commandes `\pos` et `\pos*`.

`\haut` `\haut{<valeur>} = \pos{1}{<valeur>}`.

```
174 \newcommand*{\haut}{\pos{1}}
```

`\bas` `\bas{<valeur>} = \pos{3}{<valeur>}`.

```
175 \newcommand*{\bas}{\pos{3}}
```

`\mil` `\mil{<valeur>} = \pos*{2}{<valeur>}`.

```
176 \newcommand*{\mil}{\pos*{2}}
```

Voici enfin deux macros qui permettent si besoin d’ajuster le positionnement (en plus des commandes L<sup>A</sup>T<sub>E</sub>X `\hspace` ou `\makebox`).

`\zbox` Place son contenu dans une boîte de largeur nulle : affiche le contenu mais considère que l’espace occupé est nul pour le calcul des largeurs de colonne. Sa syntaxe est : `\zbox[<pos>]{<contenu>}` où *<opt>* = *c* (par défaut), *l* (left) ou *r* (right).

```
177 \newcommand*{\zbox}[2][c]{\makebox[0pt][#1]{#2}}
```

`\vdecal` Décalage vertical : `\vdecal{<decal>}{<contenu>}`, le 1<sup>er</sup> paramètre est le décalage (positif = vers le haut ou négatif = vers le bas), le second est le contenu à placer.

```
178 \newcommand*{\vdecal}[2]{\smash{\raisebox{#1}{#2}}}
```

`\smash` a pour effet d’annuler la hauteur de la boîte afin de ne pas agrandir la ligne courante.

### 3.8 Barres, discontinuités et valeurs remarquables

`\bb` La macro `\bb`, qui produit une double barre, provient de l’extension `variations` de Christian Obrecht `\def\bb{\vrule\kern1pt\vrule}`. Nous avons ajouté 1pt d’espace avant et après, afin d’éviter le contact entre la double barre et les limites à gauche ou à droite.

```
179 \newcommand*{\bb}{\kern1pt\vrule\kern1pt\vrule\kern1pt}
```

`\barre` La macro `\barre` permet de tracer une barre verticale pour marquer les séparations de colonne dans un tableau de signe, en passant à travers les 0. Son aspect est contrôlé par les paramètres `\tvbarrewidth` et `\tvbarrecolor`. Sa syntaxe est : `\barre[⟨valeur⟩]` où, en général, on met 0 comme argument optionnel.

```
180 \newcommand*{\barre}[1][0]{\makebox[0pt]{\$#1\$}
181 \color{tvbarrecolor}
182 \vrule width \tvbarrewidth
183 }
```

`\discont` Associe un numéro de nœud à un compteur de discontinuité (chaque discontinuité a son propre compteur). La flèche entre le nœud précédent `\discont` (enregistré dans le compteur) et le nœud suivant ne sera pas tracée.

```
184 \newcommand*{\discont}{
185 \ifthenelse{\theligne=1}{
186 % on ne compte les discontinuités qu'une seule fois, sur ligne 1
187 \stepcounter{numdiscont}
188 \setcounter{discont\thenumdiscont}{\thenoeud}
189 }{}
190 }
```

`\bblim` Sert à positionner correctement des limites à gauche et à droite d'une double barre en conservant le centrage de la double barre. Cette macro trace la double barre, place les limites et appelle `\discont`. Le principe est que lorsqu'on place la plus large des deux valeurs d'un côté de la double barre, on place une boîte fantôme de même largeur de l'autre côté ; quant à la plus étroite des deux valeurs, elle est placée dans une `\zbox` qui annule sa largeur. Sa syntaxe est :

```
\bblim{⟨ligne gauche⟩}{⟨limite gauche⟩}{⟨ligne droite⟩}{⟨limite droite⟩}
191 \newsavebox{\@tvlbox}
192 \newsavebox{\@tvrbox}
193 \newcommand*{\bblim}[4]{
194 \sbox{\@tvlbox}{\$#2\$}
195 \sbox{\@tvrbox}{\$#4\$}
196 \ifdim \wd\@tvlbox > \wd\@tvrbox
197 \pos{#1}{#2} \bb\discont
198 \zbox[1]{\pos{#3}{#4}} \pos*{#1}{}
199 \else
200 \pos*{#3}{} \zbox[r]{\pos{#1}{#2}}
201 \bb\discont \pos{#3}{#4}
202 \fi
203 }
```

`\vr` La commande `\vr` fabrique un nœud pour chaque valeur remarquable. Les nœuds sont désignés par  $X_1, X_2, \dots$  sur la ligne des  $x$  et  $Y_1, Y_2, \dots$  sur les lignes de variations. Sa syntaxe est : `\vr[⟨ligne⟩]{⟨valeur⟩}`. Le paramètre optionnel *⟨ligne⟩* vaut 2 par défaut pour  $Y$ , et n'est pas pris en compte pour  $X$  (ligne 0), le second paramètre est la valeur à afficher. Les nœuds seront ensuite reliés par la commande `\vrconnect` (appelée par `\variations`) en fonction de leur numéro.

```
204 \newcommand*{\vr}[2][2]{% ligne 2 par défaut sauf sur ligne 0
```

```

205 \stepcounter{numvr}
206 \ifthenelse{\theligne=0}{
207 \noeud{X\thenumvr}{#2}
208 }{
209 \ifthenelse{\theligne=#1}{
210 \noeud{Y\thenumvr}{#2}
211 }{}
212 }
213 }

```

### 3.9 Zones interdites

**\ZIc** Commande sans argument qui produit une zone interdite colorée. À placer dans les intervalles voulus. La densité de gris ou la couleur peuvent être modifiées en redéfinissant la macro.

```
214 \newcommand{\ZIc}{\discont \cellcolor[gray]{0.7}}
```

**\ZIh** La macro `\ZIh` sert à produire une zone interdite hachurée. Le tracé des hachures n'est déclenché que sur la dernière ligne des variations, les appels à `\ZIh` sur les lignes précédentes ne font que cumuler hauteur (`\ht`) et profondeur (`\dp`) de la ligne. Dans une ligne de signe, il ne faut qu'*un seul appel* à `\ZIh`, qui peut être placé n'importe où. En effet, pour les lignes d'indice 0, la macro ne sait pas s'il y a eu un changement de ligne, donc elle cumule systématiquement la hauteur à chaque appel. Par contre, dans la partie variations, les `\ZIh` doivent être placés autant de fois que nécessaire, dans les intervalles voulus.

La macro possède un paramètre optionnel `\ZIh[hauteur]` qui est un supplément de hauteur global, permettant de faire un ajustement fin si des lignes ont été agrandies à cause de leur contenu (grandes fractions par exemple).

Limitation : fonctionne moins bien avec *TikZ* (sensible à la taille de police).

**\ZIh\*** La macro `ZIh*` (appelée par `\ZIh` sur la dernière ligne des variations) déclenche le tracé d'un rectangle de hachures en appelant la commande `\hachure`. Elle utilise les dimensions précédemment enregistrées dans les variables de dimension (par `\ZIh`). Dans un tableau de signes sans partie variations elle doit être placée sur la dernière ligne à la place de `\ZIh` (pour les lignes précédentes). Elle possède le même argument optionnel [*hauteur*].

```

215 \newcommand*{\@ZI}[1][Opt]{%
216 \discont
217 \ifthenelse{\theligne=\value{ZIVarlignes}}{\@ZI[#1]}{%
218 \ifthenelse{\theligne=0 \or \theZI=0}{%
219 % on ne doit cumuler la hauteur qu'une fois par ligne
220 % => une seule occurrence de la macro sur une ligne 0
221 \global\advance\ZIheight by \ht\@arstrutbox
222 \global\advance\ZIheight by \dp\@arstrutbox
223 }{}
224 }
225 \stepcounter{ZI} % RAZ par \variations à chaque ligne
226 }

```

```

227
228 \newcommand*{\@ZI}[1][Opt]{% \discont inutile ici
229 \ifthenelse{\theZIstar=0}{% cumul des longueurs 1 seule fois
230 \global\advance\ZIheight by \ht\@arstrutbox
231 \global\advance\ZIheight by #1
232 \global\advance\ZIdepth by \dp\@arstrutbox
233 \global\advance\ZIwidth by \innercolsep
234 \global\advance\ZIwidth by \innercolsep
235 \global\advance\ZIwidth by 2pt % espace autour doubles barres
236 }{}
237 \hachure{-0.5\ZIwidth,-\ZIdepth}{0.5\ZIwidth,\ZIheight}
238 \stepcounter{ZIstar}
239 }
240
241 \newcommand*{\ZIh}{\@ifstar{\@ZI}{\@ZI}}

```