# L'extension tablvar\*

# Antoine Missier prenom.nom@ac-toulouse.fr

## 11 novembre 2018

# Table des matières

1	Intr	roduction	1
2	Util	lisation	<b>2</b>
	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	8
	2.5	Zones interdites	12
3	Le	code	17
	3.1	Extensions requises et options	17
	3.2	Les paramètres généraux	17
	3.3	Les commandes graphiques pstricks/tikz	18
	3.4	Longueurs et compteurs internes	19
	3.5	Les environnements tablvar et tablvar*	20
	3.6	La commande \variations	21
	3.7	Les commandes de positionnement	23
	3.8	Barres, discontinuités et valeurs remarquables	24
	3.9	Zones interdites	25

# 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 tâche, nous avons voulu produire des tableaux de manière automatisée, y compris pour des tableaux complexes, mais avec de nombreuses possibilités de réglages et d'ajustements personnels et une haute qualité graphique, en particulier pour le dessin des flèches, en exploitant les fonctionnalités des extensions

<sup>\*</sup>Ce document correspond à tablvar v1.0, dernière modification le 11/11/2018.

graphiques pstricks (plus exactement pst-node) ou tikz pour définir un graphe de næuds 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 LATEX 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.

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

L'extension possède deux options utilisées pour le tracé automatique des flèches : pstricks (par défaut) ou tikz (\usepackage[tikz]{tablvar}). Pour l'option pstricks, il faut compiler avec latex + dvips + ps2pdf; pour l'option tikz, il faut compiler DEUX fois avec pdflatex (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, entre les deux, supprimer le fichier .aux, sinon la compilation plante.

## 2 Utilisation

# 2.1 Tableaux de variations simples

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

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

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

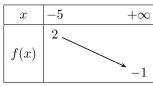
tablvar

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).

 $tablvar[\langle largeur \rangle]$ 

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.

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



\pos

Les commandes \haut et \bas ne sont en fait que des alias d'une commande plus générale qui est  $\{opt\} \{\langle ligne \rangle\} \{\langle valeur \rangle\}$ . Celle-ci place la  $\langle valeur \rangle$  sur la  $\langle ligne \rangle$  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  $\langle opt \rangle$  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

\pos\*

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

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

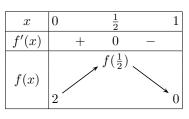
```
\begin{tablvar}{2}
    \hline
    x & 0 & & \frac12 & & 1 \\
    \hline
    f'(x) & & + & 0 & - & \\
    \hline
    \variations{\mil{f(x)} & \bas{2} && \haut{f(\frac12)} && \bas{0}}
}
    \hline
```

\end{tablvar}

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

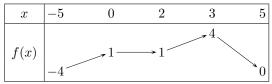
Le même tableau avec les commandes \pos et \pos\* et l'option t pour le max.

<sup>1.</sup> Ce paramètre optionnel n'est pas implémenté et est sans effet pour l'option tikz.



\fleche

On peut modifier l'aspect des flèches en redéfinissant la macro \fleche grâce aux nombreuses options offertes par pstricks ou tikz. Cette redéfinition sera locale si on la place dans l'environnement math du tableau. Ci-dessous des flèches plus fines, plus proches des nœuds et dont la pointe est plus effilée (codé ici pour pstricks uniquement).



Pour placer des nœuds de flèche au milieu, on doit utiliser  $pos{2}$  et non mil.

## 2.2 Tableaux de signes avec barres de séparation

\barre

La macro \barre, présentée dans l'exemple ci-dessous, sert de séparateur de colonne dans les tableaux de signe. Elle est bien sûr facultative de même que la

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		69 4	\	· -∞

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

### 2.3 Doubles barres et discontinuités

Une double barre s'obtient avec la commande \bb. Pour ne pas tracer de flèche discont entre deux nœuds, on place la commande \discont entre les deux.

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

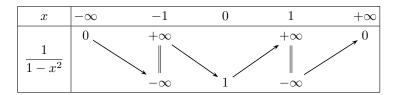
La commande \dfrac provient de l'extension amsmath (ou sinon on l'obtient avec \newcommand{\dfrac}{\displaystyle\frac}).

\begin{tablvar}{2}

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

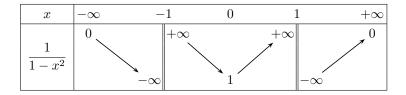
La syntaxe ci-dessus pose problème lorsque les limites à gauche et à droite d'une valeur interdite se trouvent sur des lignes différentes, ou que leur contenu n'a pas la même largeur, car alors la double barre ne serait plus au centre de sa colonne et se trouverait décalée d'une ligne à l'autre.

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).



```
\begin{tablvar}[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{tablvar}
```

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  $\langle ligne \rangle$  et  $\langle limite \rangle$  à gauche puis à droite de la double barre. Elle trace la double barre, place les limites et gère la discontinuité.



```
\begin{tablvar}[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{tablvar}
```

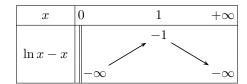
Un autre exemple avec des limites de largeur assez différentes (placées ici sur la même ligne).

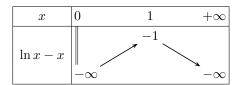
x	0	$\frac{\pi}{6}$		$\frac{\pi}{2}$ $\pi$
f'(x)	+	0	_	_
f(x)	f(0)	$f(\frac{\pi}{6})$	$-\infty$	$0 \qquad f(\pi)$

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

tablvar\*

L'environnement tablvar\* sert à gérer correctement le positionnement des doubles barres lorsqu'elles se trouvent aux extrémités. La différence avec tablvar 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.



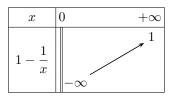


On observera l'utilisation de **\pos\*** dans le second tableau, pour tracer une double barre, sauf sur la ligne 3 où on place  $-\infty$ .

```
\begin{tablvar*}{2}
   \hline
   x & 0 && 1 && +\infty \\
   \hline
   \label{linear} $$ \left( x - x \right) & \b \pos{3}{-\left( x - x \right) & \c x - x \right) $$
       \pos{3}{-\infty} 
   \hline
\end{tablvar*}
\begin{tablvar*}{2}
   \hline
   x & 0 && 1 && +\infty \\
   \hline
   \pos{3}{-\infty} && \pos{1}{-1} && \pos{3}{-\infty} }
   \hline
\end{tablvar*}
```

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. Cidessous, de l'espace a été ajouté à droite du 1.

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



\tablvarinit

Enfin, pour personnaliser totalement la définition des colonnes, on peut revenir à l'environnement array plutôt que d'utiliser tablvar (il faut dans ce cas appeler \tablvarinit juste avant \begin{array}). Ce qui est essentiel dans cette extension est la macro \variations avec les commandes de positionnement et non l'environnement tablvar qui n'est qu'un array dans lequel on a fixé les définitions de colonnes (et initialisé des variables).

### 2.4 Valeurs remarquables

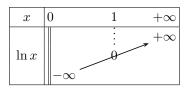
Nous appellerons  $valeur\ remarquable$ , un valeur supplémentaire que l'on place dans un tableau de variation et qui ne correspond pas à un extremum.

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

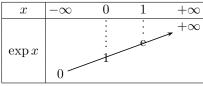
 $\v$ sur la ligne des x et une commande  $\v$ dans la partie variations. Les valeurs remarquables seront automatiquement reliées par des pointillés (tracés réalisés dans la commande  $\v$ ariations).

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

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



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



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

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).

 $\operatorname{Vr}[\langle ligne \rangle]$ 

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).

\tablvarstretch

À cause des 4 lignes de variations, la hauteur de lignes du tableau a été réduite en modifiant le facteur \tablvarstretch (fixé à 1.2 par défaut), modification locale car placée à l'intérieur de l'environnement math.

\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  $\vdecal$  dont la syntaxe est  $\vdecal\{\langle decal\}\}\{\langle contenu\}\}$  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	$\alpha_1$		$\frac{\pi}{6}$		$\alpha_2$		$\frac{\pi}{2}$
f'(x)	+	:	+	0	_		_	
f(x)	f(0)		<i></i>	$f(\frac{\pi}{6})$		• 0 _	<u></u>	$-\infty$

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{tablvar*}{4}
    \hline
    x & 0 && \vr{\alpha_1} && \frac{\pi}{6} && \alpha_2 && \frac{\pi}{2}\\
    \hline
    f'(x) && + && + & 0 & - && - & \bb \\
    \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}\bb }
    \hline
\end{tablvar*}
```

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

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

```
\begin{tablvar}{2}
   \hline
                                                       0
   x & -\pi && \vr{0} && \pi \\
                                                      1:1
   \hline
   f(x)
      \haut{ 1 \hspace{0.2em} } \discont
                                                       0
                                               0
      \vr[3]{0} \haut{ \hspace{0.2em} 1 }
      && \bas{0}
   }
   \hline
\end{tablvar}
```

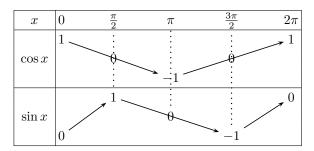
Voici un dernier exemple, traité de deux manières différentes, avec des valeurs

remarquables dans deux tableaux conjoints. Avec des nœuds c'est assez simple.

x	0	$\frac{\pi}{2}$	$\pi$	$\frac{3\pi}{2}$	$2\pi$
$\cos x$	1	• 0	-1	<b>,</b> 0	<b>,</b> 1
$\sin x$	0	1	<b>1</b> 0		0

```
\begin{tablvar}{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
```

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

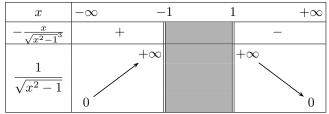


```
\vrconnect{X1}{Y1}
\vrconnect{X2}{Y2}
\vrconnect{X3}{Y3}
```

# 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.

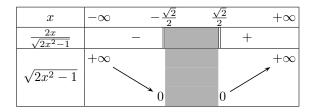


Plusieurs « interventions » manuelles sont nécessaires. D'abord sur la ligne des x, il faut permettre aux bornes de déborder sur la colonne de la zone interdite. Pour cela on les place dans une  $\zbox$ . La commande  $\zbox[\langle pos\rangle]\{\langle contenu\rangle\}$  affiche son contenu mais considère la largeur comme nulle. Le 1er paramètre (optionnel) est le positionnement dans la boite : c (par défaut), 1 (left) ou r (right). Ici zbox est aussi nécessaire pour annuler la largeur des boites 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.

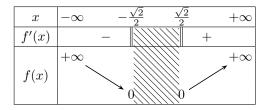
Enfin, les signes ont été décalés avec \hspace, pour améliorer leur centrage par rapport aux flèches.

```
\begin{tablvar}[4em]{3}
    \hline
    x & -\infty && \zbox{-1} && \zbox{1} && +\infty \\
    \hline
    -\frac{x}{\sqrt{x^2-1}^3} && \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[1]{\haut{+\infty}} && \bas{0}
}
    \hline
\end{tablvar}
```

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



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



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

```
\begin{tablvar}{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{tablvar}
```

 $\verb|\ZIh[|\langle hauteur|\rangle]|$ 

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  $\Time \Time \T$ 

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

Un exemple avec deux zones interdites, la fonction  $f: x \longmapsto \sqrt{(x^2-1)(x^2-4)}$ .

x	$-\infty$	-2	-1		0		1	2		$+\infty$
2x	1			_	0	+			+	
$2x^2 - 5$	+			_		_			+	
f'(x)	I			+	0	_			+	
f(x)	+8	•			, 2 <	\			<i></i>	+∞

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.

```
\hline
2x^2-5 & &+& \barre & \ZIh & \barre &-& \barre &&
    \barre &+& \\
    \hline
    f'(x) & &-& \bb & \ZIh & \bb &+& \barre[0] &-& \bb && \bb &+& \\
    \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	_			+	

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/////	0/	+		+	
x-2		_	_		_	0	+	
$(x-2)\sqrt{x^2-1}$		_	$\theta$	<b>(0</b>	_	0	+	

\begin{tablvar}{4}

Ci-dessus, la commande LATEX  $\phi$ 0 sur la ligne des x permet de créer une boite 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 1	$\sqrt{\frac{5}{2}}$	2	$+\infty$
2x	_	_	_	_	0	+	+	+	+	
$2x^2 - 5$	+	+ (	) –	_		_	_	0 +	+	
f'(x)	_			+	0	_			+	
f(x)	$+\infty$			)	* <sup>2</sup> \	\			)	$+\infty$

Si l'on ne dispose pas de ligne de signe pour placer la commande  $\$ nachure, 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.

```
\bas{0} && \haut{2} && \bas{0} \discont &&
      \pos*{3}{\hat{-}4em,12.5ex}{4em,-1.1ex}  \& \bas{0}  \& \pos*{3}{\hat{-}4em,12.5ex}  
      \haut{+\infty}
   }
   \hline
\end{tablvar}
```

#### 3 Le code

#### 3.1Extensions requises et options

```
1 \RequirePackage{array}
2 \RequirePackage{ifthen}
3 \RequirePackage{multido}
4 \RequirePackage{colortbl}
6 \newboolean{tikz}
7 \DeclareOption{tikz}{\setboolean{tikz}{true}}
9 \ProcessOptions \relax
10
11 \ifthenelse{\boolean{tikz}}{
     \RequirePackage{tikz}
     \usetikzlibrary{patterns}
13
     \newlength{\tikznode@below}
14
15
     \RequirePackage{pst-node}
16
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  $\alpha$  variance 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 maxdiscont 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'espacement vertical des lignes du tableau. La valeur par défaut est 1.2 (1 correspondant à la valeur standard d'un environnement array). Placée dans un environnement math, elle sera locale à cet environnement.

```
26 \newcommand{\tablvarstretch}{1.2}
```

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

```
27 \neq 27 
28 \setlength{\tvbarrewidth}{0.5pt}
```

tybarrecolor

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).

La commande  $fleche{\langle noeud1 \rangle} {\langle noeud2 \rangle}$  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]{
      \ifthenelse{\boolean{tikz}}{
31
           \tikz[remember picture,overlay]{\draw[->,>=stealth,
32
               line width=0.6pt] (#1) -- (#2);}
33
      }{
34
           \ncline[arrowsize=2pt 2,arrowinset=0.4,nodesep=3pt,
35
               linewidth=0.6pt]\{->\}{#1}{#2}
36
      }
37
38 }
```

\vrconnect

La commande  $\vconnect{\langle noeud1\rangle}{\langle noeud2\rangle}$  relie les nœuds définis par  $\vconnect{\langle noeud2\rangle}{\langle noeud2\rangle}{\langle noeud2\rangle}{\langle noeud2\rangle}{\langle noeud2\rangle}{\langle noeud2\rangle}$  relie les nœuds définis par  $\vconnect{\langle noeud2\rangle}{\langle n$ 

```
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

```
47 \newcommand*{\noeud}[3][c]{
      \ifthenelse{\boolean{tikz}}{
48
          \tikz[remember picture,baseline]{
49
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
          \rnode[#1]{#2}{#3}
53
      }
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]{
      \ifthenelse{\boolean{tikz}}{
57
          \tikz[remember picture,overlay]{\fill[pattern=north east lines]
58
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  $\operatorname{discont}\langle i\rangle$  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\*.

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

tablvar L'environnement tablvar : \begin{tablvar}[ $\langle width \rangle$ ] { $\langle nbintervals \rangle$ }. Le 1er paramètre (optionnel) permet de régler la largeur des colonnes intervalles (définie par

\intervalwidth par défaut), le 2ème 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.

```
97 \newenvironment{tablvar}[2][\intervalwidth]{
       \tablvarinit
 98
        \ZIinit[#1]
 99
        \begin{array}{%
100
            |c|@{\hspace{\bordercolsep}}%
101
            *{#2}{c@{\hspace{\innercolsep}}%
102
                i{#1}@{\hspace{\innercolsep}}%
103
104
105
           c@{\hspace{\bordercolsep}}|%
       }
106
107 }{\end{array}}
```

tablvar\* L'environnement tablvar\* est une variante de tablvar (même syntaxe) où la première et la dernière colonnes de valeurs sont alignées respectivement à gauche (1) 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{tablvar*}[2][\intervalwidth]{
       \% environnement tablvar* , type 1 et r pour les extrémités
110
111
       \tablvarinit
       \ZIinit[#1]
112
       \setcounter{nb@intervals}{#2}
113
114
       \addtocounter{nb@intervals}{-1}
       \begin{array}{%
115
116
           |c|@{\hspace{\bordercolsep}}%
117
           10{\hspace{\innercolsep}}%
118
           i{#1}@{\hspace{\innercolsep}}%
           *{\value{nb@intervals}}{
119
                c@{\hspace{\innercolsep}}%
120
121
                i{#1}@{\hspace{\innercolsep}}%
           }%
122
           r@{\hspace{\bordercolsep}}|%
123
       }
124
125 }{\end{array}}
```

### 3.6 La commande \variations

C'est ici qu'est la magie!

\tvloop La commande \tvloop{\langle iter\}}{\langle code\} (\langle iter\-1) fois (car la dernière ligne de variations doit subir un traitement particulier). C'est 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. \tvloop n'est autre qu'un \ifthenelse récursif.

126 \newcounter{loop@counter}

\variations

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

## 3.7 Les commandes de positionnement

\pos\* Dans la version étoilée, \pos\*{\langle ligne\rangle} \{\langle valeur\rangle}\}, 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\*.

```
\label{eq:local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_
```

Voici enfin deux macros qui permettent si besoin d'ajuster le positionnement (en plus des commandes LATFX \hspace ou \makebox).

Vzbox Place son contenu dans une boite de largeur nulle : affiche le contenu mais considère que l'espace occupé est nul pour le calcul des largeurs de colonne. Sa syntaxe est :  $\zbox[\langle pos \rangle] \{\langle contenu \rangle\}$  où  $\langle opt \rangle = c$  (par défaut), 1 (left) ou r (right).

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

\smash a pour effet d'annuler la hauteur de la boite 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 1 pt 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[\langle valeur \rangle] où, en général, on met 0 comme argument optionnel.

```
180 \newcommand*{\barre}[1][]{\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.

\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 boite 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 :

 $\label{limite gauche} $$ \left( limite \ gauche \right) = \left( limite \ droite \right) = \left($ 

```
191 \newsavebox{\@tvlbox}
192 \newsavebox{\@tvrbox}
193 \newcommand{\bblim}[4]{
        \sbox{\ensuremath{\sbox}{\$#2\$}}
194
        \strut_{\strut}\strut_{\strut}{$\#4\$}
195
        \ifdim \wd\@tvlbox > \wd\@tvrbox
196
             \pos{#1}{#2} \b\discont
197
             \zbox[1]{\pos{#3}{#4}} \pos*{#1}{\phantom{#2}}
198
199
             \pos*{#3}{\phantom{#4}} \zbox[r]{\pos{#1}{#2}}
200
201
             \b \discont \pos{#3}{#4}
202
        \fi
203 }
```

\vr La commande \vr fabrique un nœud pour chaque valeur remarquable. Les nœuds sont désignés par X1, X2, ... sur la ligne des x et Y1, Y2, ... sur les lignes de variations. Sa syntaxe est : \vr [\langle ligne \rangle \langle valeur \rangle \rangle. Le paramètre optionnel \langle ligne \rangle 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}{
            \noeud{X\thenumvr}{#2}
207
208
            \ifthenelse{\theligne=#1}{
209
                \noeud{Y\thenumvr}{#2}
210
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}}
```

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 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).

```
221
             \global\advance\ZIheight by \ht\@arstrutbox
             \global\advance\ZIheight by \dp\@arstrutbox
222
223
         }{}
      }
224
      \stepcounter{ZI} % RAZ par \variations à chaque ligne
225
226 }
227
228 \newcommand*{\00ZI}[1][0pt]{% \discont inutile ici
      229
         \global\advance\ZIheight by \ht\@arstrutbox
230
         \global\advance\ZIheight by #1
231
         \global\advance\ZIdepth by \dp\@arstrutbox
232
         \global\advance\ZIwidth by \innercolsep
233
234
         \global\advance\ZIwidth by \innercolsep
         \global\advance\ZIwidth by 2pt % espace autour doubles barres
235
      }{}
236
      237
238
      \stepcounter{ZIstar}
239 }
240
241 \newcommand{\ZIh}{\@ifstar{\@0ZI}{\0ZI}}
```