

L'extension  
**autoaligne**

v1.1  
6 octobre 2016

Christian TELLECHEA \*

Cette petite extension permet d'aligner les uns au-dessous des autres les termes et les signes d'égalité dans des expressions mathématiques s'étendant sur plusieurs lignes.

---

\*. [unbonpetit@openmailbox.org](mailto:unbonpetit@openmailbox.org)

# 1 Avant propos

C'est à force de voir, aussi bien sur `tex.stackexchange` que dans ma boîte email, des questions concernant l'extension `systeme` que j'ai décidé d'écrire une extension à la fois plus générale et plus simple que ne l'est `systeme`. En effet trop de ces questions ont pour but de faire réaliser à `systeme` des choses pour lesquelles elle n'est pas conçue!

À l'aide de la récente extension `listofitems`, coder `autoaligne` deviendrait (presque) une formalité tant les choses deviennent plus simples.

Ayant constaté que beaucoup de non francophones ne semblent en rien gênés que le manuel de `systeme` soit rédigé uniquement en français, détestant pour ma part traduire le français vers l'anglais ce qui s'apparente à un calvaire et enfin, voulant à tout prix défendre l'existence du français face au globbish dans la sphère `TEX/LATEX` et ailleurs, j'ai voulu que cette extension soit 100% en français et qu'elle le reste! Même dans le code, les noms des macros sont en français ce qui, une fois l'habitude prise, est fort agréable.

Cette extension requiert l'extension `listofitems` dans sa version 1.1 ou supérieur, doit être utilisée avec un moteur  $\epsilon$ -`TEX`, et doit être appelée sous `(pdf)(Xe)(lua)LATEX` par

```
\usepackage{autoaligne}
```

et sous `(pdf)(Xe)(lua)TEX` par

```
\input autoaligne.tex
```

## 2 Mode d'emploi

**La macro `\autoaligne`** Cette macro attend un argument obligatoire entre accolades contenant les lignes qui sont des expressions où l'alignement automatique doit entrer en vigueur.

Chaque ligne est séparée de sa voisine par « `\` ». Dans chaque ligne, on peut trouver autant de membres que l'on souhaite, séparés les uns des autres par « `=` ». Dans ces membres peuvent se trouver autant de termes que nécessaire, chacun séparé de son voisin par « `+` » ou « `-` ».

La macro `\autoaligne` va bâtir un alignement de type `\halign` dans lequel chaque terme et chaque signe est contenu dans une colonne. Toutes les colonnes sont composées en mode mathématique. Visuellement, le résultat se traduit par des termes et des signes alignés les uns au dessous des autres comme on peut le constater dans cet exemple simple :

<pre>\autoaligne{x-3y+z=2-i\5x+y-6z=1+5i\x-y-z=-3+i}</pre>	$\begin{array}{r} x - 3y + z = 2 - i \\ 5x + y - 6z = 1 + 5i \\ x - y - z = -3 + i \end{array}$
--	---

Lorsqu'un membre commence par « `-` », le signe fait partie prenante du terme et n'est pas compris comme un séparateur entre termes. On peut le constater dans le second membre de la dernière ligne.

**Termes vides** Les signes « `+` » doivent être utilisés pour insérer des termes vides. Il est déconseillé de se servir du signe « `-` ».

<pre>\autoaligne{x=a+c\+y=b-c\+z=-4c}</pre>	$\begin{array}{r} x = a + c \\ y = b - c \\ z = -4c \end{array}$
---	--

On peut noter deux choses :

1. les signes « `+` » devant les termes vides ou devant le premier terme non vide ne sont pas affichés;

2. lorsqu'une colonne ne contient que des signes non affichés comme c'est le cas dans le membre de gauche, l'encombrement horizontal de ces signes est pris en compte.

Lorsqu'un terme négatif doit apparaître en deuxième position après un terme vide, il *faut* meubler ce terme vide avec un contenu non affichable comme « {} », « \relax » ou « \null ». C'est, je pense, la seule situation où autoaligne conserve une ambiguïté qui demande, pour être levée, un petit effort de la part de l'utilisateur :

<code>\autoaligne{a-b={}-2i\ a+2b=1+i}</code>	$\begin{array}{r} a - b = -2i \\ a + 2b = 1 + i \end{array}$
---	--

**Composition des termes** Par défaut, chaque terme est composé au fer à droite comme on le constate sur cet exemple :

<code>\autoaligne{a+\sqrt{2}b=0\ a+b=10\sqrt{2}}</code>	$\begin{array}{r} a + \sqrt{2}b = 0 \\ a + b = 10\sqrt{2} \end{array}$
---	--

L'argument optionnel de `\autoaligne` permet de changer ce paramètre de composition. Si l'on souhaite spécifier ce paramètre globalement pour *tous* les membres, il faut écrire *\*(lettre)* où la *(lettre)* est « c » pour centré, « d » pour droite et « g » pour gauche. Toute autre lettre provoquera un message d'erreur et « d » sera alors prise par défaut.

Voici l'exemple ci-dessus repris avec un paramètre optionnel global « \*g » :

<code>\autoaligne[*g]{a+\sqrt{2}b=0\ a+b=10\sqrt{2}}</code>	$\begin{array}{r} a + \sqrt{2}b = 0 \\ a + b = 10\sqrt{2} \end{array}$
---	--

Pour spécifier le paramètre pour chaque membre, il faut écrire dans l'argument optionnel autant de *(lettres)* qu'il y a de membres dans l'alignement. Si l'on écrit plus de lettres qu'il y a de membres, les lettres en trop seront ignorées. S'il manque des lettres, un message d'erreur sera émis et la lettre « d » sera prise par défaut.

Voici l'exemple avec, semble-t-il, le meilleur compromis ici, c'est-à-dire « d » pour le 1<sup>er</sup> membre et « g » pour le second :

<code>\autoaligne[dg]{a+\sqrt{2}b=0\ a+b=10\sqrt{2}}</code>	$\begin{array}{r} a + \sqrt{2}b = 0 \\ a + b = 10\sqrt{2} \end{array}$
---	--

Pour contrarier l'alignement pour un seul terme dans une seule ligne, il est toujours possible de placer convenablement un (ou plusieurs) `\hfill` dans ce terme.

**Échappement de + et -** Pour écrire un signe « + » ou « - » sans qu'il ne soit interprété comme un délimiteur de terme, il faut employer les macros `\+` ou `\-` :

<code>\autoaligne{(1\+a)x+a^2y=a\ -2ax+(1\+a)y=2-a}</code>	$\begin{array}{r} (1 + a)x + a^2y = a \\ -2ax + (1 + a)y = 2 - a \end{array}$
--	---

**Combien de membres?** Autant que l'on veut !

<code>\autoaligne{a+b=1+2+3+4==+z=\alpha\ +x==+10==3\ =1+i+j-z}</code>	$\begin{array}{r} a + b = 1 + 2 + 3 + 4 = z = \alpha \\ x = 10 = 3 \\ = 1 = i + j - z \end{array}$
--	--

Si un membre est vide (comme celui précédant le 3), le signe d'égalité qui le précède n'est pas affiché par défaut. Pour changer ce comportement, il faut écrire `\egaldevantmembrevide=<nombre>`. Si le *(nombre)* (entier sur 8 bits positif ou nul) vaut autre chose que 0, le signe = sera affiché devant un membre vide.

**Augmenter l'espace vertical** Le coefficient défini par `\aavcoeff{⟨coeff⟩}` permet de modifier l'espace vertical entre les lignes en multipliant la valeur de `\baselineskip` par le `⟨coeff⟩`.

Normal <code>\autoalign{a+2b=1\\a-b=2} \qqquad</code>	Normal $a + 2b = 1$	étendu $a + 2b = 1$
<code>\aavcoeff{1.75}étendu \autoalign{a+2b=1\\a-b=2}</code>	$a - b = 2$	$a - b = 2$

**Position verticale de l'alignement** La macro `\autoalign` admet un autre argument entre parenthèses qui permet de spécifier quelle sera la position verticale de l'alignement créé par rapport à la ligne de base. Cet argument est une `⟨lettre⟩` qui vaut « c », « h » ou « b » pour que la ligne de base de l'alignement soit au centre de celui-ci, sur la ligne du haut ou sur celle du bas. La position centrée est celle par défaut. Lorsque les deux arguments optionnels sont présents (celui entre crochets et celui entre parenthèses), l'ordre des arguments optionnels *est sans importance*, [`⟨consigne⟩`](`⟨lettre⟩`) est accepté tout comme (`⟨lettre⟩`)[`⟨consigne⟩`].

Alignement c : <code>\autoalign(c){4=1+1+1+1\\=1+1+2\\=2+2\\=3+1}\medbreak</code>	Alignement c : $4 = 1 + 1 + 1 + 1$ $= 1 + 1 + 2$ $= 2 + 2$ $= 3 + 1$
Alignement h : <code>\autoalign(h){4=1+1+1+1\\=1+1+2\\=2+2\\=3+1}\medbreak</code>	Alignement h : $4 = 1 + 1 + 1 + 1$ $= 1 + 1 + 2$ $= 2 + 2$ $= 3 + 1$
Alignement b : <code>\autoalign(b){4=1+1+1+1\\=1+1+2\\=2+2\\=3+1}</code>	Alignement b : $4 = 1 + 1 + 1 + 1$ $= 1 + 1 + 2$ $= 2 + 2$ $= 3 + 1$