

Cahiers **GUT**enberg

☞ LE COIN DES GOUROUS

☞ Georges WEIL, Raymond SÉROUL

Cahiers GUTenberg, n° 6 (1990), p. 53-59.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1990__6_53_0>

© Association GUTenberg, 1990, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique
est constitutive d'une infraction pénale. Toute copie ou impression
de ce fichier doit contenir la présente mention de copyright.

Le coin des gourous

Responsable de la rubrique : Philippe LOUARN

INRIA-IRISA, Campus de Beaulieu, 35042 Rennes Cedex. 99 36 20 00 Email : gut@irisa.fr

Nous vous présentons aujourd'hui deux macro-instructions écrites toutes deux à Strasbourg, permettant de résoudre des problèmes qui se posent assez fréquemment et que les outils T_EX *standards* ne permettent pas de régler aisément. La première permet la numérotation des lignes d'un texte, indépendamment des commandes et environnements appelés dans ce texte. La seconde autorise la description d'un tableau, non plus ligne par ligne, mais colonne par colonne. Il convient de noter que ces deux commandes, écrites au départ pour *plain* T_EX, fonctionnent très bien avec L^AT_EX.

Si vous aussi, vous avez écrit une macro T_EX d'intérêt général, et que vous voulez en faire profiter les autres, n'hésitez pas à contacter le responsable du *Coin des Gourous*.

1. Numérotation des lignes d'un texte

(Georges WEIL, CCS Strasbourg.)

But de la macro

Permet de numérotter les lignes d'un texte dans la marge de gauche.

Appel de la macro

```
\cptligne  
par  
\cptlignefin
```

Arguments

par texte dont les lignes doivent être numérotées.

Fonctionnement

La macro `\cptligne` numérote les lignes de tous les paragraphes qui suivent jusqu'à la rencontre d'une commande `\cptlignefin`.

La lecture du membre crée une variable de type `\cptligneskip` permettant de définir l'espacement horizontal entre la numérotation et la marge de gauche, deux variables de type compteur `\cptlignepas` et `\cptligneval` permettant de définir respectivement le pas de l'impression de la numérotation et le numéro de la ligne. La commande `\cptlignefin` doit impérativement être précédée et suivie par `\par` sans quoi T_EX imprimera un message d'erreur du type `Undefined control sequence`.

`\cptligneskip` est valorisé par défaut à 5pt, `\cptlignepas` à 5.

`\cptligneval` est valorisé à 1 lors de la lecture de la macro. Il est de la responsabilité de l'utilisateur de réinitialiser ce compteur lors d'un appel ultérieur de la macro. Ce compteur est réinitialisé à 1 lors d'un changement de page.

Pour utiliser une autre fonte pour la numérotation, on utilisera la commande `\cptlignefonte = nom-de-la-fonte`.

Cette macro fonctionne en travaillant paragraphe par paragraphe. De ce fait, il

est impossible de commencer un groupe dans un paragraphe et de le terminer dans un autre. Une solution possible est :

- terminer la macro par `\cptlignefin`,
 - 5 — commencer le groupe,
 - rappeler `\cptligne`,
 - mettre le texte,
 - mettre `\cptlignefin`,
 - fermer le groupe.
- 10 Il faut faire de même pour insérer un titre, un saut vertical ou changer la valeur de `\baselineskip` entre deux paragraphes.

Exemple

Le texte précédent a été obtenu par :

```
\cptligne
```

La macro `\cptligne num\`'erote les lignes de tous...

...

...entre deux paragraphes.

```
\par
```

```
\cptlignefin \par
```

Texte source de la macro

```
\lineskip=.5\baselineskip
\newdimen\cptligneskip
\cptligneskip=5pt
\newcount\cptlignepas
\cptlignepas=1
\newcount\cptligneval
\cptligneval=1
\let\cptlignefonte=\tenrm
%
\catcode'\@=11
\newbox\cpt@lignebox
\newbox\cpt@lignesplit
\newdimen\cpt@lignedim
\newdimen\cpt@ligneinc
\newcount\cpt@lignetr
```

```
\def\cpt@ligne#1\par{\par
  \setbox\cpt@lignebox=% (a)
  \vbox{#1\par}
  \cpt@lignedim=\pagegoal% (b)
  \advance \cpt@lignedim
  by -\pagetotal
  \advance \cpt@lignedim
  by -\baselineskip
```

```
\ifdim! \ht\cpt@lignebox>% (c)
```

```
\cpt@lignedim
```

```
\ifdim\cpt@lignedim>% (d)
```

```
\baselineskip
```

```
{\vbadness=10000
```

```
\setbox\cpt@lignesplit=%
```

```
\vsplit\cpt@lignebox
```

```
to \cpt@lignedim
```

```
\setbox\cpt@lignesplit=% (e)
```

```
\vbox{\unvbox\cpt@lignesplit}
```

```
\cpt@lignedim=%
```

```
f! \ht\cpt@lignesplit
```

```
\hbox to\hsize{\cpt@@ligne% (f)
```

```
\box\cpt@lignesplit}}%
```

```
\fi
```

```
\global\cptligneval=1
```

```
\eject
```

```
\fi
```

```
\cpt@lignedim=%
```

```
f! \ht\cpt@lignebox
```

```
\advance\cpt@lignedim
```

```
by f! \dp\cpt@lignebox
```

```
\hbox to\hsize{\cpt@@ligne
```

```
\box\cpt@lignebox}}%
```

```
\par
```

```
}
```

```
\def\cpt@@ligne{\vbox{\hbox% (g)
```

```
to Opt{\hss\vbox{%
```

```
\cpt@ligneinc=\cpt@lignedim%
```

```
\loop\cpt@ligneins
```

```
\advance\cpt@ligneinc
```

```
by -\baselineskip%
```

```
\global\advance\cptligneval by 1
```

```
\ifdim\cpt@ligneinc>Opt\repeat
```

```
} \hskip \cptligneskip}}
```

```
}
```

```
\def\cpt@ligneins{% (h)
```

```
\cpt@lignetr=\cptligneval
```

```
\divide\cpt@lignetr
```

```
by \cptlignepas
```

```
\multiply\cpt@lignetr
```

```
by \cptlignepas
```

```
\ifnum\cpt@lignetr=\cptligneval%
```

```
\hbox{\cptlignefonte
```

```
\the\cptligneval}
```

```
\else
```

```
\hbox{\ }
```

```
\fi
```

```
}
```

```
\long\def\cptligne#1\par{%
```

```
\ifx#1\cptlignefin
```

```
\let\next@=\relax
```

```
\else \cpt@ligne #1\par
```

```

\let\next@=\cptligne
\fi \next@
}

\catcode'\@=12

```

Explications sur la macro

Le principe est de mettre le paragraphe dans une boîte verticale de nom `\cpt@lignebox` (a), de créer une boîte ayant la même hauteur que `\cpt@lignebox` contenant les numéros de lignes (macro `\cpt@@ligne`; puis d'imprimer une ligne dilatée vers la droite contenant les deux boîtes (g).

Les lignes de (b) à (c) servent à calculer la place disponible en fin de page. La valeur est mise dans `\cpt@lignedim`. Si la boîte ne peut être contenue dans le bas de la page (d), on imprime la partie de la boîte remplissant le bas de la page (f). La commande `\vbadness=10000` empêche l'impression d'un message "underfull vbox".

La commande `\setbox\cpt@lignesplit = \vbox{\unvbox\cpt@lignesplit}` (e) permet de supprimer l'espace indésirable que la commande `\vsplit` peut insérer au bas de la nouvelle boîte.

La macro `\cpt@@ligne` crée une boîte ayant le même nombre de lignes que la boîte `\cpt@lignebox`. Pour chaque ligne on appelle la macro `\cpt@lignesins` (h) qui range soit le numéro de la ligne, si ce dernier est un multiple de `\cpt@lignespas`, soit un espace.

La macro `\cpt@ligne` isole chaque paragraphe pour le traiter. C'est une boucle qui s'arrête à la reconnaissance d'une commande `\cpt@lignesfin` en début de paragraphe.

2. La commande `\valign`

(Raymond SEROUL, Laboratoire de Typographie Informatique, Université de Strasbourg.)

Commandes

Pour ne pas surcharger la suite, j'utiliserai les définitions suivantes :

```

\def\tvi{\vrule height 9pt
           depth 4pt width 0pt}
\let\bl=\enspace

```

Le problème

La commande `\halign` permet de coder facilement un tableau à condition que celui-ci soit naturellement structuré en *lignes*. Or ce n'est pas toujours le cas. Certains tableaux se décrivent mieux en termes de *colonnes* :

abel	basile	céline	désiré
agnès	bernard	christian	diane
amandine	bertrand	corinne	
anatole	brigitte		
armance			

Vous pouvez, bien entendu, coder ce tableau en utilisant la commande `\halign`. Mais cela présente deux gros inconvénients :

- les noms ne se suivront pas naturellement lors du codage, ce qui sera une source de tension et d'erreurs ;
- que se passera-t-il si je vous demande d'insérer "aimé" dans la première colonne, entre "agnès" et "amandine" ?

Heureusement, Knuth a prévu ce cas. La version "verticale" de `\halign` s'appelle `\valign`. Voici le codage *naturel* du

tableau :

```

$$\hbox{%
\valign{&\hbox{\tvi\bl#\bl}\cr
abel&
agn\'es&      | brigitte\cr
amandine&     | c\'eline&
anatole&      | christian&
armance\cr    | corinne\cr
basile&       | d\'esir\'e&
bernard&      | diane\cr
bertrand&    | ]}$$

```

Commentaires :

1. Occupons-nous d'abord du codage du contenu du tableau. Nous voyons qu'il faut décrire ce tableau colonne par colonne. On commence par la première colonne :

```
abel&...&armance\cr
```

puis on passe à la deuxième colonne :

```
basile&...&brigitte\cr
```

et ainsi de suite. Le changement de ligne dans une colonne est signalé par un esperluète & et le changement de colonne par un \cr. L'insertion (ou la suppression) d'un nom dans une colonne ou d'une colonne toute entière est donc un jeu d'enfant.

2. Le préambule suit les mêmes lois que le préambule d'un \halign. Ici, et pour ne pas surcharger le codage, j'ai utilisé l'exception "&...#\cr" qui permet de répéter indéfiniment le même motif. (Je reviendrai sur ce sujet un peu plus loin.)
3. Il est essentiel, avec \valign, d'enchaîner les # du préambule dans des \hbox :

```
\hbox{...#\cr}&\hbox{...#\cr}&
... \cr
```

En effet, lorsque \valign compose le tableau, il *empile* le matériel obtenu en combinant le préambule et le codage de la première colonne. Par exemple, la première colonne de notre tableau est obtenue en empilant les boîtes

```

\hbox{\bl abel\bl}
\hbox{\bl agn\'es\bl}
:
\hbox{\bl armance\bl}

```

Il y a cependant une exception à cette règle : *les filets horizontaux* (dont je parlerai un peu plus loin) ne doivent pas être placés dans une \hbox.

4. Que se passerait-il en cas d'oubli des \hbox ? En T_EX, il y a deux règles essentielles : un caractère *n'est pas une boîte* et un caractère *est toujours contenu dans une boîte*. Si je ne précise pas moi-même la boîte, T_EX place le (ou les) caractères dans une \line. Dans le cas de notre tableau, le résultat sera catastrophique : sa première colonne sera de largeur \hsize. Les colonnes suivantes seront donc en dehors de la feuille de papier et nous serons abreuvés de "overfull \hboxes".
5. J'ai placé un \tvi dans les \hbox. Il s'agit d'un trait vertical invisible de dimensions convenables qui écarte (verticalement) les lignes du tableau. Une autre façon d'écarter les lignes consiste à modifier la valeur de \tabskip. Eh, oui ! tout ce qui est "horizontal" avec \halign devient "vertical" avec \valign et vice-versa.

L'écart (horizontal) entre les colonnes est obtenu en plaçant un \bl

au début et à la fin de chaque `\hbox` du préambule.

6. On ne peut pas placer un `\valign` dans une `\vbox`. En effet, cette macro fabrique des colonnes qui doivent être *juxtaposées* (il faut enfiler des perles sur un fil) : le mode horizontal est indispensable.

Pour centrer le tableau, il faut placer celui-ci dans une `\hbox` avant d'utiliser des mathématiques centrées :

```
$$\hbox{\valign{...}}$$
```

7. Complément de la remarque 2. Si je n'avais pas utilisé l'exception

```
\valign{&...#\cr...}
```

j'aurais dû transmettre à la macro `\valign` le *formatage complet* des colonnes du tableau ! En effet, un préambule tel que

```
\valign{...#\cr...#\cr...#\cr...#\cr...#\cr}
```

définit un tableau de trois lignes seulement...

Filets verticaux

Pour obtenir un filet vertical, on insère des `\noalign{\vrule}` entre les colonnes (c'est-à-dire immédiatement après un `\cr`). Pour que les filets verticaux ne touchent pas le contenu des colonnes, il faut penser à introduire du blanc au début et à la fin de chaque `\hbox` du préambule. Si l'on fait suivre chaque `\cr` du tableau initial (y compris le `\cr` du préambule) par un `\noalign{\vrule}` :

```
\noalign{\vrule} | brigitte\cr
abel&           | \noalign{\vrule}
...             | ...
armance\cr     | diane\cr
\noalign{\vrule} | \noalign{\vrule}
...             |
```

on obtient le résultat suivant :

abel	basile	céline	désiré
agnès	bernard	christian	diane
amandine	bertrand	corinne	
anatole	brigitte		
armance			

Les filets verticaux *partiels* que voici

abel	basile	céline	désiré
agnès	bernard	christian	diane
amandine	bertrand	corinne	
anatole	brigitte		
armance			

s'obtiennent en introduisant un petit filet vertical (`\tvi\vrule`) au début de la `\hbox` dans le préambule :

```
\valign{&\hbox{\tvi\vrule\bl#\bl}
\cr...}
```

Chaque cellule du tableau débute par un trait vertical. Comme un empilement de boîtes est justifié à gauche, ces petits traits verticaux se souderont pour former le filet vertical désiré (pas besoin de `\offinterlineskip`).

Filets horizontaux

Pour obtenir un filet horizontal, on a envie d'introduire un `\omit\hrule` (le `\omit` est indispensable car il désactive le préambule). Mais le codage

```
abel&           | basile&
\omit\hrule&   | bernard&
agn\'es&       | \omit\hrule&
anatole&       | bertrand&
amandine&      | ...
armance\cr     |
```

ne donne pas un résultat satisfaisant :

```

abel      basile   céline   désiré
           bernard christian diane
agnès      corinne
anatole    bertrand
amandine   brigitte
armance
    
```

Demandons alors à la macro `\valign` d'insérer les filets à notre place. Pour cela, modifions le préambule

```

\valign{
  &\hbox{\tvi\bl#\bl}&\hrule#\cr
...}
    
```

en ajoutant systématiquement un `\hrule` au-dessous de chaque nom. Maintenant, la commande `\valign` empile, dans chaque colonne, un nom, un filet, un nom, un filet, etc. Les filets sont correctement écartés des noms grâce au `\tvi`.

TRÈS IMPORTANT :

surtout, ne placez pas le `\hrule` dans une `\hbox` ! D'abord, un filet n'est pas un caractère. Mais la raison principale est qu'un `\hrule` est interdit dans une `\hbox` pour des raisons de conflit entre mode horizontal et vertical.

```

abel      basile   céline   désiré
agnès     bernard christian diane
anatole   bertrand corinne
amandine brigitte
armance
    
```

Ce résultat impeccable nous est fourni par un codage plus simple que le codage erroné

précédent :

```

\valign{
&\hbox{\tvi\bl#\bl}&\hrule#\cr
abel&&
agn\'es&&      |   brigitte&\cr
anatole&&      |   c\'eline&&
amandine&&     |   christian&&
armance&\cr    |   corinne&\cr
basile&&       |   d\'esir\'e&&
bernard&&      |   diane&\cr
bertrand&&     |   }
    
```

Bien entendu, on peut combiner filets horizontaux et filets verticaux. Si, dans l'exemple précédent, nous introduisons les traits verticaux `\noalign{\vrule}` après les `\cr`, nous obtiendrons ceci :

abel	basile	céline	désiré
agnès	bernard	christian	diane
anatole	bertrand	corinne	
amandine	brigitte		
armance			

Remarque : le premier filet horizontal ne fait pas partie du tableau. Il a été obtenu en empilant un `\hrule` et le tableau dans une `\vbox` :

```

\vbox{\hrule
      \hbox{\valign{...}}
}
    
```

Voici un dernier exemple plus sportif :
il s'agit d'obtenir le tableau irrégulier
suivant :

abel	basile	céline	désiré
agnès	bernard	christian	diane
amandine	bertrand	corinne	
anatole	brigitte		
armance			

Pour "boucher" la ligne "armance", il suffit d'introduire un nom "vide" au-dessous de "brigitte". Par contre, un problème se pose lorsqu'on veut boucher les lignes "désiré" et "diane". La solution précédente n'est pas correcte, car elle introduit un petit trait horizontal dû aux deux blancs `\bl` du préambule. Il faut désactiver le préambule. Pour cela, posons

```
\def\tv{\omit\hbox{\tvi\vrule}}
```

Ici, la `\hbox` est primordiale, car un `\vrule` exige d'être en mode horizontal.

```
$$
\def\tv{\omit\hbox{\tvi\vrule}}
\ vbox{\hrule\hbox{
\ valign{&
\ hbox{\tvi\vrule\bl#\bl\hss}&
\hrule#\cr
abel&&
agn\ 'es&&      |   c\ 'eline&&
amandine&&      |   christian&&
anatole&&       |   corinne&&\cr
armance&\cr     |   d\ 'esir\ 'e&&
basile&&        |   diane&&\cr
bernard&&       |   \tv&\omit&
bertrand&&      |   \tv&\omit\cr
brigitte&&\cr   |   }}}$$
```