

L'extension dcolumn*

David Carlisle

28/10/2014

Ce fichier est maintenu par l'équipe du « L^AT_EX Project ». Les rapports d'anomalie peuvent être envoyés en anglais à <http://latex-project.org/bugs.html> (catégorie `tools`).

Résumé

Cette extension définit des colonnes dans un tableau `array` ou `tabular` avec lesquelles l'alignement se fait sur le séparateur décimal.

Cette extension définit un spécificateur de colonne `D` ayant trois arguments `D{<sep.tex>}{<sep.dvi>}{<nombre de décimales>}` où :

- l'argument `<sep.tex>` doit être un unique caractère, qui sera utilisé comme séparateur dans le fichier `.tex`. Ce sera en général « . » ou « , » ;
- l'argument `<sep.dvi>` est utilisé comme séparateur dans la sortie, il peut être identique au premier argument, comme il peut être toute expression en mode mathématique, comme `\cdot`. Notez que `dcolumn` utilise toujours le mode mathématique pour composer les chiffres et le séparateur.
- l'argument `<nombre de décimales>` est le nombre maximal de chiffres placés après la virgule dans la colonne. Avec un argument à valeur négative, un nombre quelconque de chiffres pourra être utilisé dans la colonne, et toutes les entrées seront centrées sur le (bord gauche du) séparateur : notez que ceci peut rendre la colonne trop large, comme dans le cas pour la première colonne de l'exemple ci-dessous. Avec un argument à valeur positive, la colonne utilise des commandes équivalentes à `\rightdots` et `\endrightdots` de l'extension `array`, sinon les commandes sont essentiellement équivalentes à `\centerdots` et `\endcenterdots`.

Vous pouvez préférer ne pas préciser l'ensemble de ces trois entrées dans le préambule de `array` ou `tabular`, et vous pouvez ici définir vos propres spécificateurs pour le préambule en utilisant `\newcolumnntype`.

Par exemple, vous pouvez écrire :

```
\newcolumnntype{d}[1]{D{.}{\cdot}{#1}}
```

*Ce fichier a pour numéro de version v1.06 et a été mis à jour le 28/10/2014. La première traduction, basée sur la version v1.05, a été publiée par Jean-Pierre Drucbert en 2000.

Ici, le spécificateur « d » prend un seul argument, qui indique le nombre de chiffres décimaux. Le fichier .tex utilisera un « . », tandis que la sortie utilisera un « · ».

```
\newcolumnntype{.}{D{.}{.}{-1}}
```

Le spécificateur « . » indique une colonne d'entrées centrées sur le point.

```
\newcolumnntype{,}{D{,}{,}{2}}
```

Le spécificateur « , » indique une colonne d'entrée avec au plus deux décimales après la virgule « , ».

Avec cela, une table débutant par `\begin{tabular}{|d{-1}|d{2}|.|,|}` pourra donner par exemple :

| | | | |
|--------|--------|--------|--------|
| 1.2 | 1.2 | 1.2 | 1,2 |
| 1.23 | 1.23 | 12.5 | 300,2 |
| 1121.2 | 1121.2 | 861.20 | 674,29 |
| 184 | 184 | 10 | 69 |
| .4 | .4 | | ,4 |
| | | .4 | |

Notez que la première colonne, qui a un *nombre de décimales* négatif (premier argument du spécificateur d), est plus large que la seconde colonne. Le point décimal apparaît au milieu de la colonne. Notez aussi que les entrées sans virgule décimale, sans partie décimale ou sans partie entière, de même que les entrées vides sont correctement traitées.

Si vous avez des cellules avec entêtes (saisies avec `\multicolumn{1}{c}{..}` pour remplacer le type de colonne D), alors il se peut qu'aucune des deux formes « centrées » et « alignées à droite » ne soit satisfaisante.

| entête | entête | entête | | entête large | entête large | entête large |
|---------|---------|---------|--|--------------|--------------|--------------|
| 1.2 | 1.2 | 1.2 | | 1.2 | 1.2 | 1.2 |
| 11212.2 | 11212.2 | 11212.2 | | .4 | .4 | .4 |
| .4 | .4 | .4 | | | | |

Dans ces deux tables, la première colonne est définie par `D{.}{.}{-1}` pour afficher une colonne centrée sur le « . » ; la deuxième colonne est définie par `D{.}{.}{1}` pour afficher une colonne alignée à droite.

La colonne centrée produit une colonne qui plus large que nécessaire pour s'assurer que le point décimal est centré dans tous les cas. La deuxième colonne, qui est alignée à droite, n'a pas cet inconvénient mais, sous un entête large, l'apparence visuelle est assez peu satisfaisante lorsqu'il y a peu de chiffres dans les colonnes.

Dans la version v1.03, une troisième alternative a été introduite. Le troisième argument, le *nombre de décimales*, peut définir *à la fois* le nombre de chiffres à gauche et à droite de la virgule. La troisième colonne ainsi est définie par `D{.}{.}{5.1}` dans la table de gauche ci-dessus et par `D{.}{.}{1.1}` dans la table de droite, pour indiquer respectivement « cinq chiffres à gauche et un à droite » et « un chiffre à gauche et un chiffre à droite » (le « . », lui, peut être remplacé par « , » ou toute autre unité lexicale). Les colonnes de chiffres sont alors positionnées de telle sorte qu'un nombre avec le format indiqué soit centré dans la colonne.

Cette notation permet également de centrer les colonnes sur le milieu du séparateur plutôt que sur son bord gauche ; par exemple, `D{+}{\,\pm\,}{3,3}` donnera une belle mise en forme symétrique pour des nombres comportant jusqu'à trois chiffres de chaque côté d'un signe \pm .

1 Les commandes

1 `\package`

Tout d'abord, nous chargeons l'extension `array` si ce n'est pas déjà le cas.

2 `\RequirePackage{array}`

Les idées de base derrière ces commandes sont expliquées dans la documentation de l'extension `array`. Toutefois, elles se servent de trois astuces qui peuvent être utiles dans d'autres contextes.

- le séparateur est entouré par une paire d'accolade supplémentaire afin qu'il bénéficie de l'espacement de `\mathord`, sinon, par exemple, un séparateur comme la virgule aurait une espace en plus après lui.
- le séparateur ne se voit pas attribuer sa définition spéciale en le rendant actif car cela ne fonctionnerait pas avec une entrée comme « `& .5 &` » dans la mesure où la première unité lexicale d'une entrée d'alignement est lue *avant* la partie en préambule dans le où ce serait `\omit`, auquel cas le préambule doit être omis. En lieu et place, nous basculons le code mathématique à 8000 (en hexadécimal), ce qui fait que l'unité lexicale agit comme si elle était active.
- Bien que `\mathcode'."8000` fait agit « `.` » comme s'il était actif, il n'est pas autorisé dans les constructions telles que `\def.{}`, même en mode mathématique, aussi nous avons à construire une version active du séparateur, ce qui est obtenu en en faisant la version majuscule de « `~` », et en utilisant la construction `\uppercase{\def~}{\langle définition \rangle}`.
Notez que la *⟨définition⟩* n'est pas mise en majuscule, ce qui fait que la définition peut faire référence à la version standard non active du séparateur.

`\DC@` Cette commande met à jour la table de majuscules comme souhaité et récupère alors la première partie de l'argument numérique dans `\count@`.

```
3 \def\DC@#1#2#3{%
4   \uccode'\~='#1\relax
5   \m@th
6   \afterassignment\DC@x\count@#3\relax{#1}{#2}}
```

`\DC@x` Si `\count@` est négatif, tout est centré sur la virgule décimale. S'il est positif, soit `#1` est vide auquel cas la partie décimale est étendue au nombre de décimales indiqué par `\count@` ou (nouvelle fonctionnalité de la version v1.03) il n'est pas vide, auquel cas `\count@` contient le nombre de chiffres à gauche de la virgule et `#1` contient une unité lexicale inutile (probablement « `.` ») suivie par le nombre de décimales à droite de la virgule. Dans ces deux derniers cas, `\DC@right` est utilisé.

```
7 \def\DC@x#1\relax#2#3{%
```

```

8 \ifnum\z@>\count@
9 \expandafter\DC@centre
10 \else
11 \expandafter\DC@right
12 \fi
13 {\#2}{\#3}{\#1}}

```

`\DC@centre` Si les éléments sont centrés autour de la virgule décimale, il faut mettre en boîte les deux moitiés de la cellule.

```

14 \def\DC@centre#1#2#3{%
15 \let\DC@end\DC@endcentre
16 \uppercase{\def~}{\egroup\setbox\tw@=\hbox\bgroup${\#2}}}%
17 \setbox\tw@=\hbox{${\phantom{{\#2}}}}$}%
18 \setbox\z@=\hbox\bgroup$\mathcode'\#1="8000 }

```

`\DC@endcentre` et la plus petite des deux boîtes est augmentée afin qu'il y ait la même quantité d'éléments de deux côtés de la virgule.

```

19 \def\DC@endcentre{\egroup
20 \ifdim \wd\z@>\wd\tw@
21 \setbox\tw@=\hbox to\wd\z@{\unhbox\tw@\hfill}%
22 \else
23 \setbox\z@=\hbox to\wd\tw@{\hfill\unhbox\z@}\fi
24 \box\z@\box\tw@}

```

`\DC@right` Ceci traite les deux cas où un nombre de décimales est donné.

```

25 \def\DC@right#1#2#3{%
26 \ifx\relax#3\relax

```

Si `#3` est vide, une commande `\hfill` est ajoutée par aligner la colonne à droite. La commande `\DC@rl` est mise pour débiter un groupe afin que rien de farfelu ne se passe avec la partie entière du nombre.

```

27 \hfill
28 \let\DC@rl\bgroup
29 \else

```

Sinon `\DC@rl` est définie pour la partie entière soit mise dans une boîte large de `\count@` fois la largeur d'un chiffre. Afin de partager du code avec l'autre branche possible, le paramètre `#3` (le nombre de décimales) est placé dans `\count@` en se débarrassant du « . » de l'utilisateur.

```

30 \edef\DC@rl{to\the\count@\dimen@ii\bgroup\hss\hfill}%
31 \count@\@gobble#3\relax
32 \fi
33 \let\DC@end\DC@endright

```

La boîte 2 (`\tw@`) contient la partie décimale et est définie comme étant de largeur `\dimen@` qui est calculée ci-dessous : `\count@` fois la largeur d'un chiffre plus la largeur de la « virgule décimale ».

```

34 \uppercase{\def~}{\egroup\setbox\tw@\hbox to\dimen@\bgroup${\#2}}}%
35 \setbox\z@\hbox{${\#1}}\dimen@ii\wd\z@

```

```

36 \dimen@ \count@ \dimen@ii
37 \setbox\z@\hbox{\$ \{#2\}} \advance \dimen@ \wd \z@
38 \setbox\tw@\hbox to \dimen@{\}%

```

La boîte zéro (`\z@`) contient la partie entière, soit juste à sa taille naturelle pour les colonnes alignées à droite ou soit mise à (l'ancienne valeur de) `\count@` fois la largeur d'un chiffre. La commande `\DC@rl` définie plus haut distingue les deux cas.

```

39 \setbox\z@\hbox\DC@rl$\mathcode'#1="8000 }

```

`\DC@endright` Cette commande finit la seconde boîte et restitue les deux boîtes.

```

40 \def\DC@endright{\$ \hfil \egroup \box\z@ \box\tw@}

```

- D L'interface utilisateur passe par la colonne D qui prend les trois arguments. Pour des traitements particuliers, vous pouvez utiliser directement `\DC@` plutôt que la colonne D, par exemple pour obtenir une version en gras des fonctionnalités vues ici :

```

\newcolumnntype{E}[3]{>\boldmath\DC@{#1}{#2}{#3}}c<\DC@end}

```

```

41 \newcolumnntype{D}[3]{>\DC@{#1}{#2}{#3}}c<\DC@end}

```

```

42 \endpackage

```