Chronique 11

Encore des tableaux

On a plusieurs fois évoqué les tableaux dans ces chroniques ; les environnements `tabular` et `array` donnent déjà des résultats intéressants ; voir :
saison 1 chronique 9 (http://p8.storage.canalblog.com/81/36/1082582/85069648.pdf) et
saison 1 chronique 10 (http://p8.storage.canalblog.com/82/43/1082582/86268737.pdf).

Dans cette chronique-ci on va voir quelques astuces supplémentaires ainsi que l'utilisation d'un package spécialisé dans les tableaux : `tabularx`.

11.1 Quelques compléments

11.1.1 Résolution d'équation

Voyons une façon de présenter proprement la résolution d'une équation.

On résout l'équation :	$2x+3$	$=$	$4x-11$
On ajoute 11 aux deux membres :	$2x+14$	$=$	$4x$
On retranche $2x$ aux deux membres :	14	$=$	$2x$
On divise par 2 les deux membres :	7	$=$	x
Donc :	x	$=$	7

Ce tableau est fait au moyen de l'environnement `tabular` qui définit quatre colonnes : celle du texte (alignée à gauche), celle du membre de gauche (alignée à droite), celle du signe = (centrée) et celle du membre de droite (alignée à gauche). Le résultat est obtenu par :

```
\begin{tabular}{l r c l}
On résout l'équation:              & $2x+3$  & = & $4x-11$ \\
On ajoute 11 aux deux membres:     & $2x+14$ & = & $4x$    \\
On retranche $2x$ aux deux membres: & $14$    & = & $2x$    \\
On divise par 2 les deux membres:  & $7$     & = & $x$     \\
Donc:                              & $x$     & = & $7$
\end{tabular}
```

On peut remarquer que le texte ne commence pas tout à fait le long de la marge gauche mais qu'il y a un petit décalage ; il suffit de rajouter `@{}` au début de la définition des colonnes pour l'éviter :

```
\begin{tabular}{@{} l r c l}
```

On peut également trouver que l'espacement autour du signe = est un peu grand. Pour remédier à cet inconvénient, on peut supprimer la colonne contenant le signe = et faire afficher ce signe = au changement de colonne ; cela se fait au moyen de `@{\,=\,}` qui signifie à peu près : quand on appuie sur le caractère &, il s'affiche le signe = entouré de deux petites espaces créées par `\,` :

```
\begin{tabular}{@{} l r @{\,=\,} l}
```

Si on ne mettait pas les espaces, ce serait moche !
On peut encore améliorer les choses en utilisant une propriété contenue dans le package `array` ; à la place de l'arobase et des espaces, on peut mettre le point d'exclamation (sans les espaces cette fois). On a alors le code suivant :

```
\begin{tabular}{@{} l r !{=} l}
On résout l'équation:                & $2x+3$  & $4x-11$ \\
On ajoute 11 aux deux membres:       & $2x+14$ & $4x$    \\
On retranche $2x$ aux deux membres:  & $14$    & $2x$    \\
On divise par 2 les deux membres:    & $7$     & $x$     \\
Donc:                                & $x$     & $7$
\end{tabular}
```

qui donne le résultat suivant :

On résout l'équation	$2x+3$	$=$	$4x-11$
On ajoute 11 aux deux membres	$2x+14$	$=$	$4x$
On retranche $2x$ aux deux membres	14	$=$	$2x$
On divise par 2 les deux membres	7	$=$	x
Donc	x	$=$	7

Une remarque à propos du package `array` ; il est chargé automatiquement par le package `numprint`, mais également par le package `tabularx` que l'on verra dans le paragraphe suivant ; si au moins l'un de ces deux packages est chargé, c'est inutile de charger le package `array`.

On peut se rendre compte que, dans ce tableau, les colonnes 2 et 3 ne contiennent que des formules mathématiques et qu'il faut donc à chaque fois entrer le signe $ deux fois.
On peut automatiser le processus en disant au tableau que ces deux colonnes seront écrites en mode mathématique, et donc on ne sera pas obligé d'écrire chaque cellule en mode mathématique. Il suffit d'écrire `>{$}` devant la définition de la colonne (entrée en mode mathématique) et `<{$}` derrière (sortie du mode mathématique).
Voici le code complet du tableau :

```
\begin{tabular}{@{} l >{$}r<{$} !{=} >{$}l<{$}}
On résout l'équation:                & 2x+3  & 4x-11 \\
On ajoute 11 aux deux membres:       & 2x+14 & 4x    \\
On retranche $2x$ aux deux membres:  & 14    & 2x    \\
On divise par 2 les deux membres:    & 7     & x     \\
Donc:                                & x     & 7
\end{tabular}
```

Pour finir, on peut écrire la première colonne en rouge (pourquoi pas ?) ; voilà comment on la définira :

```
\begin{tabular}{@{} >{\red}l >{$}r<{$} !{=} >{$}l<{$}}
```

11.1.2 Tableau de largeur fixe

On peut, avec les environnements `tabular` et `array`, créer des tableaux ayant des colonnes de largeurs fixes dans lesquelles le passage à la ligne se fait de façon automatique (si besoin est) :

1	2	3
colonne de largeur 3 cm	largeur 5 cm	cette troisième colonne a une largeur de 4 cm cette troisième colonne a une largeur de 4 cm

C'est `p{3cm}` qui permet de créer une colonne de largeur 3 cm et le texte est justifié dans chaque cellule :

```
\begin{tabular}{|p{3cm} | p{5cm} | p{4cm}|}
\hline
1 & 2 & 3\\
\hline
colonne de largeur 3 cm
 & largeur 5 cm
 & cette troisième colonne a une largeur de 4 cm
   cette troisième colonne a une largeur de 4 cm   \\
\hline
\end{tabular}
```

Le package `array` rajoute deux formats de paragraphes, `m{...}` et `b{...}` que je ne maitrise pas encore ! Une autre fois peut-être...

Avec `p{...}` les colonnes sont de largeurs fixées, mais le texte n'est pas centré horizontalement dans chaque cellule. On peut le faire (pour la première colonne par exemple) :

1	2	3
colonne de largeur 3 cm	largeur 5 cm	cette troisième colonne a une largeur de 4 cm cette troisième colonne a une largeur de 4 cm

La commande à rajouter est `>{\centering\arraybackslash}` devant la définition de la colonne, mais on peut naturellement redéfinir cette commande en lui donnant un nom plus court pour l'appliquer à toutes les colonnes ; voici ce que cela donne :

1	2	3
colonne de largeur 3 cm	largeur 5 cm	cette troisième colonne a une largeur de 4 cm cette troisième colonne a une largeur de 4 cm

en entrant le code :

```
\newcommand{\ca}{\centering\arraybackslash}
\begin{tabular}{|>{\ca}p{3cm} | >{\ca}p{5cm} | >{\ca}p{4cm}|}
\hline
1 & 2 & 3\\
\hline
colonne de largeur 3 cm
& largeur 5 cm
& cette troisième colonne a une largeur de 4 cm
  cette troisième colonne a une largeur de 4 cm \\
\hline
\end{tabular}
```

Enfin on peut centrer un texte dans une cellule sans centrer toute la colonne, et le plus simple est d'entrer `\hfill` devant le texte, et `\hfill\null` derrière le texte.

11.2 Avec le package tabularx

On a vu qu'on pouvait déjà faire plein de choses avec les environnements `tabular` et `array`. David Carlisle, qui avait déjà travaillé sur les deux environnements précédents, en a créé un nouveau appelé `tabularx`.

On le charge en entrant `\usepackage{tabularx}` dans le préambule du document. Ce package est inclus dans la plupart des distributions LaTeX donc pas besoin de le télécharger !

Cet environnement permet essentiellement (et c'est déjà bien !) de créer un tableau dont on fixe la largeur totale ; les largeurs des colonnes sont alors ajustées en fonction de leur nombre.
Si on veut un tableau sur toute la largeur de la page, on prendra comme largeur la variable `\linewidth` ; sinon on peut définir la largeur en centimètres, points...

Voyons quelques exemples qui parleront mieux que de longs discours.

- Quatre colonnes de largeurs identiques sur toute la largeur de la ligne :

quatre colonnes de largeurs identiques ; première colonne	quatre colonnes de largeurs identiques ; deuxième colonne	quatre colonnes de largeurs identiques ; troisième colonne	quatre colonnes de largeurs identiques ; quatrième colonne

Et le code qui génère le tableau précédent :

```
\begin{tabularx}{\linewidth}{|X|X|X|X|}
\hline
quatre colonnes de largeurs identiques; première colonne
& quatre colonnes de largeurs identiques; deuxième colonne
& quatre colonnes de largeurs identiques; troisième colonne
& quatre colonnes de largeurs identiques; quatrième colonne
\\
\hline
\end{tabularx}
```

On peut forcer un changement de ligne dans une cellule en entrant `\newline`.

- Une colonne de largeur 2 cm et quatre colonnes de largeurs identiques pour que la largeur totale soit la largeur de la ligne s'obtient avec :

```
\begin{tabularx}{\linewidth}{|p{2cm}|*{4}{X|}}
```

numéro	1	2	3	4
lettre	a	b	c	d

- Une colonne centrée de largeur ajustable et trois colonnes centrées pour que la largeur totale soit la largeur de la ligne, s'obtient avec :

```
\newcommand{\ca}{\centering\arraybackslash}
\begin{tabularx}{\linewidth}{|c| *{3}{>{\ca}X|}}
```

numéros	1	2	3
texte	aa	bbb	ccc

- Le package `tabularx` est compatible avec la fusion de cellules horizontalement :

numéros	327		45
texte	aa	bbb	ccc

obtenu par :

```
\begin{tabularx}{\linewidth}{|c| *{3}{>{\ca}X|}}
\hline
numéros & \multicolumn{2}{c|}{327} & 45\\
\hline
texte & aa & bbb & ccc\\
\hline
\end{tabularx}
```

À vous de faire de beaux tableaux tels que vous les voulez !