

# Corrigé de l'exercice 7 — Im204

Manuel PÉGOURIÉ-GONNARD

31 octobre 2008

## 1 Préambule

Début classique :

```
1 \documentclass[a4paper, 11pt]{article}
2 \usepackage[latin1]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{textcomp}
```

On modifié la taille de fonte de base à `11pt` comme demandé dans l'énoncé (ligne ??). On charge maintenant `geometry` comme l'énoncé (ligne ??) le demande aussi.

```
5 \usepackage{geometry}
```

On charge encore tous les modules relatifs aux tableaux dont on a parlé en cours et qu'on va utiliser ici :

```
6 \usepackage{array, multirow, hhline, tabularx, slashbox, graphicx}
```

(`graphicx` n'est pas précisément spécifique aux tableaux mais on verra quand même ligne ?? qu'il peut être utile dans ce cadre).

On charge enfin `amsthm` pour définir facilement l'environnement `diff` utilisé pour signaler les exercices difficiles, puis les paquets de la fin.

```
7 \usepackage{amsthm}
8 \usepackage{xspace}
9 \usepackage[british, frenchb]{babel}
```

On utilise alors `\newtheorem`, fournit par `amsthm`, pour définir un environnement `diff`. La mise en forme standard des théorèmes nous convient, par contre on utilise l'étoile pour supprimer la numérotation.

```
11 \newtheorem*{diff}{Plus difficile}
```

## 2 Corps du document

```
13 \begin{document}
```

Le paragraphe d'instructions est centré, en fonte un peu plus grosse que le reste du texte. Comme d'habitude, on profite de l'environnement pour restreindre la portée de la commande `\large` sans avoir à rajouter d'accolades. Au sein des instructions, comme toujours dans `center`, on peut forcer un retour à la ligne avec `\\`.

```
15 \begin{center} \large
16 \textbf{Instructions}\\
17 Certains exercices sont signalés comme plus difficiles : n'hésitez pas à
18 les passer dans un premier temps. Toujours dans un premier temps, ne
19 reproduisez que les tableaux. Quand vous remplirez le texte, pensez à
20 définir un environnement \og diff \fg pour les exercices difficiles.\\
21 La police principale fait 11 points
22 et le module \textsf{geometry} a été chargé
23 pour modifier les marges.
24 \end{center}
```

Remarquez que la mise en forme n'est pas très élégante. Quand on fait des textes centrés, et qu'on veut être soigneux, on peut avoir intérêt (vu que la partie centrée est souvent courte) à choisir soi-même toutes les coupures de ligne pour obtenir une forme harmonieuse. Dans le contexte d'un exercice, ce qui comptait était que les coupures de lignes forcées soient aisément identifiables.

```
26 Un tableau simple pour commencer. Attention, il ne va pas se centrer tout
27 seul. (La première colonne est alignée à droite, la deuxième centrée, et la
28 troisième alignée à gauche.)
```

Comme indiqué, il faut penser à l'environnement `center`. Les instructions donnent les types de colonnes à utiliser : `r`, `c` et `l` dans cet ordre. En rajoutant les réglures verticales, on obtient le préambule `r|c|l`. Il n'est pas question ici d'utiliser les types de colonnes plus compliqués (comme `p`) qui sont parfaitement inutiles pour les cas simples.

```
30 \begin{center}
31 \begin{tabular}{r|c|l}
32 \textbf{Colonne 1} & \textbf{Colonne 2} & \textbf{Colonne 3} \\ \hline
33 Cellule 1a & Cellule 1b & Cellule 1c \\
34 Cellule 2a & Cellule 2b & Cellule 2c \\
35 \end{tabular}
36 \end{center}
```

Il n'y a pas moyen de mettre d'un seul coup en gras toute une ligne : il faut le faire cellule par cellule. C'est un peu pénible mais c'est comme ça.

```
38 Une version plus chargée du même tableau : la version précédente était plus
39 élégante. Cette fois par contre, aligné à gauche. (Penser à faire du
40 copier-coller\dots)
```

Encore une fois, penser à `\dots` dans le paragraphe précédent.

Pour la prochaine version, on rajoute des filets : verticaux dans le préambule, et horizontaux dans le corps du tableau. Il faut bien penser à terminer la dernière ligne par `\\` avant de placer le dernier filet avec `\hline`. Par ailleurs, pour l'alignement à gauche, on utilise `flushleft` qui s'occupe aussi de rajouter un peu d'espace vertical.

```

42 \begin{flushleft}
43   \begin{tabular}{|r|c|l|} \hline
44     \textbf{Colonne 1} & \textbf{Colonne 2} & \textbf{Colonne 3} \\
45     \hline\hline
46     Cellule 1a & Cellule 1b & Cellule 1c \\
47     Cellule 2a & Cellule 2b & Cellule 2c \\
48   \end{tabular}
49 \end{flushleft}

```

Pour taper le paragraphe suivant signalant un exercice difficile, on utilise l'environnement `diff` défini **ligne ??**. Ceci assure automatiquement l'insertion de la mention « plus difficile » et la mise en forme du texte.

```

51 \begin{diff}
52   En consultant la documentation du module \textsf{hhline}, trouvez comment
53   obtenir le résultat suivant (les filets verticaux ne sont pas
54   interrompus).

```

Si vous avez des difficultés pour trouver la documentation de `hhline`, reportez-vous au document d'aide **doc-trouver-la-doc.pdf** disponible à l'adresse habituelle. Une fois que vous l'avez trouvée, vous constatez qu'elle n'est pas forcément très lisible mais qu'elle a l'avantage d'être courte. Vous pouvez donc la parcourir assez rapidement et réaliser qu'il y a un exemple. À l'aide de cet exemple et des explications du début sur les significations des symboles, on arrive (éventuellement après quelques essais malheureux) au résultat suivant.

```

56 \begin{center} \normalfont
57   \begin{tabular}{|r|c|l|} \hline
58     \textbf{Colonne 1} & \textbf{Colonne 2} & \textbf{Colonne 3} \\
59     \hhline{|=|=|}
60     Cellule 1a & Cellule 1b & Cellule 1c \\
61     Cellule 2a & Cellule 2b & Cellule 2c \\
62   \end{tabular}
63 \end{center}
64 \end{diff}

```

Un dernier piège se glisse ici : vu que dans l'environnement `diff`, tout est en italique, il faut rétablir la police par défaut à l'aide de `\normalfont`. Il faut de plus le faire *avant* le début de `tabular` : sinon, on sera déjà dans la première case du tableau, et on ne change donc la fonte que pour cette case et pas tout le tableau.

```

66 Dans le tableau suivant, faites en sorte que les cellules soient toutes
67 automatiquement en mode mathématique.

```

Une précision d'abord sur cette consigne : si l'on ne passe pas en mode mathématique, on a bien sûr des messages d'erreur sur les symboles comme `\pi` ou les opérations comme `^`. Il faut donc *a priori* saisir un `$` au début de chaque cellule et un `$` à la fin, ce qui est vite énervant. On utilise donc les spécificateurs pré- et post-cellule fournis par `array`. Si on n'avait pas pensé à le charger **ligne ??**, on aurait le message d'erreur :

! LaTeX Error: Illegal character in array arg.

Comme ceci est pénible à écrire trois fois de suite, on utilise en plus la syntaxe avec `*` pour se simplifier la vie. Au final, on a

```
69 \begin{center}
70   \begin{tabular}{*{3}{>{\(}c{)}}|}
71     a^2 - b^2 & 1 & \sqrt{2} \\ \hline
72     1 & 0 & -2 \\ \hline
73     \pi & \pi^2 - 1 & \pi^2 - 2 \\ \hline
74   \end{tabular}
75 \end{center}
```

Remarque que j'ai choisi d'utiliser `\(...\)` pour le mode mathématique, mais j'aurais plus aussi utiliser `$....$`, c'est une question de goûts.

```
77 \begin{diff}
78   La commande \verb|\arraystretch| a normalement pour définition \og 1 \fg.
79   C'est un facteur qui contrôle l'espace entre les lignes d'un tableau.
80   En le redéfinissant localement avec 1.2 pour valeur, obtenez la version
81   suivante, plus satisfaisante, du tableau précédent.
```

En d'autres termes, `\arraystretch` est normalement défini de la façon<sup>1</sup> suivante :

```
\newcommand\arraystretch{1}
```

C'est une commande comme une autre<sup>2</sup> (même si elle est utilisée en interne pas  $\text{\LaTeX}$  pour faire les tableaux) : on peut donc la redéfinir avec `\renewcommand`. Il faut faire cette redéfinition avant le début du tableau pour qu'elle agisse. On utilise l'environnement `center` pour restreindre la portée de cette opération et éviter qu'elle n'agisse sur tous les tableaux à venir.

```
82 \begin{center} \normalfont \renewcommand\arraystretch{1.2}
83   \begin{tabular}{*{3}{>{\(}c{)}}|}
84     a^2 - b^2 & 1 & \sqrt{2} \\ \hline
85     1 & 0 & -2 \\ \hline
86     \pi & \pi^2 - 1 & \pi^2 - 2 \\ \hline
87   \end{tabular}
88 \end{center}
89 \end{diff}
```

```
91 La colonne du milieu fait la moitié de la largeur totale de la ligne (donnée
92 par \verb|\linewidth|) et celle de droite fait 2cm, dans le tableau suivant.
```

Comme ces colonnes doivent contenir du texte sur plusieurs lignes, elles doivent être de type `p`. Les indications ci-dessus fournissent les largeurs à indiquer pour ces colonnes.

```
94 \begin{center}
95   \begin{tabular}{rp{.5\linewidth}p{2cm}}
96     Raoul Volfoni & L'alcool à c't'âge-là ! \\
```

1. En fait, les commandes utilisées sont différentes mais le résultat est équivalent.

2. Vous découvrez ainsi un point particulier de la programmation  $\text{\LaTeX}$  : une commande peut être utilisée pour stocker des valeurs, en d'autres termes  $\text{\TeX}$  ne distingue pas les variables des fonctions.

```

97 Monsieur Fernand & C'est un scandale hein ? \\
98 Raoul Volfoni & Nous par contre, on est des adultes, on pourrait peut
99 être s'en faire un petit ? \\
100 Monsieur Fernand & Ça\dots le fait est\dots Maître Folace ? \\
101 Maître Folace & Seulement, le tout-venant a été piraté par les mêmes.
102 Qu'est ce qu'on fait ? on se risque sur le bizarre ?\dots Ça va rajeunir
103 personne.
104 & (sort la bouteille) \\
105 Raoul Volfoni & Ben nous voilà sauvés. \\
106 Maître Folace & Sauvés\dots Faut voir ! \\
107 Jean & Tiens, vous avez sorti le vitriol ? & (regards inquiets)
108 \end{tabular}
109 \end{center}

```

Procédons par ordre pour améliorer le tableau. Les instructions sont les suivantes :

```

111 On va maintenant améliorer un peu. Faisons en sorte que les noms soient
112 automatiquement en petites capitales, les commentaires en italiques, et
113 qu'on ait pas à insérer les parenthèses dans la troisième colonne, qui
114 devrait par ailleurs être alignée à gauche. On va ensuite ajouter un titre
115 sur plusieurs colonnes. Puis, pour profiter automatiquement de toute la
116 place disponible pour la colonne du milieu, on va utiliser
117 \texttt{tabularx}. On rajoute enfin quelques filets pour la décoration. Vous
118 n'êtes pas obligés de tout faire d'un coup. Au final, on obtient :

```

Déjà, on va utiliser `tabularx` de façon à profiter de tout l'espace disponible pour la colonne du milieu, le tableau devant occuper toute la largeur de la ligne. Si on oublie l'argument obligatoire `{\linewidth}`, on obtient le message

! Missing number, treated as zero.

caractéristique de cette erreur courante en passant de `tabular` à `tabularx`. Commençons donc par

```

119 \begin{center}
120 \begin{tabularx}{\linewidth}{%

```

puis attaquons les spécifications de colonnes calmement une par une. Pour que la première colonne soit toute en petites capitales, on a juste à ajouter `\scshape` au début, grâce au spécificateur `>` fournit par le module `array`.

```

121 {\scshape}r

```

Vient ensuite la colonne centrale, qui doit occuper tout l'espace disponible et est donc de type `X` :

```

122

```

puis la colonne de droite, toujours de type `p{2cm}`, mais où on utilise de plus `>` et `<` pour insérer les parenthèses, mettre en italique, et aligner à gauche ; de plus `\arraybackslash` après le `\raggedright` permet de terminer la ligne normalement par `\\` : il faudrait utiliser la commande `\tabularnewline` sinon, comme expliqué en cours.

123 `{\raggedright\arraybackslash\itshape{ }p{2cm}<{}}}`

Remarquons que la version précédente du tableau provoquait un problème

Underfull \hbox (badness 10000) in paragraph at lines 101--102

parce que  $\text{\LaTeX}$  n'arrivait pas à justifier correctement le texte dans une colonne aussi étroite sans créer un espace démesuré entre les mots « sort » et « la ». Il s'agit maintenant d'écrire le titre centré au-dessus de toutes les colonnes, entre deux filets horizontaux.

124 `\hline`

125 `\multicolumn{3}{c}{\bfseries Extrait des tontons flingueurs} \\`

126 `\hline`

Le corps du tableau reste identique, à l'exception du `\hline` final.

127 `Raoul Volfoni & L'alcool à c't'âge-là ! \\`

128 `Monsieur Fernand & C'est un scandale hein ? \\`

129 `Raoul Volfoni & Nous par contre, on est des adultes, on pourrait peut`

130 `être s'en faire un petit ? \\`

131 `Monsieur Fernand & Ça\ldots le fait est\ldots Maître Folace ? \\`

132 `Maître Folace & Seulement, le tout-venant a été piraté par les mêmes.`

133 `Qu'est ce qu'on fait ? on se risque sur le bizarre ?\ldots Ça va rajeunir`

134 `personne. & sort la bouteille \\`

135 `Raoul Volfoni & Ben nous voilà sauvés. \\`

136 `Maître Folace & Sauvés\ldots Faut voir! \\`

137 `Jean & Tiens, vous avez sorti le vitriol ? & regards inquiets \\ \hline`

138 `\end{tabularx}`

139 `\end{center}`

Le fait que le corps du tableau soit resté presque identique alors que l'on a modifié la mise en forme du contenu illustre une des grandes forces de  $\text{\LaTeX}$  : la séparation fond-forme. Je ne saurais trop vous encourager à utiliser les outils fournis par  $\text{\LaTeX}$  pour mettre en œuvre ce type de conception du document.

141 Le tableau suivante flotte, donc vous ne le verrez qu'en haut de la

142 page~\pageref{tab-couleurs} : c'est la table~\ref{tab-couleurs}. Utilisez

143 `\verb|\multicolumn|`, et évitez d'avoir à écrire trois fois le motif pour les

144 trois dernières colonnes.

On a ici utilisé des références au label qui sera défini **ligne ??**. C'est d'ailleurs la raison pour laquelle il faut deux compilations pour avoir des références correctes : certains labels sont définis après avoir été utilisés. Écrivons donc maintenant le code de ce tableau qui apparaîtra en haut de la page suivante (rappelons qu'au moment où on écrit le code, on ne sait pas du tout si on est proches ou pas du bas de la page).

146 `\begin{table}[t] \centering`

147 `\begin{tabular}{|r|*{3}{c|}}`

148 `\cline{2-4}`

149 `\multicolumn{1}{c|}{ } & Rouge & Jaune & Bleu \\ \hline`

```

150   Rouge                & Rouge & Orange & Violet \\ \hline
151   Jaune                & Orange & Jaune & Vert \\ \hline
152   Bleu                 & Violet & Vert & Bleu \\ \hline
153   \end{tabular}
154   \caption{Mélanges de couleurs} \label{tab-couleurs}
155   \end{table}

```

Pour les flottants, il est plus efficace de faire le centrage en utilisant la commande `\centering` que l'environnement `center` : c'est plus court à écrire, et l'environnement `center` ajouterait des espaces verticaux superflus. Les techniques particulières ici sont le `\cline` et surtout le `\multicolumn` sur la case vide qui sert uniquement à supprimer le filet vertical à gauche de la case. Attention, il faut trois arguments : même si le contenu est vide, on ne peut pas supprimer la dernière paire d'accolades `}`. Pour l'exemple, on a demandé avec l'argument optionel `[t]` de `table` que le tableau soit placé en haut d'un page à l'exclusion de tout autre emplacement.

Avant le paragraphe suivant, on laisse un peu d'espace avec `\medskip`.

```

157 \medskip
158 Pour le tableau d'après, utilisons le module \verb|slashbox|.
159 \begin{center}
160   \begin{tabular}{|r|c|c|} \hline
161     \backslashbox{Outil}{Caractéristiques} & Qualités & Défauts \\ \hline
162     \LaTeX & Grandes & Surmontables \\ \hline
163     Word & Moyennes & Agaçants \\ \hline
164   \end{tabular}
165 \end{center}

```

Ce tableau n'appelle pas beaucoup de commentaires, à part que la commande `\backslashbox` ne sera pas disponible si on oublie de charger `slashbox` comme on l'a fait à la **ligne ??**.

La question suivante est un peu plus difficile mais intéressante, car la présentation ci-dessus est quand même assez moche. On utilise encore l'environnement `diff` pour l'énoncé, et on pense à rétablir la police normale avant le tableau.

```

167 \begin{diff} Obtenir la présentation alternative suivante. Pensez aux
168   colonnes \og invisibles \fg, utilisez \verb|\multirow| et
169   \verb|\rotatebox|.
170   \begin{center} \normalfont
171     \begin{tabular}{r|r|c|c|c|}

```

Comme le dit l'énoncé, il faut prévoir une colonne supplémentaire pour la légende « outil » : c'est le premier `r` qui joue ce rôle. Il faudra faire bien attention aux filets verticaux. Sur la première ligne, on utilise deux `\multicolumn` : un premier juste pour supprimer les filets des colonnes non utilisées, un deuxième pour le titre.

```

172 \multicolumn{2}{c}{}
173 \multicolumn{3}{c}{\bfseries Caractéristiques} \\ \cline {3-5}
174 \multicolumn{2}{c}{}
175 Qualités & Défauts & Prix \\ \cline{2-5}

```

Sur la deuxième ligne, on utilise la même astuce pour supprimer le premier filet vertical. Entre la première et la deuxième, puis la deuxième et la troisième ligne, `\cline` permet de contrôler l'étendue des filets horizontaux. Vient maintenant la partie nouvelle : l'utilisation conjointe de `\multirow` et `\rotatebox` pour écrire le mot « outil » verticalement, sur l'étendue de plusieurs colonnes.

```
176 \multirow{3}{*}{\rotatebox{90}{\bfseries Outil}}
177 \LaTeX & Grandes & Surmontables & Gratuit \\ \cline{2-5}
178 OoWriter & Moyennes & Agaçants & Gratuit\\ \cline{2-5}
179 Word & Moyennes & Agaçants & Cher \\ \cline{2-5}
180 \end{tabular}
181 \end{center}
182 \end{diff}
```

Le reste du tableau ne pose pas de problème particulier, à par de penser à utiliser toujours `\cline` pour ne pas couper la première colonne.

```
184 \end{document}
```

C'est tout pour cette fois.