Aide-Mémoire Matrices

► On peut définir directement des matrices d'origine dans LaTeXpar les commandes d'environnement **\left (\begin{array}{cc} .. \end{array}\right)** mais c'est assez fastidieux. Les environnements prédéfinis du package ***amsmath*** permettent d'être plus efficace. On dispose des environnements :
\begin{matrix} \begin{pmatrix} \begin{vmatrix} \begin{bmatrix} \begin{Vmatrix}

$$\begin{matrix}1 & -1\\ 0 & 2\end{matrix} \qquad \begin{pmatrix}1 & -1\\ 0 & 2\end{pmatrix} \qquad \begin{vmatrix}1 & -1\\ 0 & 2\end{vmatrix} \qquad \begin{bmatrix}1 & -1\\ 0 & 2\end{bmatrix} \qquad \begin{Vmatrix}1 & -1\\ 0 & 2\end{Vmatrix}$$

► Pour obtenir des matrices *de plus petite taille*, plus condensées, voire les utiliser en mode « *inline* », on dispose de l'environnement **\begin{smallmatrix}..\end{smallmatrix}** du package ***amsmath***.
On peut aussi utiliser la commande **\scalebox[coeff]{..}** :

$$\begin{pmatrix}1 & a\\ a & 2\end{pmatrix} \qquad \left(\begin{smallmatrix}1 & a\\ a & 2\end{smallmatrix}\right) \qquad \scalebox{0.8}{$\begin{pmatrix}1 & a\\ a & 2\end{pmatrix}$}$$

```
\left (\begin{smallmatrix}
    1&a\\ a&2
\end{smallmatrix}\right)
\scalebox{0.8}
    {$\begin{pmatrix} 1&a\\ a&2 \end{pmatrix}$}
```

► *L'écartement entre les lignes* est géré par la *longueur* qui gère l'interligne des arrays dans LaTeX, cad **arraystretch**. On peut la redéfinir par **\renewcommand{\arraystretch}{coeff}**. L'inconvénient de cette commande est qu'elle agit globalement. Il faut corriger son effet après en la réappelant avec le coefficient 1. Pour agir localement, on dispose de la commande **\setstretch{coeff}** du package ***setspace***. (A noter que cette commande engendre un warning lors de la compilation, tout au moins dans ma distribution MikTeX 2.9)

$$\begin{pmatrix}2 & 3\\ 1 & 4\end{pmatrix} \qquad \begin{pmatrix}2 & 3\\ 1 & 4\end{pmatrix} \qquad \renewcommand{\arraystretch}{0.6}\begin{pmatrix}2 & 3\\ 1 & 4\end{pmatrix}$$

```
\renewcommand{\arraystretch}{0.6}
\begin{pmatrix} 2&3 \\ 1&4 \end{pmatrix}
\renewcommand{\arraystretch}{1}
\setstretch{2}
\begin{pmatrix} 2&3 \\ 1&4 \end{pmatrix}
```

► *Positionnement vertical des colonnes* : par défaut, les environnements précédents centrent les colonnes des matrices. ceci peut produire des effets pas toujours agréables à l'œil. On peut via les commandes LaTeX redéfinir un tableau ou alors utiliser les environnements **\begin{matrix∗} \begin{pmatrix∗} \begin{smallmatrix∗}**...du package ***mathtools*** qui permettent de choisir l'alignement interne aux colonnes. La commande est **\begin{pmatrix∗}[type]** où type est n'importe quel type de positionnement valide dans l'environnement array comme **r l c**.

$$\begin{pmatrix}2 & -3\\ -1 & 4\end{pmatrix} \qquad \begin{pmatrix*}[r]2 & -3\\ -1 & 4\end{pmatrix*}$$

```
\begin{pmatrix} 2 & −3 \\ −1 & 4 \end{pmatrix}
\begin{pmatrix*}[r] 2 & −3 \\ −1 & 4 \end{pmatrix*}
```

► *L'écartement des colonnes* est géré automatiquement par LaTeX qui adapte la largeur au contenu. On peut néanmoins combiner un **\phantom** avec un **\\[negatif]** pour donner sa propre taille.

$$\begin{pmatrix} 1&1&1&1&1&1 \\ 2&2&2&2&2&2 \\ 3&3&3&3&3&3 \\ 4&4&4&4&4&4 \\ 5&5&5&5&5&5 \\ 6&6&6&6&6&6 \\[-0.7cm] \aaa&\aaa&\aaa&\aaa&\aaa&\aaa \end{pmatrix}$$

```
\newcommand{\aaa}
   {\hphantom{\hspace{1cm}}}
\setstretch{2}
\begin{pmatrix}
   1&1&1&1&1&1 \\
   2&2&2&2&2&2 \\
   3&3&3&3&3&3 \\
   4&4&4&4&4&4 \\
   5&5&5&5&5&5 \\
   6&6&6&6&6&6 \\[-0.7cm]
   \aaa&\aaa&\aaa&\aaa&\aaa&\aaa
\end{pmatrix}
```

► Les matrices de taille $(n \times n)$ nécessitent l'usage de pointillés. Dans LaTeX sont prédéfinis les commandes **\dots \vdots \ddots** mais il manque la diagonale inverse. On peut la trouver dans de divers packages mathématiques. Sinon, on peut tout simplement se créer la macro **\newcommand{\revdots}{\reflectbox{$\ddots$}}**. A noter aussi, lu sur le web, la commande inverse basée sur la formule de **\ddots** donnée p.359 dans « *The TeXbook* » de D. Knuth.

```
\newcommand{\revdots}{\mathinner
   {\mkern1mu\raise1pt\vbox{\kern7pt\hbox{.}}
   \mkern2mu\raise4pt\hbox{.}
   \mkern2mu\raise7pt\hbox{.}\mkern1mu}
   }
```

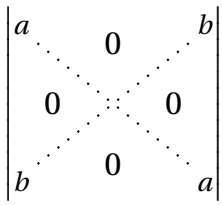
Je m'en suis d'ailleurs inspiré aussi, pour réaliser un **\croixdots** qui m'a servi juste pour dessiner le classique déterminant « *en croix* » donné en calcul à mes élèves (Il y a bien sûr le problème de la parité…) :

```
\newcommand{\croixdots}{%
   \mathinner{
      \mkern1mu\raise7pt\vbox{\kern7pt % Utile ?
      \hbox{.}}
      \mkern-5mu % fait de visu....
      \raise1pt\vbox{\hbox{.}}
      \mkern2mu\raise4pt\hbox{.}
      \mkern2mu\raise1pt\hbox{.}
      \mkern-5mu % fait de visu....
      \raise7pt\vbox{\hbox{.}}
      \mkern1mu
   }%
}
```

$$\begin{vmatrix} a & 0 & \dots & \dots & \dots & 0 & b \\ 0 & a & \ddots & & \iddots & b & 0 \\ \vdots & \ddots & \ddots & & \iddots & \iddots & \vdots \\ \vdots & & & \ddots & & & \vdots \\ \vdots & \iddots & \iddots & & \ddots & \ddots & \vdots \\ 0 & b & \iddots & & \ddots & a & 0 \\ b & 0 & \dots & \dots & \dots & 0 & a \end{vmatrix}$$

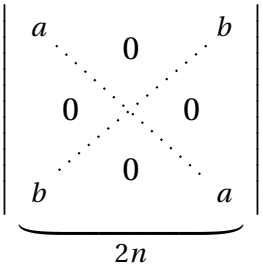
```
\begin{vmatrix}
  a & 0 & \dots & \dots  & \dots & 0 & b \\
  0 & a & \ddots &        & \iddots & b & 0 \\
  \vdots & \ddots & \ddots &  & \iddots & \iddots & \vdots \\
  \vdots &  &  & \ddots &  &  & \vdots \\
  \vdots & \iddots & \iddots &  & \ddots & \ddots & \vdots \\
  0 & b & \iddots &  & \ddots & a & 0 \\
  b & 0 & \dots & \dots & \dots & 0 & a
\end{vmatrix}
```

On peut dessiner une matrice nettement plus jolie (et plus condensée) avec Ti*kZ* : Il y a deux méthodes, celle « *à l'ancienne* », où on tatonne pour les pointillés :




```
\node(A)at(0,0){$ \begin{vmatrix}
        a&\hphantom{\hspace{1.5cm}}&b
        \\[1.5cm]
        b&&a
    \end{vmatrix}$};
\draw[loosely dotted,thick](-1,0.8)--(1,-0.8);
\draw[loosely dotted,thick](1,0.8)--(-1,-0.8);
\foreach \i/\j  in {0/.8,0/-.8,.8/0,-.8/0}
    \node[scale=1.2] at (\i,\j){$0$};
```

La deuxième méthode, beaucoup plus efficace et dans le style de Ti*kZ* , est d'utiliser en préambule le chargement du package ***usetikzlibrary{matrix}*** puis la commande **\matrix[matrix of nodes]**, voire même **\matrix[matrix of math nodes]** qui permet l'utilisation directe du mode maths dans les cellules (nodes) de la matrice. Cette commande permet de joindre les nœuds de façon méthodique et évite le tâtonnement (comme plus haut) dans les pointillés. On donne un nom à la matrice, dans l'exemple c'est *M*, puis on repère chaque noeud (cellule/élément) par **M−i−j**. Il suffit ensuite d'utiliser les traditionnels « *anchors* » des nodes qui sont **north south west north east**...Dernière option, le choix des « *délimiteurs* », options **left delimiter right delimiter** et même **below delimiter above delimiter** :



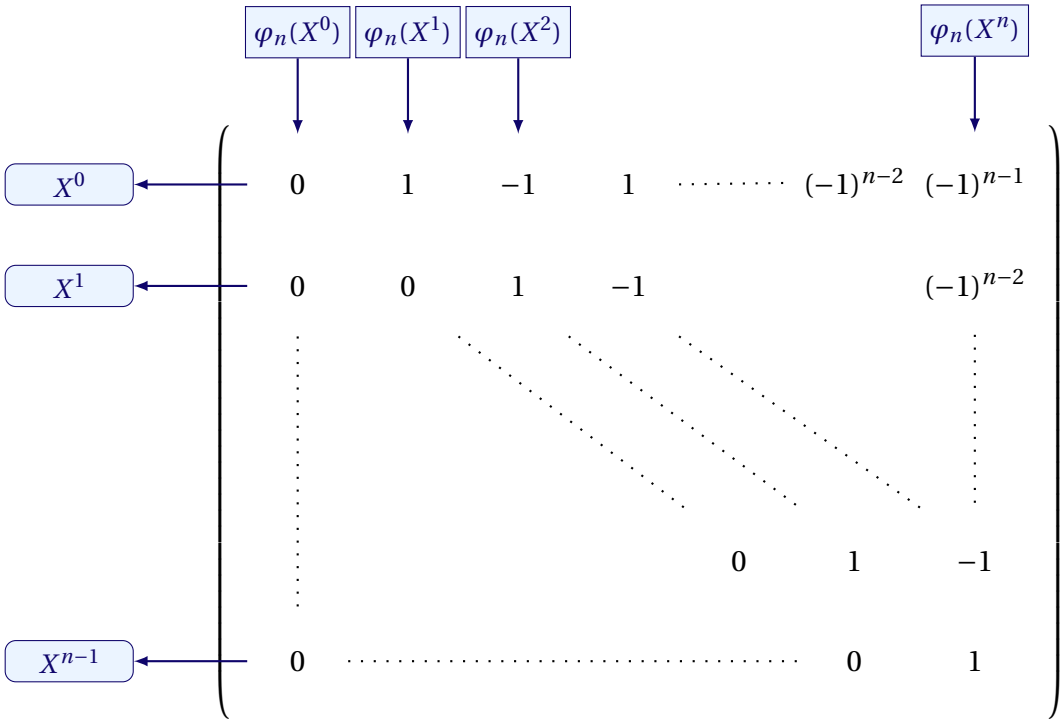

```
\matrix(M) [matrix of math nodes, left delimiter=|, right delimiter=|,
column sep=2cm, row sep=1.7cm, below delimiter=) ]
{   a & b\\
    b & a\\  };
\node[below=2mm] at (M.south) {$2n$};
\draw[loosely dotted,thick](M-1-1.south east)--(M-2-2.north west);
\draw[loosely dotted,thick](M-1-2.south west)--(M-2-1.north east);
\foreach \pos in {(0,.8),(0,-.8),(.8,0),(-.8,0)}
    \node[scale=1.2] at \pos {$0$};
```

► A noter la commande **\bordermatrix** pour « *border* » une matrice à gauche. Il y a aussi **\bordermatrix*{..}** pour border à droite et pour utiliser d'autres « *parenthèses* » comme [.] ou {.} on utilise la syntaxe **\bordermatrix[{ [ ] }]{ ... }** et **\bordermatrix[\{ \}]{ ... }**

$$\begin{array}{c} \\ e_1 \\ e_2 \end{array} \begin{array}{c} \begin{array}{cc} f(e_1) & f(e_2) \end{array} \\ \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \end{array}$$

```
\bordermatrix{& f(e_1) & f(e_2) \cr e_1 & 2 & 0 \cr e_2 & 0 & 3}
```

Là-encore, on peut réaliser des choses plus jolies avec Ti*kZ* , et cela prend peu de temps, juste le temps de comprendre comment fonctionne le positionnement et les styles :




```
\newcommand{\aaa}{\hphantom{\hspace{1.1cm}}}
\matrix(M) [matrix of math nodes, left delimiter=(, right delimiter=),
 nodes={minimum size=1.2cm,scale=1.1}] %row sep=0.1cm,
{   0 & 1 & −1 & 1 &    & (−1)^{n−2} & (−1)^{n−1} \\
    0 & 0 & 1  & −1 &   &            & (−1)^{n−2} \\
    \aaa&\aaa&\aaa&\aaa&\aaa&\aaa&\aaa \\[1cm]
      &   &   &   & 0 &   1        &   & −1      & \\
    0 &   &   &   &   &   0        &   & 1       & \\
};
\tikzset{fleche/.style={thick,bleunuit,>=latex}};
\tikzset{image/.style={above,draw,thin,minimum height=2em,fill=bleuciel!15}};
\tikzset{image2/.style={left,draw,thin,minimum width=5em,fill=bleuciel!15,rounded corners}};
\tikzset{pointille/.style={loosely dotted,thick}};
\foreach \i/\j in {1/0,2/1,5/n−1}
    \draw[fleche,−>](M−\i−1.west)−−++(−1.5,0) node[image2]{$X^{\j}$};
%\draw[fleche2](M−1−1.north)−−++(0,1) node[image] {$\phi_n(1)$} ;
\foreach \i/\j in {1/0,2/1,3/2,7/n}
    \draw[fleche,<−](M−1−\i.north)−−++(0,1) node[image] {$\phi_n(X^{\j})$} ;
\draw[pointille] (M−2−2.south east)−−(M−4−5.north west);
\draw[pointille] (M−2−3.south east)−−(M−4−6.north west);
\draw[pointille] (M−2−4.south east)−−(M−4−7.north west);
\draw[pointille] (M−2−1.south)−−(M−5−1.north);
\draw[pointille] (M−5−1.east)−−(M−5−6.west);
\draw[pointille] (M−2−7.south)−−(M−4−7.north);
\draw[pointille] (M−1−4.east)−−(M−1−6.west);
```

► Pour écrire des *matrices par blocs*, deux exemples :

$$\left(\begin{array}{c|c|c} I_n & J & 0 \\ \hline 0 & A & 0 \\ \hline 0 & 0 & I_n \end{array}\right)$$

```
\left (\begin{array}{c|c|c}
    I_n & J & 0 \\
    \hline
    0 & A & 0 \\
    \hline
    0 & 0 & I_n
\end{array}\right)
```

$$\left(\begin{array}{c@{}c} \begin{array}{|cc|}\hline 1 & 2 \\ -1 & 0 \\ \hline \end{array} & 0 \\ 0 & \begin{array}{|ccc|}\hline 1 & 2 & -2 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \\ \hline \end{array} \end{array}\right)$$

```
\left (\begin{array}{c@{}c}
    \begin{array}{|cc|}
        \hline
        1 & 2 \\ -1& 0 \\
        \hline
    \end{array}
    & 0 \\
    0 & \begin{array}{|ccc|}
        \hline
         1 & 2 & -2 \\
         0 & -1 & 1 \\
         1 & 0 & -2\\
        \hline
    \end{array}
\end{array}\right)
```

La même matrice avec Ti*k*Z , et c'est pas vraiment plus long à taper :

$$\left(\begin{array}{cc} \begin{array}{|cc|}\hline 1 & 2 \\ -1 & 0 \\ \hline \end{array} & 0 \\ 0 & \begin{array}{|ccc|}\hline 1 & 2 & -2 \\ 0 & -1 & 1 \\ 1 & 0 & -2 \\ \hline \end{array} \end{array}\right)$$

```
\tikzset{bloc/.style={fill =bleuciel,rounded corners,fill opacity=0.2}};
\matrix(M) [matrix of math nodes, left delimiter=(,
right  delimiter=), column sep=2mm, row sep=2mm]
    {   1 & 2   &&& \\
       -1 & 0   &&& \\
        && 1 & 2 & -2 \\
        && 0 & -1 & 1 \\
        && 1 & 0 & -2\\ };
\draw [bloc] (M-1-1.north west) rectangle (M-2-2.south east);
\draw [bloc] (M-3-3.north west) rectangle (M-5-5.south east);
\node[scale=1.2,above=5mm] at (M-3-4.north) {$0$};
\node[scale=1.2,left=1cm] at (M-4-4.west){$0$};
```

20 novembre 2011. Cet aide-mémoire a été rédigé par Benoit Caritey. Pour tout commentaire ou amélioration, vous pouvez lui envoyer un mail à l'adresse ✉ *bcaritey@free.fr*