

L'extension `nicematrix`*

F. Pantigny
fpantigny@wanadoo.fr

18 juillet 2019

Résumé

L'extension LaTeX `nicematrix` fournit de nouveaux environnements similaires aux environnements classiques `{array}` et `{matrix}` mais avec des fonctionnalités supplémentaires. Parmi ces fonctionnalités figurent la possibilité de fixer la largeur des colonnes et de tracer des traits en pointillés continus entre les cases du tableau.

1 Présentation

Cette extension peut être utilisée avec `xelatex`, `lualatex` et `pdflatex` mais aussi avec le cheminement classique `latex-dvips-ps2pdf` (ou Adobe Distiller). Deux ou trois compilations successives peuvent être nécessaires. Cette extension nécessite et charge les extensions `expl3`, `l3keys2e`, `xparse`, `array`, `amsmath` et `tikz`. Elle charge aussi la bibliothèque Tikz fit.

Cette extension fournit quelques outils supplémentaires pour dessiner des matrices (au sens mathématique). Les principales caractéristiques sont les suivantes :

- lignes en pointillés continues¹ ;
- une première rangée et une dernière colonne pour les labels ;
- un contrôle sur la largeur des colonnes.

$$\begin{array}{c} C_1 \quad C_2 \dots \dots C_n \\ \left[\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right] \begin{array}{c} L_1 \\ L_2 \\ \vdots \\ L_n \end{array} \end{array}$$

Une commande `\NiceMatrixOptions` est fournie pour régler les options (la portée des options fixées par cette commande est le groupe TeX courant).

Un exemple d'utilisation pour les lignes en pointillés continus

Considérons par exemple le code suivant qui utilise un environnement `{pmatrix}` de l'extension `amsmath`.

```
$A = \begin{pmatrix}
1 & & \cdots & & 1 & \\
0 & & \ddots & & & \vdots \\
\vdots & & \ddots & & \ddots & \vdots \\
0 & & \cdots & & 0 & 1
\end{pmatrix}$
```

$$A = \begin{pmatrix} 1 & \dots & \dots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

Ce code compose la matrice A représentée à droite.

Maintenant, si nous utilisons l'extension `nicematrix` avec l'option `transparent`, le même code va donner le résultat ci-contre à droite.

$$A = \begin{pmatrix} 1 & \dots & \dots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

*Ce document correspond à la version 2.3 of `nicematrix`, en date du 2019/07/18.

1. Si l'option de classe `draft` est utilisée, ces lignes en pointillés ne sont pas tracées pour accélérer la compilation.

2 Les environnements de cette extension

L'extension `nicematrix` définit les nouveaux environnements suivants :

<code>\NiceMatrix</code>	<code>\NiceArray</code>	<code>\pNiceArrayC</code>	<code>\pNiceArrayRC</code>
<code>\pNiceMatrix</code>		<code>\bNiceArrayC</code>	<code>\bNiceArrayRC</code>
<code>\bNiceMatrix</code>		<code>\BNiceArrayC</code>	<code>\BNiceArrayRC</code>
<code>\BNiceMatrix</code>		<code>\vNiceArrayC</code>	<code>\vNiceArrayRC</code>
<code>\vNiceMatrix</code>		<code>\VNiceArrayC</code>	<code>\VNiceArrayRC</code>
<code>\VNiceMatrix</code>		<code>\NiceArrayCwithDelims</code>	<code>\NiceArrayRCwithDelims</code>

Par défaut, les environnements `\NiceMatrix`, `\pNiceMatrix`, `\bNiceMatrix`, `\BNiceMatrix`, `\vNiceMatrix` et `\VNiceMatrix` se comportent quasiment comme les environnements correspondants de `amsmath` : `\matrix`, `\pmatrix`, `\bmatrix`, `\Bmatrix`, `\vmatrix` et `\Vmatrix`.

L'environnement `\NiceArray` est similaire à l'environnement `\array` de l'extension `\array`. Néanmoins, pour des raisons techniques, dans le préambule de l'environnement `\NiceArray`, l'utilisateur doit utiliser les lettres `L`, `C` et `R` au lieu de `l`, `c` et `r`. Il est possible d'utiliser les constructions `w{...}{...}`, `W{...}{...}`, `l`, `>{...}`, `<{...}`, `@{...}`, `!{...}` et `*{n}{...}` mais les lettres `p`, `m` et `b` ne doivent pas être employées. Voir p. 7 la partie concernant `\NiceArray`.

Les environnements avec un `C` à la fin de leur nom, `\pNiceArrayC`, `\bNiceArrayC`, `\BNiceArrayC`, `\vNiceArrayC` et `\VNiceArrayC` sont similaires aux environnements `\NiceArray` (en particulier les lettres spéciales `L`, `C` et `R`) mais créent une colonne extérieure (à droite du délimiteur). Voir p. 8 la partie concernant `\pNiceArrayC`.

Les environnements dont le nom se termine par `RC` (`\pNiceArrayRC`, `\bNiceArrayRC`, `\BNiceArrayRC`, `\vNiceArrayRC` et `\VNiceArrayRC`) sont similaires à l'environnement `\NiceArray` mais créent une rangée extérieure (au-dessus de la matrice principale) et une colonne extérieure. Voir p. 9 la partie concernant `\pNiceArrayRC`.

3 Les lignes en pointillés continues

À l'intérieur des environnements de l'extension `nicematrix`, de nouvelles commandes sont définies : `\Ldots`, `\Cdots`, `\Vdots`, `\Ddots`, and `\Iddots`. Ces commandes sont conçues pour être utilisées à la place de `\dots`, `\cdots`, `\vdots`, `\ddots` et `\iddots`.²

Chacune de ces commandes doit être utilisée seule dans la case du tableau et elle trace une ligne en pointillés entre les premières cases non vides³ situées de part et d'autre de la case courante. Bien entendu, pour `\Ldots` et `\Cdots`, c'est une ligne horizontale ; pour `\Vdots`, c'est une ligne verticale et pour `\Ddots` et `\Iddots`, ce sont des lignes diagonales.

```
\begin{bNiceMatrix}
a_1      & \Cdots &      & & a_1 \\
\Vdots   & a_2      & \Cdots & & a_2 \\
        & \Vdots & \Ddots & \\
\\
a_1      & a_2      &      & & a_n \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} a_1 & \cdots & \cdots & \cdots & a_1 \\ \vdots & & & & \vdots \\ & a_2 & \cdots & \cdots & a_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_1 & a_2 & & & a_n \end{bmatrix}$$

Pour représenter la matrice nulle, on peut choisir d'utiliser le codage suivant :

```
\begin{bNiceMatrix}
0      & \Cdots & 0 \\
\Vdots & & \Vdots \\
0      & \Cdots & 0 \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

2. La commande `\iddots`, définie dans `nicematrix`, est une variante de `\ddots` avec les points allant vers le haut : $\cdot^{\cdot^{\cdot}}$. Si `mathdots` est chargée, la version de `mathdots` est utilisée. Elle correspond à la commande `\adots` de `unicode-math`.
3. La définition précise de ce qui est considéré comme une «case vide» est donnée plus loin (cf. p. 12).

On peut néanmoins souhaiter une matrice plus grande. Habituellement, dans un tel cas, les utilisateurs de LaTeX ajoutent une nouvelle ligne et une nouvelle colonne. Il est possible d'utiliser la même méthode avec `nicematrix` :

```
\begin{bNiceMatrix}
0      & \Cdots & \Cdots & 0      & \\
\Vdots &         &         & \Vdots & \\
\Vdots &         &         & \Vdots & \\
0      & \Cdots & \Cdots & 0      & \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

Dans la première colonne de cet exemple, il y a deux instructions `\Vdots` mais une seule ligne en pointillés sera tracée (il n'y a pas d'objets qui se superposent dans le fichier PDF résultant⁴). Néanmoins, des calculs inutiles sont effectués par TeX avant qu'il ne détecte que les deux instructions définissent la même ligne en pointillés. C'est pourquoi l'extension `nicematrix` fournit des versions étoilées de `\Ldots`, `\Cdots`, etc. : `\Ldots*`, `\Cdots*`, etc. Ces versions sont simplement équivalentes à `\hphantom{\ldots}`, `\hphantom{\cdots}`, etc. L'utilisateur devrait utiliser des versions étoilées à chaque fois qu'une version classique (sans étoile) a déjà été utilisée pour la même ligne en pointillés.

```
\begin{bNiceMatrix}
0      & \Cdots & \Cdots* & 0      & \\
\Vdots &         &         & \Vdots & \\
\Vdots* &         &         & \Vdots* & \\
0      & \Cdots & \Cdots* & 0      & \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

En fait, dans cet exemple, il aurait été possible de tracer la même matrice sans commandes étoilées avec le codage suivant :

```
\begin{bNiceMatrix}
0      & \Cdots &         & 0      & \\
\Vdots &         &         & \Vdots & \\
        &         &         & \Vdots & \\
0      &         & \Cdots & 0      & \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & & & \vdots \\ & & & \vdots \\ 0 & & \cdots & 0 \end{bmatrix}$$

Il y a aussi d'autres moyens de changer la taille d'une matrice. On pourrait vouloir utiliser l'argument optionnel de la commande `\` pour l'espacement vertical et la commande `\hspace*` dans une case pour l'espacement horizontal.⁵

Toutefois, une commande `\hspace*` pourrait interférer dans la construction des lignes en pointillés. C'est pourquoi l'extension `nicematrix` fournit une commande `\Hspace` qui est une variante de `\hspace` transparente pour la construction des lignes en pointillés de `nicematrix`.

```
\begin{bNiceMatrix}
0      & \Cdots & \Hspace*{1cm} & 0      & \\
\Vdots &         &         & \Vdots & \\
0      & \Cdots &         & 0      & \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & 0 \end{bmatrix}$$

4. Et il n'est pas possible de tracer une ligne `\Ldots` et une ligne `\Cdots` entre les mêmes cases.

5. Le meilleur moyen de régler la largeur des colonnes est, de toutes manières, d'utiliser l'environnement `{NiceArray}` avec une colonne de type `w` ou `W`.

3.1 L'option nullify-dots

Considérons la matrice suivante qui a été composée classiquement avec l'environnement `{pmatrix}` de `amsmath`.

```
$A = \begin{pmatrix}
a_0 & b \\
a_1 & \\
a_2 & \\
a_3 & \\
a_4 & \\
a_5 & b
\end{pmatrix}
```

$$A = \begin{pmatrix} a_0 & b \\ a_1 & \\ a_2 & \\ a_3 & \\ a_4 & \\ a_5 & b \end{pmatrix}$$

Si nous ajoutons des instructions `\vdots` dans la seconde colonne, la géométrie de la matrice est modifiée.

```
$B = \begin{pmatrix}
a_0 & b \\
a_1 & \vdots \\
a_2 & \vdots \\
a_3 & \vdots \\
a_4 & \vdots \\
a_5 & b
\end{pmatrix}
```

$$B = \begin{pmatrix} a_0 & b \\ a_1 & \vdots \\ a_2 & \vdots \\ a_3 & \vdots \\ a_4 & \vdots \\ a_5 & b \end{pmatrix}$$

Par défaut, avec `nicematrix`, si nous remplaçons `{pmatrix}` par `{pNiceMatrix}` et `\vdots` par `\Vdots` (or `\Vdots*` pour la rapidité de compilation), la géométrie de la matrice n'est pas changée.

```
$C = \begin{pNiceMatrix}
a_0 & b \\
a_1 & \Vdots \\
a_2 & \Vdots* \\
a_3 & \Vdots* \\
a_4 & \Vdots* \\
a_5 & b
\end{pNiceMatrix}
```

$$C = \begin{pmatrix} a_0 & b \\ a_1 & \vdots \\ a_2 & \vdots \\ a_3 & \vdots \\ a_4 & \vdots \\ a_5 & b \end{pmatrix}$$

On pourrait toutefois préférer la géométrie de la première matrice A et vouloir avoir la même géométrie avec une ligne en pointillés continue dans la seconde colonne. C'est possible en utilisant l'option `nullify-dots` (et une seule instruction `\Vdots` suffit).

```
$D = \begin{pNiceMatrix}[nullify-dots]
a_0 & b \\
a_1 & \Vdots \\
a_2 & \\
a_3 & \\
a_4 & \\
a_5 & b
\end{pNiceMatrix}
```

$$D = \begin{pmatrix} a_0 & b \\ a_1 & \vdots \\ a_2 & \vdots \\ a_3 & \vdots \\ a_4 & \vdots \\ a_5 & b \end{pmatrix}$$

L'option `nullify-dots` «smashe» les instructions `\Ldots` (et ses variantes) verticalement mais aussi horizontalement.

Il doit n'y avoir aucun espace devant le crochet ouvrant (`[`) des options de l'environnement.

3.2 La commande `\Hdotsfor`

Certaines personnes utilisent habituellement la commande `\hdotsfor` de l'extension `amsmath` pour tracer des lignes en pointillés horizontales dans une matrice. Dans les environnements de `nicematrix`,

il convient d'utiliser `\Hdotsfor` à la place pour avoir les lignes en pointillés similaires à toutes celles tracées par l'extension `nicematrix`.

Comme avec les autres commandes de `nicematrix` (comme `\Cdots`, `\Ldots`, `\Vdots`, etc.), la ligne en pointillés tracée par `\Hdotsfor` s'étend jusqu'au contenu des cases de part et d'autre.

```


$$\begin{pNiceMatrix}
1 & 2 & 3 & 4 & 5 \\
1 & \Hdotsfor{3} & & & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
\end{pNiceMatrix}$$


```

Néanmoins, si ces cases sont vides, la ligne en pointillés s'étend seulement dans les cases spécifiées par l'argument de `\Hdotsfor` (par conception).

```


$$\begin{pNiceMatrix}
1 & 2 & 3 & 4 & 5 \\
& \Hdotsfor{3} & & & \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
\end{pNiceMatrix}$$


```

La commande `\hdotsfor` de `amsmath` prend un argument optionnel (entre crochets) qui est utilisé pour un réglage fin de l'espace entre deux points consécutifs. Par homogénéité, `\Hdotsfor` prend aussi un argument optionnel mais cet argument est écarté silencieusement.

Remarque : Contrairement à la commande `\hdotsfor` de `amsmath`, la commande `\Hdotsfor` est compatible avec l'extension `colortbl`.

3.3 Comment créer les lignes en pointillés de manière transparente

L'extension `nicematrix` fournit une option appelée `transparent` qui permet d'utiliser du code existant de manière transparente dans les environnements `{matrix}`. Cette option peut être passée en option du `\usepackage` ou bien être fixée avec la commande `\NiceMatrixOptions`.

En fait, cette option est un alias pour la conjonction de deux options : `renew-dots` et `renew-matrix`.

— L'option `renew-dots`

Avec cette option, les commandes `\ldots`, `\cdots`, `\vdots`, `\ddots`, `\iddots`⁶ et `\hdotsfor` sont redéfinies dans les environnements de `nicematrix` et agissent comme `\Ldots`, `\Cdots`, `\Vdots`, `\Ddots`, `\Iddots` et `\Hdotsfor` ; la commande `\dots` (points de suspension «automatiques» de `amsmath`) est aussi redéfinie et se comporte comme `\Ldots`.

— L'option `renew-matrix`

Avec cette option, l'environnement `{matrix}` est redéfini et se comporte comme `{NiceMatrix}` et il en est de même pour les cinq variantes.

Par conséquent, avec l'option `transparent`, un code classique donne directement le résultat fourni par `nicematrix`.

```


$$\begin{matrix}
1 & & \cdots & & \cdots & & 1 \\
0 & & \ddots & & & & \vdots \\
\vdots & & \ddots & & \ddots & & \vdots \\
0 & & \cdots & & 0 & & 1
\end{matrix}$$


```

6. La commande `\iddots` n'est pas une commande de LaTeX mais est définie dans l'extension `nicematrix`. Si `mathdots` est chargée, la version de `mathdots` est utilisée.

4 Les nœuds Tikz créés par l'extension nicematrix

L'extension `nicematrix` crée un nœud Tikz pour chaque case du tableau considéré. Ces nœuds sont utilisés pour tracer les lignes en pointillés entre les cases du tableau. Toutefois, l'utilisateur peut utiliser directement ces nœuds. On commence par donner un nom au tableau (avec l'option `name`). Cela étant fait, les nœuds sont accessibles à travers les noms «*nom-i-j*» où *nom* est le nom donné au tableau et *i* et *j* les numéros de ligne et de colonne de la case considérée.

```
\begin{pNiceMatrix}[name=ma-matrice]
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{pNiceMatrix}
\tikz[remember picture,overlay]
\draw (ma-matrice-2-2) circle (2mm) ;
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & \textcircled{5} & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Ne pas oublier les options `remember picture` et `overlay`.

Dans l'exemple suivant, nous avons surligné toutes les cases de la matrice.

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix}$$

En fait, l'extension `nicematrix` peut créer des nœuds supplémentaires (*extra nodes* en anglais). Ces nouveaux nœuds sont créés si l'option `create-extra-nodes` est utilisée. Il y a deux séries de nœuds supplémentaires : les «nœuds moyens» (*medium nodes* en anglais) et les «nœuds larges» (*large nodes* en anglais).

Les noms des «nœuds moyens» s'obtiennent en ajoutant le suffixe «`-medium`» au nom des nœuds normaux. Dans l'exemple suivant, on a surligné tous les «nœuds moyens». Nous considérons que cet exemple se suffit à lui-même comme définition de ces nœuds.

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix}$$

Les noms des «nœuds larges» s'obtiennent en ajoutant le suffixe «`-large`» au nom des nœuds normaux. Dans l'exemple suivant, on a surligné tous les «nœuds larges». Nous considérons que cet exemple se suffit à lui-même comme définition de ces nœuds.⁷

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix}$$

Les «nœuds larges» de la première colonne et de la dernière colonne peuvent apparaître trop petits pour certains usages. C'est pourquoi il est possible d'utiliser les options `left-margin` et `right-margin` pour ajouter de l'espace des deux côtés du tableau et aussi de l'espace dans les «nœuds larges» de la première colonne et de la dernière colonne. Dans l'exemple suivant, nous avons utilisé les options `left-margin` et `right-margin`.⁸

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix}$$

7. Dans les environnements du type de `{pNiceArrayC}` et `{pNiceArrayRC}`, il n'y a pas de «nœuds larges» créés dans la rangée supplémentaire et la colonne supplémentaire.

8. Les options `left-margin` et `right-margin` prennent des dimensions comme valeurs mais, si aucune valeur n'est donnée, c'est la valeur par défaut qui est utilisée et elle est égale à `\arraycolsep`.

Il est aussi possible d'ajouter de l'espace des deux côtés du tableau avec les options `extra-left-margin` et `extra-right-margin`. Ces marges ne sont pas incorporées dans les «nœuds larges». Dans l'exemple suivant, nous avons utilisé `extra-left-margin` et `extra-right-margin` avec la valeur 3 pt.

$$\left(\begin{array}{|c|c|c|} \hline a & a+b & a+b+c \\ \hline a & a & a+b \\ \hline a & a & a \\ \hline \end{array} \right)$$

Dans le cas présent, si on veut un contrôle sur la hauteur des rangées, on peut ajouter un `\strut` dans chaque rangée du tableau.

$$\left(\begin{array}{|c|c|c|} \hline a & a+b & a+b+c \\ \hline a & a & a+b \\ \hline a & a & a \\ \hline \end{array} \right)$$

On explique plus loin comment surligner les nœuds créés par Tikz (cf. p. 16).

5 Le code-after

L'option `code-after` peut être utilisée pour indiquer du code qui sera exécuté après la construction de la matrice (et donc, en particulier, après la construction de tous les nœuds).

Dans le `code-after`, les nœuds Tikz doivent être désignés sous la forme $i-j$ (sans le préfixe correspondant au nom de l'environnement).

De plus, une commande spéciale, nommée `\line` est disponible pour tracer directement des lignes en pointillés entre les nœuds).

```
$\begin{pNiceMatrix}[code-after = {\line {1-1} {3-3}}]
0 & 0 & 0 \\
0 & & 0 \\
0 & 0 & 0 \\
\end{pNiceMatrix}$
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

6 L'environnement {NiceArray}

L'environnement `{NiceArray}` est similaire à l'environnement `{array}`. Comme pour `{array}`, l'argument obligatoire est le préambule du tableau. Néanmoins, pour des raisons techniques, l'utilisateur doit utiliser les lettres `L`, `C` et `R`⁹ au lieu de `l`, `c` et `r`.

Il est possible d'utiliser les constructions `w{...}{...}`, `W{...}{...}`, `|`, `>{...}`, `<{...}`, `@{...}`, `!{...}` et `*{n}{...}` mais les lettres `p`, `m` et `b` ne doivent pas être employées.¹⁰

L'environnement `{NiceArray}` accepte les options classiques `t`, `c` et `b` de `{array}` mais aussi d'autres options définies par `nicematrix` (`renew-dots`, `columns-width`, etc.).

Un exemple avec un système linéaire (on a besoin de `{NiceArray}` pour le trait vertical) :

```
$\left[\begin{NiceArray}{CCCC|C}
a_1 & ? & & \Cdots & ? & ? \\
0 & & & \Ddots & \Vdots & \Vdots \\
\Vdots & \Ddots & \Ddots & & ? & \\
0 & \Cdots & 0 & & a_n & ? \\
\end{NiceArray}\right]$
```

$$\left[\begin{array}{cccc|c} a_1 & ? & \cdots & ? & ? \\ 0 & & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & & ? \\ 0 & \cdots & 0 & a_n & ? \end{array} \right]$$

9. Les types de colonnes `L`, `C` et `R` sont définis localement à l'intérieur de `{NiceArray}` avec la commande `\newcolumnstype` de `array`. Cette définition masque une éventuelle définition précédente. En fait, les types de colonnes `w` and `W` sont également redéfinis.

10. Dans une commande `\multicolumn`, on doit également utiliser les lettres `L`, `C` et `R`.

Un exemple où on utilise `{NiceArray}` parce que l'on veut utiliser les types de colonne `L` et `R` :

```

 $\left(\begin{array}{lcr} a_{11} & \cdots & a_{1n} \\ a_{21} & & a_{2n} \\ \vdots & & \vdots \\ a_{n-1,1} & \cdots & a_{n-1,n} \end{array}\right)$ 

```

7 L'environnement `{pNiceArrayC}` et ses variantes

L'environnement `{pNiceArrayC}` compose une matrice avec une colonne extérieure.

L'environnement `{pNiceArrayC}` prend un argument obligatoire qui est le préambule du tableau. Les types de colonne disponibles sont les mêmes que pour l'environnement `{NiceArray}`. Néanmoins, aucune spécification de colonne ne doit être donnée pour la dernière colonne. Ce sera automatiquement (et obligatoirement) une colonne de type `L`.

Une option spéciale, nommée `code-for-last-col`, permet d'indiquer des tokens qui seront insérés au début de chaque case de la dernière colonne. L'option `columns-width` ne s'applique pas à la colonne extérieure.

```

 $\begin{pNiceArrayC}{*6C|C}[nullify-dots,code-for-last-col={\scriptstyle}]$ 

$$\begin{array}{cccccc|c} 1 & 1 & 1 & \cdots & 1 & 0 & \\ 0 & 1 & 0 & \cdots & 0 & & L_2 \text{ \scriptstyle gets } L_2-L_1 \\ 0 & 0 & 1 & \ddots & \vdots & & L_3 \text{ \scriptstyle gets } L_3-L_1 \\ & & & \ddots & & \vdots & \\ \vdots & & & \ddots & 0 & & \\ 0 & & & \cdots & 0 & 1 & L_n \text{ \scriptstyle gets } L_n-L_1 \end{array}$$


```

$$\left(\begin{array}{cccccc|c} 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & 1 & 0 & \cdots & 0 & \vdots \\ 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{array} \right) \begin{array}{l} L_2 \leftarrow L_2 - L_1 \\ L_3 \leftarrow L_3 - L_1 \\ \vdots \\ L_n \leftarrow L_n - L_1 \end{array}$$

On remarquera qu'un trait horizontal tracé avec `\hline` ne s'étend *pas* dans la dernière colonne.

```

 $\begin{pNiceArrayC}$ 

$$\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \begin{array}{l} L_1 \\ L_2 \\ L_3 \end{array}$$


$$\hline \begin{array}{ccc} S_1 & S_2 & S_3 \end{array}$$


```

En fait, l'environnement `{pNiceArrayC}` et ses variantes sont fondés sur un environnement plus général, appelé `{NiceArrayCwithDelims}`. Les deux premiers arguments obligatoires de cet environnement sont les délimiteurs gauche et droit qui seront utilisés dans la construction de la matrice. Il est possible d'utiliser `{NiceArrayCwithDelims}` si on a besoin de délimiteurs atypiques.

```

 $\begin{NiceArrayCwithDelims}$ 

$$\begin{array}{ccc|c} 1 & 2 & 3 & L_1 \\ 4 & 5 & 6 & L_2 \\ 7 & 8 & 9 & L_3 \end{array}$$


```


8 L'environnement `{pNiceArrayRC}` et ses variantes

L'environnement `{pNiceArrayRC}` compose une matrice avec une rangée extérieure et une colonne extérieure.

Cet environnement `{pNiceArrayRC}` prend un argument obligatoire qui est le préambule du tableau. Comme pour l'environnement `{pNiceArrayC}`, aucune spécification ne doit être fournie pour la dernière colonne qui sera automatiquement une colonne de type L.

Une option spéciale, appelée `code-for-first-row`, permet d'indiquer des tokens qui seront insérés au début de chaque case de la rangée extérieure.

```
\begin{pNiceArrayRC}{CCC}% (ici, % est obligatoire)
  [columns-width = auto,
   code-for-first-row = \color{blue},
   code-for-last-col  = \color{blue}]
C_1 & C_2 & C_3 \\
1 & 2 & 3 & L_1 \\
4 & 5 & 6 & L_2 \\
7 & 8 & 9 & L_3 \\
\end{pNiceArrayRC}
```

$$\begin{pmatrix} C_1 & C_2 & C_3 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \end{matrix}$$

La première rangée d'un environnement `{pNiceArrayRC}` a le numéro 0, et non 1. Ce numéro est utilisé pour les noms des nœuds Tikz (les noms de ces nœuds sont utilisés, par exemple, par la commande `\line` dans le `code-after`).

Pour des raisons techniques, il n'est pas possible d'utiliser l'option de la commande `\\` après la première rangée (le placement des délimiteurs serait erroné).

Si on veut écrire un système linéaire, on peut utiliser le code suivant, avec un préambule `CCC|C` :

```
\begin{pNiceArrayRC}{CCC|C}
C_1 & \Cdots & C_n \\
a_{11} & \Cdots & a_{1n} & b_1 \\
\Vdots & & \Vdots & \Vdots \\
a_{n1} & \Cdots & a_{nn} & b_n \\
\end{pNiceArrayRC}
```

$$\begin{pmatrix} C_1 & \dots & C_n \\ a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{vmatrix} b_1 \\ \vdots \\ b_n \end{vmatrix}$$

On remarquera que le trait vertical ne s'étend pas dans la première rangée.¹¹

En fait, l'environnement `{pNiceArrayRC}` et ses variantes sont fondés sur un environnement plus général, appelé `{NiceArrayRCwithDelims}`. Les deux premiers arguments obligatoires de cet environnement sont les délimiteurs gauche et droit qui seront utilisés dans la construction de la matrice. Il est possible d'utiliser `{NiceArrayRCwithDelims}` si on a besoin de délimiteurs atypiques.

```
\begin{NiceArrayRCwithDelims}
  {\downarrow}{\downarrow}{CCC}[columns-width=auto]
C_1 & C_2 & C_3 \\
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{NiceArrayRCwithDelims}
```

$$\begin{array}{ccc} C_1 & C_2 & C_3 \\ \left| \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right| \end{array}$$

9 Les lignes en pointillés pour séparer les rangées et les colonnes

Dans les environnements `nicematrix`, il est possible d'utiliser la commande `\hdottedline` (fournie par `nicematrix`) qui est l'équivalent pour les pointillés des commandes `\hline` et `\hdashline` (de `arydshln`).

11. Il s'agit d'une caractéristique apparue dans la version 2.3 de `nicematrix`. Dans les versions antérieures, le trait s'étendait dans la première rangée. On doit remarquer que si `arydshln` est chargé, le trait va se prolonger dans la première rangée (c'est parce que `arydshln` redéfinit de nombreuses macros internes de `array`).

```

\begin{pNiceMatrix}
1 & 2 & 3 & 4 & 5 \\
\hdottedline
6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15
\end{pNiceMatrix}

```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ \hdottedline 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

Dans les environnements avec un préambule explicite (comme `{NiceArray}`, etc.), il est possible de dessiner un trait vertical en pointillés avec le spécificateur «:».

```

\left(\begin{NiceArray}{CCCC:C}
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15
\end{NiceArray}\right)

```

$$\left(\begin{array}{cccc:c} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{array}\right)$$

Ces lignes en pointillés ne s'étendent *pas* dans la première rangée et la dernière colonne des environnements qui proposent ces fonctionnalités (par ex. `{pNiceArrayRC}`).

```

$\begin{pNiceArrayRC}{CCC:C}%
[ code-for-first-row = \color{blue}\scriptstyle,
  code-for-last-col = \color{blue}\scriptstyle ]
C_1 & C_2 & C_3 & C_4 \\
1 & 2 & 3 & 4 & L_1 \\
5 & 6 & 7 & 8 & L_2 \\
9 & 10 & 11 & 12 & L_3 \\
\hdottedline
13 & 14 & 15 & 16 & L_4
\end{pNiceArrayRC}$

```

$$\begin{array}{cccc:c} C_1 & C_2 & C_3 & C_4 & \\ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ \hdottedline 13 & 14 & 15 & 16 \end{pmatrix} & L_1 & L_2 & L_3 & L_4 \end{array}$$

Il est possible de changer dans `nicematrix` la lettre utilisée pour indiquer dans le préambule un trait vertical en pointillés avec l'option `letter-for-dotted-lines` disponible dans `\NiceMatrixOptions`. Par exemple, dans ce document, nous avons chargé l'extension `arydshln` qui utilise la lettre «:» pour indiquer un trait vertical en tiretés. Par conséquent, en utilisant l'option `letter-for-dotted-lines`, on peut utiliser les traits verticaux fournis à la fois par `arydshln` et par `nicematrix`.

```

\NiceMatrixOptions{letter-for-dotted-lines = V}
\left(\begin{NiceArray}{C|C:CVC}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12
\end{NiceArray}\right)

```

$$\left(\begin{array}{c|cc:c} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{array}\right)$$

10 La largeur des colonnes

Dans les environnements avec un préambule explicite (comme `{NiceArray}`, `{pNiceArrayC}`, etc.), il est possible de fixer la largeur d'une colonne avec les lettres classiques `w` et `W` de l'extension `array`.

```

$\left(\begin{NiceArray}{wc{1cm}CC}
1 & 12 & -123 \\
12 & 0 & 0 \\
4 & 1 & 2
\end{NiceArray}\right)$

```

$$\left(\begin{array}{cc} 1 & 12 & -123 \\ 12 & 0 & 0 \\ 4 & 1 & 2 \end{array}\right)$$

Il est aussi possible de fixer la largeur de toutes les colonnes de la matrice directement avec l'option `columns-width` (disponible dans tous les environnements de `nicematrix`).

```

$\begin{pNiceMatrix}[columns-width = 1cm]
1 & 12 & -123 \\
12 & 0 & 0 \\
4 & 1 & 2
\end{pNiceMatrix}$

```

$$\begin{pmatrix} 1 & 12 & -123 \\ 12 & 0 & 0 \\ 4 & 1 & 2 \end{pmatrix}$$

Noter que l'espace inséré entre deux colonnes (égal à `2 \arraycolsep`) n'est pas supprimé (il est évidemment possible de le supprimer en mettant `\arraycolsep` à 0).

Il est possible de donner la valeur `auto` à l'option `columns-width` : toutes les colonnes du tableau auront alors une largeur égale à la largeur de la case la plus large du tableau. **Deux ou trois compilations peuvent être nécessaires.**

```
\begin{pNiceMatrix}[columns-width = auto]
1 & 12 & -123 \\
12 & 0 & 0 \\
4 & 1 & 2
\end{pNiceMatrix}
```

$$\begin{pmatrix} 1 & 12 & -123 \\ 12 & 0 & 0 \\ 4 & 1 & 2 \end{pmatrix}$$

Il est possible de fixer la largeur de toutes les colonnes de toutes les matrices dans une certaine portion de document avec la commande `\NiceMatrixOptions`.

```
\NiceMatrixOptions{columns-width=10mm}
\begin{pNiceMatrix}
a & b \\ c & d \\
\end{pNiceMatrix}
=
\begin{pNiceMatrix}
1 & 1245 \\ 345 & 2 \\
\end{pNiceMatrix}
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 1245 \\ 345 & 2 \end{pmatrix}$$

Mais il est aussi possible de fixer une zone dans laquelle toutes les matrices auront leurs colonnes de la même largeur, égale à la largeur de la case la plus large de toutes les matrices de la zone. Cette construction utilise l'environnement `{NiceMatrixBlock}` avec l'option `auto-columns-width`.¹²

```
\begin{NiceMatrixBlock}[auto-columns-width]
\begin{pNiceMatrix}
a & b \\ c & d \\
\end{pNiceMatrix}
=
\begin{pNiceMatrix}
1 & 1245 \\ 345 & 2 \\
\end{pNiceMatrix}
\end{NiceMatrixBlock}
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 1245 \\ 345 & 2 \end{pmatrix}$$

11 L'option hlines

Il est bien entendu possible d'ajouter des filets horizontaux entre les lignes avec la commande `\hline` dans les environnements de `nicematrix` mais il est aussi possible de demander que tous les filets horizontaux soient tracés avec l'option `hlines`.

```
\begin{NiceArray}{|*{4}{C|}}[hlines]
e & a & b & c \\
a & e & c & b \\
b & c & e & a \\
c & b & a & e
\end{NiceArray}
```

<i>e</i>	<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i>	<i>e</i>	<i>c</i>	<i>b</i>
<i>b</i>	<i>c</i>	<i>e</i>	<i>a</i>
<i>c</i>	<i>b</i>	<i>a</i>	<i>e</i>

12 Utilisation du type de colonne S de siunitx

Si l'extension `siunitx` est chargée (avant ou après `nicematrix`), il est possible d'utiliser les colonnes de type `S` de `siunitx` dans les environnements de `nicematrix`. L'implémentation n'utilise explicitement

¹². Pour le moment, c'est le seul usage de l'environnement `{NiceMatrixBlock}` mais il pourrait y en avoir davantage dans le futur.

aucune macro privée de `siunitx`.

```
\begin{pNiceArrayRC}[SCwc{1cm}C][nullify-dots]
{C_1} & \Cdots & & C_n \\
2.3 & 0 & \Cdots & 0 \\
12.4 & \Vdots & & \Vdots \\
1.45 & \\
7.2 & 0 & \Cdots & 0 \\
\end{pNiceArrayRC}
```

$$\begin{pmatrix} C_1 & \dots & C_n \\ 2.3 & 0 & \dots & 0 \\ 12.4 & \vdots & & \vdots \\ 1.45 & \vdots & & \vdots \\ 7.2 & 0 & \dots & 0 \end{pmatrix}$$

13 Remarques techniques

13.1 Lignes diagonales

Par défaut, toutes les lignes diagonales¹³ d'un même tableau sont «parallélisées». Cela signifie que la première diagonale est tracée et que, ensuite, les autres lignes sont tracées parallèlement à la première (par rotation autour de l'extrémité la plus à gauche de la ligne). C'est pourquoi la position des instructions `\Ddots` dans un tableau peut avoir un effet marqué sur le résultat final.

Dans les exemples suivants, la première instruction `\Ddots` est marquée en couleur :

Exemple avec parallélisation (comportement par défaut) :

```
$A = \begin{pNiceMatrix}
1 & \Cdots & & 1 \\
a+b & \Ddots & & \Vdots \\
\Vdots & \Ddots & & \\
a+b & \Cdots & a+b & 1 \\
\end{pNiceMatrix}
```

$$A = \begin{pmatrix} 1 & \dots & \dots & 1 \\ a+b & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ a+b & \dots & a+b & 1 \end{pmatrix}$$

```
$A = \begin{pNiceMatrix}
1 & \Cdots & & 1 \\
a+b & & & \Vdots \\
\Vdots & \Ddots & \Ddots & \\
a+b & \Cdots & a+b & 1 \\
\end{pNiceMatrix}
```

$$A = \begin{pmatrix} 1 & \dots & \dots & 1 \\ a+b & & & \vdots \\ \vdots & \dots & \dots & \vdots \\ a+b & \dots & a+b & 1 \end{pmatrix}$$

Il est possible de désactiver la parallélisation avec l'option `parallelize-diags` mise à `false` :

Le même exemple sans parallélisation :

$$A = \begin{pmatrix} 1 & \dots & \dots & 1 \\ a+b & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ a+b & \dots & a+b & 1 \end{pmatrix}$$

13.2 Les cases «vides»

Une instruction comme `\Ldots`, `\Cdots`, etc. essaye de déterminer la première case vide de part et d'autre de la case considérée. Néanmoins, une case vide n'est pas nécessairement sans contenu dans le codage TeX (c'est-à-dire sans aucun token entre les deux esperluettes `&`). En effet, une case dont le contenu est `\hspace*{1cm}` peut être considérée comme vide.

Pour `nicematrix`, les règles précises sont les suivantes :

- Une case implicite est vide. Par exemple, dans la matrice suivante

13. On parle des lignes créées par `\Ddots` et non des lignes créées par une commande `\line` dans le `code-after`.

```

\begin{pmatrix}
a & b \\
c & \\
\end{pmatrix}

```

la dernière case (deuxième rangée et deuxième colonne) est vide.

- Chaque case avec un rendu par TeX de largeur inférieure à 0.5 pt est vide.
- Une case qui contient une commande `\Ldots`, `\Cdots`, `\Vdots`, `\Ddots` ou `\Iddots` (ou une des versions étoilées) est vide. On rappelle que ces commandes doivent être employées seules dans une case.
- Une case avec une commande `\Hspace` (ou `\Hspace*`) est vide. Cette commande `\Hspace` est une commande définie par l’extension `nicematrix` avec la même signification que `\hspace` excepté que la case où cette commande est utilisée est considérée comme vide. Cette commande peut être utilisée pour fixer la largeur des colonnes sans interférer avec le tracé des lignes en pointillés par `nicematrix`.

13.3 L’option `exterior-arraycolsep`

L’environnement `{array}` insère un espace horizontal égal à `\arraycolsep` avant et après chaque colonne. En particulier, il y a un espace égal à `\arraycolsep` avant et après le tableau. Cette caractéristique de l’environnement `{array}` n’était probablement pas une bonne idée.¹⁴

L’environnement `{matrix}` et ses variantes (`{pmatrix}`, `{vmatrix}`, etc.) de `amsmath` préfèrent supprimer ces espaces avec des instructions explicites `\hskip -\arraycolsep` et `{NiceArray}` agit de même.

Néanmoins, l’utilisateur peut changer ce comportement avec l’option `exterior-arraycolsep` de la commande `\NiceMatrixOptions`. Avec cette option, `{NiceArray}` va insérer le même espacement horizontal que `{array}`.

Cette option est seulement pour la «compatibilité» puisque l’extension `nicematrix` fournit un contrôle plus précis avec les options `left-margin`, `right-margin`, `extra-left-margin` et `extra-right-margin`.

13.4 L’option de classe `draft`

L’extension `nicematrix` est relativement lente pour tracer les lignes en pointillés (créées par `\Cdots`, `\Ldots`, `\Ddots`, etc. mais aussi par le spécificateur «:»).¹⁵

C’est pourquoi, avec l’option de classe `draft`, les lignes en pointillés ne sont pas tracées, pour accélérer la compilation.

13.5 Un problème technique avec l’argument de `\\`

Pour des raisons techniques, si vous utilisez l’argument optionnel de la commande `\\`, l’espace vertical sera aussi ajouté au nœud («normal») correspondant à la case précédente.

```

\begin{pNiceMatrix}
a & \frac{A}{B} \\[2mm]
b & c
\end{pNiceMatrix}

```

$$\begin{pmatrix} a & \frac{A}{B} \\ b & c \end{pmatrix}$$

Il y a deux solutions pour résoudre ce problème. La première solution est d’utiliser une commande TeX pour insérer l’espace entre les deux rangées.

14. Dans la documentation de `{amsmath}`, on peut lire : *The extra space of `\arraycolsep` that `array` adds on each side is a waste so we remove it [in `{matrix}`] (perhaps we should instead remove it from `array` in general, but that’s a harder task)*. Il est possible de supprimer les espaces pour un environnement `{array}` donné par une construction du type `\begin{array}{@{}cccc@{}}`.

15. La principale raison est que nous voulons des lignes en pointillés avec des points ronds et non carrés. Pour atteindre ce but, il a fallu construire notre propre système de lignes en pointillés.

```

\begin{pNiceMatrix}
a & \frac{AB}{B} \\
\noalign{\kern2mm}
b & c \\
\end{pNiceMatrix}

```

$$\begin{pmatrix} a & \frac{A}{B} \\ b & c \end{pmatrix}$$

L'autre solution est d'utiliser la commande `\multicolumn` dans la case précédente :

```

\begin{pNiceMatrix}
a & \multicolumn{1}{C}{\frac{AB}{B}} \\
b & c \\
\end{pNiceMatrix}

```

$$\begin{pmatrix} a & \frac{A}{B} \\ b & c \end{pmatrix}$$

14 Exemples

14.1 Lignes en pointillés

Une matrice tridiagonale :

```

$\begin{pNiceMatrix}[nullify-dots]
a & b & 0 & & \cdots & 0 & \\
b & a & b & \ddots & & \vdots & \\
0 & b & a & \ddots & & & \\
& \ddots & \ddots & \ddots & & 0 & \\
\vdots & & & & & b & \\
0 & \cdots & & 0 & b & a & \\
\end{pNiceMatrix}$

```

$$\begin{pmatrix} a & b & 0 & \cdots & 0 \\ b & a & b & & \vdots \\ 0 & b & a & & 0 \\ & \ddots & \ddots & \ddots & b \\ \vdots & & & & b \\ 0 & \cdots & 0 & b & a \end{pmatrix}$$

Une matrice de permutation :

```

$\begin{pNiceMatrix}
0 & 1 & 0 & & \cdots & 0 & \\
\vdots & & & \ddots & & \vdots & \\
& & & \ddots & & & \\
& & & \ddots & & 0 & \\
0 & 0 & & & & 1 & \\
1 & 0 & & \cdots & & 0 & \\
\end{pNiceMatrix}$

```

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ & & & \ddots & 0 \\ & & & \ddots & 1 \\ 0 & 0 & & & 1 \\ 1 & 0 & \cdots & & 0 \end{pmatrix}$$

Un exemple avec `\Iddots` :

```

$\begin{pNiceMatrix}
1 & \cdots & & 1 & \\
\vdots & & & 0 & \\
& \Iddots & \Iddots & \vdots & \\
1 & 0 & \cdots & 0 & \\
\end{pNiceMatrix}$

```

$$\begin{pmatrix} 1 & \cdots & & 1 \\ \vdots & & & 0 \\ & \ddots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}$$

Un exemple avec `\multicolumn` :

```
\begin{pNiceMatrix}[nullify-dots]
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10\\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10\\
\Cdots & & \multicolumn{6}{C}{10 \text{ autres lignes}} & \Cdots \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{pNiceMatrix}
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \cdots \cdots \cdots 10 \text{ autres lignes} \cdots \cdots \cdots \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

Un exemple avec `\Hdotsfor` :

```
\begin{pNiceMatrix}[nullify-dots]
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
\Vdots & & \Hdotsfor{4} & & \Vdots \\
& & \Hdotsfor{4} & & \\
& & \Hdotsfor{4} & & \\
& & \Hdotsfor{4} & & \\
0 & 1 & 1 & 1 & 1 & 0
\end{pNiceMatrix}
```

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Un exemple pour le résultant de deux polynômes :

```
\setlength{\extrarowheight}{1mm}
\[\begin{NiceArray}{|CCCC|}[columns-width=6mm]
a_0 & & & & b_0 & & & \\
a_1 & & \Ddots & & b_1 & & \Ddots & \\
\Vdots & & \Ddots & & \Vdots & & \Ddots & b_0 \\
a_p & & & a_0 & & & b_1 & \\
& & \Ddots & a_1 & & b_q & & \Vdots \\
& & & \Vdots & & \Ddots & & \\
& & a_p & & & & b_q & \\
\end{NiceArray}\]
```

$$\left| \begin{array}{cccc|cccc} a_0 & & & & b_0 & & & \\ a_1 & & \cdots & & b_1 & & \cdots & \\ \vdots & & \cdots & & \vdots & & \cdots & b_0 \\ a_p & & & a_0 & & & b_1 & \\ & & \cdots & a_1 & & b_q & & \vdots \\ & & & \vdots & & \cdots & & \\ & & a_p & & & & b_q & \end{array} \right|$$

14.2 Largeur des colonnes

Dans l'exemple suivant, nous utilisons `{NiceMatrixBlock}` avec l'option `auto-columns-width` parce que nous voulons la même largeur (automatique) pour toutes les colonnes.

```
\begin{NiceMatrixBlock}[auto-columns-width]
\NiceMatrixOptions{code-for-last-col = \color{blue}\scriptstyle}
\setlength{\extrarowheight}{1mm}
\quad $\begin{pNiceArrayC}{CCCC:C}
1&1&1&1&1&\backslash\backslash
2&4&8&16&9&\backslash\backslash
3&9&27&81&36&\backslash\backslash
4&16&64&256&100&\backslash\backslash
\end{pNiceArrayC}$
...
\end{NiceMatrixBlock}
```

$$\begin{array}{c}
 \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 2 & 4 & 8 & 16 & \vdots & 9 \\ 3 & 9 & 27 & 81 & \vdots & 36 \\ 4 & 16 & 64 & 256 & \vdots & 100 \end{array} \right) & \left| & \begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 0 & 1 & 3 & 7 & \vdots & \frac{7}{2} \\ 0 & 0 & 3 & 18 & \vdots & 6 \\ 0 & 0 & -2 & -14 & \vdots & -\frac{9}{2} \end{array} \right) & \begin{array}{l} L_3 \leftarrow -3L_2 + L_3 \\ L_4 \leftarrow L_2 - L_4 \end{array} \\
 \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 0 & 2 & 6 & 14 & \vdots & 7 \\ 0 & 6 & 24 & 78 & \vdots & 33 \\ 0 & 12 & 60 & 252 & \vdots & 96 \end{array} \right) & \begin{array}{l} L_2 \leftarrow -2L_1 + L_2 \\ L_3 \leftarrow -3L_1 + L_3 \\ L_4 \leftarrow -4L_1 + L_4 \end{array} & \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 0 & 1 & 3 & 7 & \vdots & \frac{7}{2} \\ 0 & 0 & 1 & 6 & \vdots & 2 \\ 0 & 0 & -2 & -14 & \vdots & -\frac{9}{2} \end{array} \right) & \begin{array}{l} L_3 \leftarrow \frac{1}{3}L_3 \end{array} \\
 \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 0 & 1 & 3 & 7 & \vdots & \frac{7}{2} \\ 0 & 3 & 12 & 39 & \vdots & \frac{33}{2} \\ 0 & 1 & 5 & 21 & \vdots & 8 \end{array} \right) & \begin{array}{l} L_2 \leftarrow \frac{1}{2}L_2 \\ L_3 \leftarrow \frac{1}{2}L_3 \\ L_4 \leftarrow \frac{1}{12}L_4 \end{array} & \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & \vdots & 1 \\ 0 & 1 & 3 & 7 & \vdots & \frac{7}{2} \\ 0 & 0 & 1 & 6 & \vdots & 2 \\ 0 & 0 & 0 & -2 & \vdots & -\frac{1}{2} \end{array} \right) & \begin{array}{l} L_4 \leftarrow 2L_3 + L_4 \end{array}
 \end{array}$$

14.3 Comment surligner les cases

Pour mettre en évidence une case, il est possible de «dessiner» l'un des nœuds (le «nœud normal», le «nœud moyen» ou le «nœud large»). Dans l'exemple suivant, on utilise les «nœuds larges» de la diagonale de la matrice (avec la clé de Tikz «`name suffix`», il est facile d'utiliser les «nœuds larges»). Pour avoir la continuité des lignes, on doit fixer `inner sep = -\pgflinewidth/2`.

```
$\left(\backslash\begin{NiceArray}{>{\strut}CCCC}%
[create-extra-nodes,left-margin,right-margin,
code-after = {\begin{tikzpicture}
\name suffix = -large,
every node/.style = {draw,
inner sep = -\pgflinewidth/2}]
\node [fit = (1-1)] {} ;
\node [fit = (2-2)] {} ;
\node [fit = (3-3)] {} ;
\node [fit = (4-4)] {} ;
\end{tikzpicture}}]
a_{11} & a_{12} & a_{13} & a_{14} & \backslash\backslash
a_{21} & a_{22} & a_{23} & a_{24} & \backslash\backslash
a_{31} & a_{32} & a_{33} & a_{34} & \backslash\backslash
a_{41} & a_{42} & a_{43} & a_{44} &
\end{NiceArray}\backslash\right)$
```


$$\begin{pmatrix} \boxed{a_{11}} & a_{12} & a_{13} & a_{14} \\ a_{21} & \boxed{a_{22}} & a_{23} & a_{24} \\ a_{31} & a_{32} & \boxed{a_{33}} & a_{34} \\ a_{41} & a_{42} & a_{43} & \boxed{a_{44}} \end{pmatrix}$$

L'extension `nicematrix` est construite au-dessus de l'environnement `{array}` et, par conséquent, il est possible d'utiliser l'extension `colortbl` dans les environnements de `nicematrix`.

```
\begin{bNiceMatrix}
0 & \Cdots & 0 \\
\rowcolor{red!15} 1 & \Cdots & 1 \\
0 & \Cdots & 0 \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 0 & \cdots & 0 \end{bmatrix}$$

Le résultat peut paraître décevant. C'est pourquoi nous proposons une autre méthode pour surligner une rangée de la matrice. Nous créons un nœud Tikz rectangulaire qui englobe les nœuds de la deuxième rangée en utilisant les outils de la bibliothèque Tikz `fit`. Ce nœud est rempli après la construction de la matrice. Pour que l'on puisse voir le texte *sous* le nœud, nous devons utiliser la transparence avec le `blend mode` égal à `multiply`. Attention : certains lecteurs de PDF ne sont pas capables de rendre la transparence correctement.

```
\tikzset{highlight/.style={rectangle,
    fill=red!15,
    blend mode = multiply,
    rounded corners = 0.5 mm,
    inner sep=1pt}}

\begin{bNiceMatrix}[code-after = {\tikz \node[highlight, fit = (2-1) (2-3)] {} ;}]
0 & \Cdots & 0 \\
1 & \Cdots & 1 \\
0 & \Cdots & 0 \\
\end{bNiceMatrix}
```

$$\begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 0 & \cdots & 0 \end{bmatrix}$$

Ce code échoue avec `latex-dvips-ps2pdf` parce que Tikz pour `dvips`, pour le moment, ne prend pas en charge les *blend modes*. Néanmoins, le code suivant, dans le préambule du document LaTeX, devrait activer les *blend modes* pour ce mode de compilation.

```
\ExplSyntaxOn
\makeatletter
\tl_set:Nn \l_tmpa_tl {pgfsys-dvips.def}
\tl_if_eq:NNT \l_tmpa_tl \pgfsysdriver
{ \cs_set:Npn \pgfsys@blend@mode#1{\special{ps:~/\tl_upper_case:n #1~.setblendmode}}}
\makeatother
\ExplSyntaxOff
```

On considère maintenant la matrice suivante que l'on a appelée `exemple`.

```
\begin{pNiceArrayC}{CCC}[name=exemple,create-extra-nodes]
a & a + b & a + b + c & L_1 \\
a & a & a + b & L_2 \\
a & a & a & L_3 \\
\end{pNiceArrayC}
```

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \end{matrix}$$

Si on veut surligner chaque rangée de la matrice, on peut utiliser la technique précédente trois fois.

```
\tikzset{mes-options/.style={remember picture,
                             overlay,
                             name prefix = exemple-,
                             every node/.style = {fill = red!15,
                                                    blend mode = multiply,
                                                    inner sep = 0pt}}

\begin{tikzpicture}[mes-options]
\node [fit = (1-1) (1-3)] {} ;
\node [fit = (2-1) (2-3)] {} ;
\node [fit = (3-1) (3-3)] {} ;
\end{tikzpicture}
```

On obtient la matrice suivante.

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \end{matrix}$$

Le résultat peut paraître décevant. On peut l'améliorer en utilisant les «nœuds moyens» au lieu des «nœuds normaux».

```
\begin{tikzpicture}[mes-options, name suffix = -medium]
\node [fit = (1-1) (1-3)] {} ;
\node [fit = (2-1) (2-3)] {} ;
\node [fit = (3-1) (3-3)] {} ;
\end{tikzpicture}
```

On obtient la matrice suivante.

$$\begin{pmatrix} a & a+b & a+b+c \\ a & a & a+b \\ a & a & a \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \end{matrix}$$

Dans l'exemple suivant, on utilise les «nœuds larges» pour surligner une zone de la matrice.

```
\left(\,\,\begin{NiceArray}{>{\strut}CCCC}%
[create-extra-nodes,left-margin,right-margin,
code-after = {\tikz \path [name suffix = -large,
                           fill = red!15,
                           blend mode = multiply]
(1-1.north west)
|- (2-2.north west)
|- (3-3.north west)
|- (4-4.north west)
|- (4-4.south east)
|- (1-1.north west) ; } ]
A_{11} & A_{12} & A_{13} & A_{14} \\\
A_{21} & A_{22} & A_{23} & A_{24} \\\
A_{31} & A_{32} & A_{33} & A_{34} \\\
A_{41} & A_{42} & A_{43} & A_{44} \\
\end{NiceArray}\,,\right)
```

$$\left(\begin{array}{cccc} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{array} \right)$$

14.4 Matrices par blocs

Dans l'exemple suivant, on utilise les «nœuds larges» pour construire une matrice par blocs (les lignes en tiretés ont été tracées avec `arydshln`).

```
\NiceMatrixOptions{letter-for-dotted-lines = V}
\left(\begin{NiceArray}{CC:CC}%
  [create-extra-nodes,
   code-after = { \tikz \node [fit = (1-1-large) (2-2-large), inner sep = 0 pt]
                  {\$0_{22}\$} ; } ]
    & & a_{13} & a_{14} \\
    & & a_{23} & a_{24} \\
\hdashline
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{34} & a_{44}
\end{NiceArray}\right)
```

$$D = \left(\begin{array}{cc|cc} & 0_{22} & a_{13} & a_{14} \\ & & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{34} & a_{44} \end{array} \right)$$