L'extension nicematrix *

F. Pantigny
fpantigny@wanadoo.fr

6 avril 2020

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 pgfcore ainsi que le module shapes de PGF (l'extension tikz n'est *pas* chargée). L'utilisateur final n'a qu'à charger l'extension nicematrix avec l'instruction habituelle : `\usepackage{nicematrix}`.

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

— des lignes en pointillés continues [1] ;
— des rangées et colonnes extérieures pour les labels ;
— un contrôle sur la largeur des colonnes.

$$\begin{array}{c} \\ L_1 \\ L_2 \\ \vdots \\ L_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \end{array}$$

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 continues

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

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

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

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.

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

*Ce document correspond à la version 3.15 of nicematrix, en date du 2020/04/06.

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 :

{NiceMatrix}	{NiceArray}	{pNiceArray}
{pNiceMatrix}		{bNiceArray}
{bNiceMatrix}		{BNiceArray}
{BNiceMatrix}		{vNiceArray}
{vNiceMatrix}		{VNiceArray}
{VNiceMatrix}		{NiceArrayWithDelims}

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{...}{...}[2], |, >{...}, <{...}, @{...}, !{...} et *{n}{...} mais les lettres p, m et b ne doivent pas être employées. Voir p. 8 la partie concernant {NiceArray}.

3 Les lignes en pointillés continues

À l'intérieur des environnements de l'extension nicematrix, de nouvelles commandes sont définies : \Ldots, \Cdots, \Vdots, \Ddots, et \Iddots. Ces commandes sont conçues pour être utilisées à la place de \dots, \cdots, \vdots, \ddots et \iddots.[3]

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 vides[4] 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. On peut changer la couleur d'une ligne avec l'option color.[5]

```
\begin{bNiceMatrix}
a_1      & \Cdots &        & & a_1     \\
\Vdots   & a_2    & \Cdots & & a_2     \\
         & \Vdots & \Ddots[color=red] \\
\\
a_1      & a_2    &        & & a_n
\end{bNiceMatrix}
```

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

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}
```

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

2. Pour les colonnes de type w et W, les cases sont composées en mode mathématique (dans les environnements de nicematrix) alors que dans {array} de array, elles sont composées en mode texte.

3. La commande \iddots, définie dans nicematrix, est une variante de \ddots avec les points allant vers le haut. Si mathdots est chargée, la version de mathdots est utilisée. Elle correspond à la commande \adots de unicode-math.

4. La définition précise de ce qui est considéré comme une « case vide » est donnée plus loin (cf. p. 19).

5. Il est aussi possible de changer la couleur de toutes ces lignes pointillées avec l'option xdots/color (*xdots* pour rappeler que cela s'applique à \Cdots, \Ldots, Vdots, etc.) : cf. p. 5).

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 résultant [6]).

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

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

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

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. [7]
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 \\[1cm]
0      & \Cdots &               & 0
\end{bNiceMatrix}
```

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

3.1 L'option nullify-dots

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

```
$A = \begin{pmatrix}
h & i & j & k & l & m \\
x &   &   &   &   & x \\
\end{pmatrix}$
```

$$A = \begin{pmatrix} h & i & j & k & l & m \\ x & & & & & x \end{pmatrix}$$

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

```
$B = \begin{pmatrix}
h & i & j & k & l & m \\
x & \ldots & \ldots & \ldots & \ldots & x \\
\end{pmatrix}$
```

$$B = \begin{pmatrix} h & i & j & k & l & m \\ x & \ldots & \ldots & \ldots & \ldots & x \end{pmatrix}$$

6. Et il n'est pas possible de tracer une ligne `\Ldots` et une ligne `\Cdots` entre les mêmes cases.
7. Dans nicematrix, il faut utiliser `\hspace*` et non `\hspace` car nicematrix utilise array. Remarquons aussi que l'on peut également régler la largeur des colonnes en utilisant l'environnement `{NiceArray}` (ou une de ses variantes) avec une colonne de type `w` ou `W` : cf. p. 12

Par défaut, avec nicematrix, si nous remplaçons `{pmatrix}` par `{pNiceMatrix}` et `\ldots` par `\Ldots`, la géométrie de la matrice n'est pas changée.

```
$C = \begin{pNiceMatrix}
h & i & j & k & l & m \\
x & \Ldots & \Ldots & \Ldots & \Ldots & x \\
\end{pNiceMatrix}$
```

$$C = \begin{pmatrix} h & i & j & k & l & m \\ x & \cdots & \cdots & \cdots & \cdots & x \end{pmatrix}$$

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 rangée. C'est possible en utilisant l'option `nullify-dots` (et une seule instruction `\Ldots` suffit).

```
$D = \begin{pNiceMatrix}[nullify-dots]
h & i & j & k & l & m \\
x & \Ldots & & & & x \\
\end{pNiceMatrix}$
```

$$D = \begin{pmatrix} h & i & j & k & l & m \\ x & \cdots & \cdots & \cdots & \cdots & x \end{pmatrix}$$

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

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.

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

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & \cdots & \cdots & \cdots & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

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

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

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ & \cdots & \cdots & \cdots & \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

Remarque : Contrairement à la commande `\hdotsfor` de amsmath, la commande `\Hdotsfor` est utilisable lorsque l'extension colortbl est chargée (mais vous risquez d'avoir des problèmes si vous utilisez `\rowcolor` sur la même rangée que `\Hdotsfor`).

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 de l'amsmath : `{matrix}`, `{pmatrix}`, etc. En fait, cette option est un alias pour la conjonction de deux options : `renew-dots` et `renew-matrix`.[8]

8. Comme toutes les autres options, les options `renew-dots`, `renew-matrix` et `transparent` peuvent être fixées avec la commande `\NiceMatrixOptions`, mais elles peuvent aussi être passées en option du `\usepackage` (ce sont les trois seules).

— L'option `renew-dots`
Avec cette option, les commandes `\ldots`, `\cdots`, `\vdots`, `\ddots`, `\iddots`[3] et `\hdotsfor` sont redéfinies dans les environnements de nicematrix et agissent alors 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.

```
\NiceMatrixOptions{transparent}
\begin{pmatrix}
1      & \cdots & \cdots & 1      \\
0      & \ddots &        & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0      & \cdots & 0      & 1
\end{pmatrix}
```

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

3.4 Les labels des lignes en pointillés

Les commandes `\Ldots`, `\Cdots`, `\Vdots`, `\Ddots`, `\Iddots` et `\Hdotsfor` (ainsi que la commande `\line` dans le `code-after` décrite p. 8) peuvent en fait prendre deux arguments optionnels spécifiés par les caractères `_` et `^` pour des labels situés au-dessous et au-dessus de la ligne. Les arguments sont composés en mode mathématique avec `\scriptstyle`.

```
$\begin{bNiceMatrix}
1 & \hspace*{1cm}             & 0 \\[8mm]
  & \Ddots^{n \text{ times}} &   \\
0 &                           & 1 \\
\end{bNiceMatrix}$
```

$$\begin{bmatrix} 1 & & 0 \\ & \overset{n \text{ times}}{\iddots} & \\ 0 & & 1 \end{bmatrix}$$

3.5 Personnalisation des lignes en pointillés

Les lignes pointillées tracées par `\Ldots`, `\Cdots`, `\Vdots`, `\Ddots`, `\Iddots` et `\Hdotsfor` (ainsi que par la commande `\line` dans le `code-after` décrite p. 8) peuvent être paramétrées par trois options (que l'on met entre crochets après la commande) :

— `color` ;
— `shorten` ;
— `line-style`.

Ces options peuvent aussi être fixées avec `\NiceMatrixOptions` ou bien au niveau d'un environnement mais elles doivent alors être préfixées par `xdots`, ce qui fait que leurs noms deviennent :

— `xdots/color` ;
— `xdots/shorten` ;
— `xdots/line-style`.

Pour la clarté, dans la suite, on utilisera ces noms-là.

L'option xdots/color

L'option `xdots/color` indique bien entendu la couleur de la ligne tracée. On remarquera néanmoins que les lignes tracées dans les rangées et colonnes extérieures (décrites plus loin) bénificient d'un régime spécial : cf. p. 10.

L'option xdots/shorten

L'option `xdots/shorten` indique la marge qui est laissée aux deux extrémités de la ligne. Le nom s'inspire des options « `shorten >` » et « `shorten <` » de Tikz, mais il faut remarquer que nicematrix

ne propose que `xdots/shorten`. La valeur initiale de ce paramètre est de 0.3 em (il est conseillé d'utiliser une unité de mesure dépendante de la fonte courante).

L'option xdots/line-style

Il faut savoir que, par défaut, les lignes de Tikz tracées avec le paramètre `dotted` sont composées de points carrés et non pas ronds. [9]

```
\tikz \draw [dotted] (0,0) -- (5,0) ;
```

Voulant proposer des lignes avec des points ronds dans le style de celui de `\ldots` (au moins celui des fontes *Computer Modern*), l'extension nicematrix contient en interne son propre système de ligne en pointillés (qui, au passage n'utilise que pgf et non tikz). Ce style est appelé le style `standard`. Cette valeur est la valeur initiale du paramètre `xdots/line-style`.

Néanmoins (quand Tikz est chargé), on peut utiliser pour `xdots/line-style` n'importe quel style proposé par Tikz, c'est-à-dire n'importe quelle suite d'options Tikz applicables à un chemin (à l'exception de « `color` », « `shorten >` » et « `shorten <` »).

Voici par exemple une matrice tridiagonale avec le style `loosely dotted` :

```
$\begin{pNiceMatrix}[nullify-dots,xdots/line-style=loosely dotted]
a      & b      & 0      &        & \Cdots & 0      & \\
b      & a      & b      & \Ddots &        & \Vdots & \\
0      & b      & a      & \Ddots &        &        & \\
       & \Ddots & \Ddots & \Ddots &        & 0      & \\
\Vdots &        &        &        &        & b      & \\
0      & \Cdots &        & 0      & b      & a      &
\end{pNiceMatrix}$
```

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

4 Les nœuds PGF-Tikz créés par l'extension nicematrix

L'extension nicematrix crée un nœud PGF-Tikz pour chaque case (non vide) du tableau considéré. Ces nœuds sont utilisés pour tracer les lignes en pointillés entre les cases du tableau. Toutefois, l'utilisateur peut aussi utiliser directement ces nœuds (s'il a chargé Tikz [10]). 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 rangée 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`.

9. La raison de départ est que le format PDF comporte un système de description de lignes en tiretés, qui, puisqu'il est incorporé dans le PDF, est affiché très rapidement par les lecteurs de PDF. Il est facile à partir de ce type de ligne de créer des lignes de points carrés alors qu'une ligne de points ronds doit être construite explicitement point par point.

10. On rappelle que depuis la version 3.13, nicematrix ne charge plus Tikz par défaut, mais seulement PGF (Tikz est une surcouche de PGF).

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 deux séries de nœuds supplémentaires (*extra nodes* en anglais) : les « nœuds moyens » (*medium nodes* en anglais) et les « nœuds larges » (*large nodes* en anglais). Les premiers sont créés avec l'option `create-medium-nodes` et les seconds avec l'option `create-large-nodes`. [11]

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. [12]

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

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`. [13]

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

Il est aussi possible d'ajouter de l'espace sur les 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.

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

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

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

On explique plus loin comment surligner les nœuds créés par Tikz (cf. p. 24).

11. Il existe aussi l'option `create-extra-nodes` qui est un alias pour la conjonction de `create-medium-nodes` et `create-large-nodes`.

12. Il n'y a pas de « nœuds larges » créés dans les rangées et colonnes extérieures (pour ces rangées et colonnes, voir p. 10).

13. 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` (par défaut, 5 pt).Il existe aussi une option `margin` pour fixer à la fois `left-margin` et `right-margin`.

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.
Si on a chargé Tikz [14], on peut accéder à ces nœuds avec des instructions Tikz classiques. Les nœuds devront 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). Elle peut par exemple être utilisée pour tracer une ligne entre deux cases adjacentes comme dans l'exemple suivant.

```
\NiceMatrixOptions{xdots/shorten = 0.6 em}
\begin{pNiceMatrix}[code-after=\line{2-2}{3-3}]
I       & 0      & \Cdots  &0     \\
0       & I      & \Ddots  &\Vdots\\
\Vdots  &\Ddots  & I       &0     \\
0       &\Cdots  & 0       &I
\end{pNiceMatrix}
```

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

Pour améliorer la lisibilité du code, une syntaxe alternative est proposée : on peut spécifier les instructions du `code-after` à la fin de l'environnement, après le mot-clé `\CodeAfter`.

```
\NiceMatrixOptions{xdots/shorten = 0.6 em}
\begin{pNiceMatrix}
I       & 0      & \Cdots  &0     \\
0       & I      & \Ddots  &\Vdots\\
\Vdots  &\Ddots  & I       &0     \\
0       &\Cdots  & 0       &I
\CodeAfter
\line{2-2}{3-3}
\end{pNiceMatrix}
```

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

6 L'environnement {NiceArray} et ses variantes

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` [15] 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. [16]

Sans surprise, l'extension nicematrix fournit également les variantes `{pNiceArray}`, `{vNiceArray}`, `{VNiceArray}`, `{bNiceArray}` et `{BNiceArray}`.

L'un des intérêts de `{pNiceArray}` par rapport à `{pNiceMatrix}` est la possibilité de tracer des filets verticaux :

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

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

14. On rappelle que depuis la version 3.13, nicematrix ne charge plus Tikz par défaut, mais seulement PGF (Tikz est une surcouche de PGF).
15. Les types de colonnes `L`, `C` et `R` sont définis localement à l'intérieur de `{NiceArray}` avec la commande `\newcolumntype` de array. Cette définition masque une éventuelle définition précédente. En fait, les types de colonnes `w` et `W` sont également redéfinis.
16. Dans une commande `\multicolumn`, on doit également utiliser les lettres `L`, `C` et `R`.

Un autre intérêt est la possibilité d'utiliser plusieurs types d'alignement de colonnes.

```
$\begin{pNiceArray}{LCR}
a_{11}    & \Cdots & a_{1n} \\
a_{21}    &        & a_{2n} \\
\Vdots    &        & \Vdots \\
a_{n-1,1} & \Cdots & a_{n-1,n}
\end{pNiceArray}$
```

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & & a_{2n} \\ \vdots & & \vdots \\ a_{n-1,1} & \cdots & a_{n-1,n} \end{pmatrix}$$

En fait, l'environnement `{pNiceArray}` et ses variantes sont fondés sur un environnement plus général, appelé `{NiceArrayWithDelims}`. 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 `{NiceArrayWithDelims}` si on a besoin de délimiteurs atypiques ou asymétriques.

```
$\begin{NiceArrayWithDelims}
   {\downarrow}{\uparrow}{CCC}[margin]
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{NiceArrayWithDelims}$
```

$$\left\downarrow \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right\uparrow$$

6.1 Le positionnement vertical des matrices

L'extension nicematrix propose aussi une option `baseline` par la position verticale des matrices. Cette option `baseline` prend comme valeur un entier qui indique le numéro de rangée dont la ligne de base servira de ligne de base pour l'environnement `{NiceArray}`.

```
$A = \begin{pNiceMatrix}[baseline=2]
\frac{1}{\sqrt{1+p^2}} & p & 1-p \\
1 & 1 & 1 \\
1 & p & 1+p
\end{pNiceMatrix}$
```

$$A = \begin{pmatrix} \frac{1}{\sqrt{1+p^2}} & p & 1-p \\ 1 & 1 & 1 \\ 1 & p & 1+p \end{pmatrix}$$

L'option `baseline` peut aussi prendre les trois valeurs spéciales `t`, `c` et `b`. Ces trois lettres peuvent aussi être utilisées de manière absolue comme pour l'option de l'environnement `{array}` de array. La valeur initiale de `baseline` est `c`.

Dans l'exemple suivant, on utilise l'option `t` (synonyme de `baseline=t`) immédiatement après un `\item` de liste. On remarquera que la présence d'un `\hline` initial n'empêche pas l'alignement sur la ligne de base de la première rangée (avec `{array}` de array, il faut utiliser `\firsthline` [17]).

```
\begin{enumerate}
\item un item
\smallskip
\item \renewcommand{\arraystretch}{1.2}
$\begin{NiceArray}[t]{LCCCCCC}
\hline
n   & 0 & 1 & 2 & 3 & 4  & 5 \\
u_n & 1 & 2 & 4 & 8 & 16 & 32
\hline
\end{NiceArray}$
\end{enumerate}
```

1. un item
2. | n | 0 | 1 | 2 | 3 | 4 | 5 |
   |---|---|---|---|---|---|---|
   | u_n | 1 | 2 | 4 | 8 | 16 | 32 |

17. On peut aussi utiliser `\firsthline` avec `{NiceArray}`.

Il est également possible d'utiliser les outils de booktabs : `\toprule`, `\bottomrule` et `\midrule`.

```
\begin{enumerate}
\item an item
\smallskip
\item
$\begin{NiceArray}[t]{LCCCCCC}
\toprule
n   & 0 & 1 & 2 & 3 & 4  & 5 \\
\midrule
u_n & 1 & 2 & 4 & 8 & 16 & 32
\bottomrule
\end{NiceArray}$
\end{enumerate}
```

1. an item
2.

n	0	1	2	3	4	5
u_n	1	2	4	8	16	32

7 Les rangées et colonnes extérieures

Les environnements de nicematrix permettent de composer des rangées et des colonnes « extérieures » grâce aux options `first-row`, `last-row`, `first-col` et `last-col`.
Si elle est présente, la « première rangée » (extérieure) est numérotée par 0 (et non 1). Il en est de même pour la « première rangée ».

```
$\begin{pNiceMatrix}[first-row,last-row,first-col,last-col,nullify-dots]
       & C_1    & \Cdots &        & C_4    &        \\
L_1    & a_{11} & a_{12} & a_{13} & a_{14} & L_1    \\
\Vdots & a_{21} & a_{22} & a_{23} & a_{24} & \Vdots \\
       & a_{31} & a_{32} & a_{33} & a_{34} &        \\
L_4    & a_{41} & a_{42} & a_{43} & a_{44} & L_4    \\
       & C_1    & \Cdots &        & C_4    &
\end{pNiceMatrix}$
```

$$\begin{array}{c} \\ L_1 \\ \vdots \\ L_4 \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & \cdots & & C_4 \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \\ \begin{array}{cccc} C_1 & \cdots & & C_4 \end{array} \end{array} \begin{array}{c} \\ L_1 \\ \vdots \\ L_4 \\ \\ \end{array}$$

Il y a plusieurs remarques à formuler.

- Si on utilise un environnement avec préambule explicite (c'est-à-dire `{NiceArray}` ou l'une de ses variantes), on ne doit pas mettre dans ce préambule de spécification de colonne pour les éventuelles première et dernière colonne : ce sera automatiquement (et nécessairement) une colonne `R` pour la première colonne et une colonne `L` pour la dernière.
- On peut se demander comment nicematrix détermine le nombre de rangées et de colonnes nécessaires à la composition de la « dernière rangée » et de la « dernière colonne ».
  - Dans le cas d'un environnement avec préambule, comme `{NiceArray}` ou `{pNiceArray}`, le nombre de colonnes se déduit évidemment du préambule.
  - Dans le cas où l'option `light-syntax` (cf. p. 17) est utilisée, nicematrix profite du fait que cette option nécessite de toutes manières le chargement complet du contenu de l'environnement (d'où l'impossibilité de mettre du verbatim dans ce cas-là) avant composition du tableau. L'analyse du contenu de l'environnement donne le nombre de rangées (mais pas le nombre de colonnes).
  - Dans les autres cas, nicematrix détermine le nombre de rangées et de colonnes à la première compilation et l'écrit dans le fichier `.aux` pour pouvoir l'utiliser à la compilation suivante.

Néanmoins, il est possible de donner le numéro de la dernière rangée et le numéro de la dernière colonne en arguments des options `last-row` *et* `last-col` *pour ce qui permettra d'accélérer le processus complet de compilation.* C'est ce que nous ferons dans la suite.

On peut contrôler l'apparence de ces rangées et colonnes avec les options `code-for-first-row`, `code-for-last-row`, `code-for-first-col` et `code-for-last-col`. Ces options sont des listes de tokens qui seront insérées au début de chaque case de la rangée ou de la colonne considérée.

```
\NiceMatrixOptions{code-for-first-row = \color{red},
                   code-for-first-col = \color{blue},
                   code-for-last-row = \color{green},
                   code-for-last-col = \color{magenta}}
$\begin{pNiceArray}{CC|CC}[first-row,last-row=6,first-col,last-col,nullify-dots]
       & C_1    & \Cdots &        & C_4    &        \\
L_1    & a_{11} & a_{12} & a_{13} & a_{14} & L_1    \\
\Vdots & a_{21} & a_{22} & a_{23} & a_{24} & \Vdots \\
\hline
       & a_{31} & a_{32} & a_{33} & a_{34} &        \\
L_4    & a_{41} & a_{42} & a_{43} & a_{44} & L_4    \\
       & C_1    & \Cdots &        & C_4    &
\end{pNiceArray}$
```

$$\begin{array}{c} \\ L_1 \\ \vdots \\ L_4 \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & \cdots & & C_4 \end{array} \\ \left(\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right) \\ \begin{array}{cccc} C_1 & \cdots & & C_4 \end{array} \end{array} \begin{array}{c} \\ L_1 \\ \vdots \\ L_4 \\ \\ \end{array}$$

Remarques

— Comme on peut le voir dans l'exemple précédent, un filet horizontal (tracé avec `\hline`) ne s'étend pas dans les colonnes extérieures et un filet vertical (spécifié par un caractère « | » dans le préambule du tableau) ne s'étend pas dans les rangées extérieures. [18]

Si on veut définir de nouveaux spécificateurs de colonnes pour des filets (par exemple plus épais), on aura peut-être intérêt à utiliser la commande `\OnlyMainNiceMatrix` décrite p. 18.

— Une spécification de couleur présente dans `code-for-first-row` s'applique à une ligne pointillée tracée dans cette « première rangée » (sauf si une valeur a été donnée à `xdots/color`). Idem pour les autres.

— Sans surprise, une éventuelle option `columns-width` (décrite p. 12) ne s'applique pas à la « première colonne » ni à la « dernière colonne ».

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

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

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

18. Ce dernier point n'est pas valable si on a chargé, en plus de nicematrix, l'extension arydshln. Les extensions nicematrix et arydshln ne sont pas parfaitement compatibles car arydshln redéfinit beaucoup de structures internes à array. Par ailleurs, si on veut vraiment un filet qui s'étende dans la première et la dernière rangée, on peut utiliser `!{\vline}` dans le préambule à la place de `|`.

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

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

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

```
\begin{pNiceArray}{CCCC:C}
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15
\end{pNiceArray}
```

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

Ces lignes en pointillés ne s'étendent pas dans les rangées et colonnes extérieures.

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

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

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 = I}
\arrayrulecolor{blue}
\begin{pNiceArray}{C|C:CIC}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12
\end{pNiceArray}
\arrayrulecolor{black}
```

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

On a utilisé `\arrayrulecolor` de colortbl pour colorier les trois traits.

Remarque : Quand l'extension array (sur laquelle s'appuie nicematrix) est chargée, les traits verticaux et horizontaux que l'on insère rendent le tableau plus large ou plus long d'une quantité égale à la largeur du trait [19]. Avec nicematrix, les lignes en pointillés tracées par `\hdottedline` et « `:` » ont le même effet.

9 La largeur des colonnes

Dans les environnements avec un préambule explicite (comme `{NiceArray}`, `{pNiceArray}`, etc.), il est possible de fixer la largeur d'une colonne avec les lettres classiques `w` et `W` de l'extension array. Dans les environnements de nicematrix, les cases des colonnes de ce type sont composées en mode mathématique (alors que dans `{array}` de array, elles sont composées en mode texte).

19. En fait, cela est vrai pour `\hline` et « `|` » mais pas pour `\cline`.

```
$\begin{pNiceArray}{Wc{1cm}CC}
1  & 12 & -123 \\
12 & 0  & 0    \\
4  & 1  & 2
\end{pNiceArray}$
```

$$\begin{pNiceArray}{Wc{1cm}CC} 1 & 12 & -123 \\ 12 & 0 & 0 \\ 4 & 1 & 2 \end{pNiceArray}$$

Dans les environnements de nicematrix, il est aussi possible de fixer la largeur *minimale* de toutes les colonnes de la matrice directement avec l'option `columns-width`.

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

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

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 avant).

Il est possible de donner la valeur spéciale `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. [20]

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

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

Sans surprise, il est possible de fixer la largeur minimale de toutes les colonnes de toutes les matrices dans une certaine portion de document avec la commande `\NiceMatrixOptions`.

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

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

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` [21]. L'environnement `{NiceMatrixBlock}` n'a pas de rapport direct avec la commande `\Block` présentée juste ci-dessous (cf. p. 14).

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

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

Plusieurs compilations peuvent être nécessaires pour obtenir le résultat désiré.

20. Le résultat est atteint dès la première compilation (mais Tikz écrivant des informations dans le fichier `.aux`, un message demandant une deuxième compilation apparaîtra).

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

10 Les matrices par blocs

Cette partie, qui introduit une commande `\Block`, n'a pas de rapport direct avec l'environnement `{NiceMatrixBlock}` présenté dans la section précédente.

Dans les environnements de nicematrix, on peut utiliser la commande `\Block` pour placer un élément au centre d'un rectangle de cases fusionnées.
La commande `\Block` doit être utilisée dans la case supérieure gauche du bloc avec deux arguments. Le premier argument est la taille de ce bloc avec la syntaxe i-j où i est le nombre de rangées et j le nombre de colonnes du bloc. Le deuxième argument, est, sans surprise, le contenu du bloc (en mode mathématique). Un nœud Tikz correspondant à l'ensemble des cases fusionnées est créé sous le nom « i-j-block » où *nom* est le nom donné au tableau. Si on a demandé la création des nœuds `medium`, alors un nœud de ce type est aussi créé pour ce bloc avec un nom suffixé par `-medium`.

Dans les exemples qui suivent, on utilise la commande `\arrayrulecolor` de colortbl.

```
\arrayrulecolor{blue}
$\begin{bNiceArray}{CCC|C}[margin]
\Block{3-3}{A} & & & 0 \\
& \hspace*{1cm} & & \Vdots \\
& & & 0 \\
\hline
0 & \Cdots& 0 & 0
\end{bNiceArray}$
\arrayrulecolor{black}
```

$$\left[\begin{array}{ccc|c} & & & 0 \\ & A & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right]$$

On peut souhaiter agrandir la taille du « A » placé dans le bloc de l'exemple précédent. Comme il est composé en mode mathématique, on ne peut pas directement utiliser une commande comme `\large`, `\Large` ou `\LARGE`. C'est pourquoi une option à mettre entre chevrons est proposée par `\Block` pour spécifier du code LaTeX qui sera inséré *avant* le début du mode mathématique.

```
\arrayrulecolor{blue}
$\begin{bNiceArray}{CCC|C}[margin]
\Block{3-3}<\Large>{A} & & & 0 \\
& \hspace*{1cm} & & \Vdots \\
& & & 0 \\
\hline
0 & \Cdots& 0 & 0
\end{bNiceArray}$
\arrayrulecolor{black}
```

$$\left[\begin{array}{ccc|c} & & & 0 \\ & A & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right]$$

Pour des raisons techniques, il n'est pas possible d'écrire `\Block{`i-j`}{<}` (mais on peut écrire `\Block{`i-j`}<>{<}` avec le résultat attendu).

11 Fonctionnalités avancées

11.1 Option d'alignement dans NiceMatrix

Les environnements sans préambule (`{NiceMatrix}`, `{pNiceMatrix}`, `{bNiceMatrix}`, etc.) proposent les options `l` et `r` (possédant `L` et `R` comme alias) qui imposent des colonnes alignées à gauche ou à droite. [22]

```
$\begin{bNiceMatrix}[R]
\cos x & - \sin x \\
\sin x & \cos x
\end{bNiceMatrix}$
```

$$\begin{bmatrix} \cos x & -\sin x \\ \sin x & \cos x \end{bmatrix}$$

22. Cela reprend une partie des fonctionnalités proposées par les environnements `{pmatrix*}`, `{bmatrix*}`, etc. de mathtools.

11.2 La commande \rotate

Utilisée au début d'une case, la commande `\rotate` (fournie par nicematrix) compose le contenu après une rotation de 90° dans le sens direct.
Dans l'exemple suivant, on l'utilise dans le `code-for-first-row`.

```
\NiceMatrixOptions%
 {code-for-first-row = \scriptstyle \rotate \text{image de },
  code-for-last-col = \scriptstyle }
$A = \begin{pNiceMatrix}[first-row,last-col=4]
e_1 & e_2 & e_3       \\
1   & 2   & 3   & e_1 \\
4   & 5   & 6   & e_2 \\
7   & 8   & 9   & e_3 \\
\end{pNiceMatrix}$
```

$$A = \begin{pmatrix} \overset{\text{image de } e_1}{1} & \overset{\text{image de } e_2}{2} & \overset{\text{image de } e_3}{3} \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{matrix} e_1 \\ e_2 \\ e_3 \end{matrix}$$

Si la commande `\rotate` est utilisée dans la "dernière rangée" (extérieure à la matrice), les éléments qui subissent cette rotation sont alignés vers le haut.

```
\NiceMatrixOptions%
 {code-for-last-row = \scriptstyle \rotate ,
  code-for-last-col = \scriptstyle }
$A = \begin{pNiceMatrix}[last-row,last-col=4]
1   & 2   & 3   & e_1 \\
4   & 5   & 6   & e_2 \\
7   & 8   & 9   & e_3 \\
\text{image de } e_1 & e_2 & e_3 \\
\end{pNiceMatrix}$
```

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ \underset{\text{image de } e_1}{} & \underset{e_2}{} & \underset{e_3}{} \end{pmatrix} \begin{matrix} e_1 \\ e_2 \\ e_3 \\ \end{matrix}$$

11.3 L'option small

Avec l'option `small`, les environnements de l'extension nicematrix sont composés d'une manière proche de ce que propose l'environnement `{smallmatrix}` de l'amsmath (et les environnements `{psmallmatrix}`, `{bsmallmatrix}`, etc. de mathtools).

```
$\begin{bNiceArray}{CCCC|C}[small,
                            last-col,
                            code-for-last-col = \scriptscriptstyle,
                            columns-width = 3mm ]
1 & -2 & 3 & 4 & 5 \\
0 & 3  & 2 & 1 & 2 & L_2 \gets 2 L_1 - L_2 \\
0 & 1  & 1 & 2 & 3 & L_3 \gets L_1 + L_3 \\
\end{bNiceArray}$
```

$$\left[\begin{array}{cccc|c} 1 & -2 & 3 & 4 & 5 \\ 0 & 3 & 2 & 1 & 2 \\ 0 & 1 & 1 & 2 & 3 \end{array}\right] \begin{matrix} \\ L_2 \leftarrow 2L_1 - L_2 \\ L_3 \leftarrow L_1 + L_3 \end{matrix}$$

On remarquera néanmoins que l'environnement `{NiceMatrix}` avec l'option `small` ne prétend pas être composé exactement comme l'environnement `{smallmatrix}`. C'est que les environnements de nicematrix sont tous fondés sur `{array}` (de array) alors que ce n'est pas le cas de `{smallmatrix}` (fondé directement sur un `\halign` de TeX).

En fait, l'option `small` correspond aux réglages suivants :

— les composantes du tableau sont composées en `\scriptstyle` ;

— `\arraystretch` est fixé à 0.47 ;

— `\arraycolsep` est fixé à 1.45 pt ;

— les caractéristiques des lignes en pointillés sont également modifiées.

11.4 Les compteurs iRow et jCol

Dans les cases du tableau, il est possible d'utiliser les compteurs LaTeX `iRow` et `jCol` qui représentent le numéro de la rangée courante et le numéro de la colonne courante [23]. Bien entendu, l'utilisateur ne doit pas modifier les valeurs de ces compteurs qui sont utilisés en interne par nicematrix.
Dans le `code-after` (cf. p. 8), `iRow` représente le nombre total de rangées (hors éventuelles rangées extérieures) et `jCol` le nombre total de colonnes (hors potentielles colonnes extérieures).

```
$\begin{pNiceMatrix}%
    [first-row,
     first-col,
     code-for-first-row = \mathbf{\alph{jCol}} ,
     code-for-first-col = \mathbf{\arabic{iRow}} ]
&   &    &    &   \\
& 1 & 2  & 3  & 4 \\
& 5 & 6  & 7  & 8 \\
& 9 & 10 & 11 & 12
\end{pNiceMatrix}$
```

$$\begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \end{array} \\ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix} \end{array}$$

Si des compteurs LaTeX nommés `iRow` ou `jCol` sont créés dans le document par d'autres extensions que nicematrix (ou tout simplement par l'utilisateur), ces compteurs sont masqués dans les environnements de nicematrix.

L'extension nicematrix propose aussi des commandes pour composer automatiquement des matrices à partir d'un motif général. Ces commandes sont nommées `\pAutoNiceMatrix`, `\bAutoNiceMatrix`, `\vAutoNiceMatrix`, `\VAutoNiceMatrix` et `\BAutoNiceMatrix`.
Chacune de ces commandes prend deux arguments obligatoires : le premier est la taille de la matrice, sous la forme *n*-*p*, où *n* est le nombre de rangées et *p* est le nombre de colonnes et le deuxième est le motif (c'est-à-dire simplement des tokens qui seront insérés dans chaque case de la matrice, exceptées celles des éventuelles rangées et colonnes extérieures).

```
$C = \pAutoNiceMatrix{3-3}{C_{\arabic{iRow},\arabic{jCol}}}$
```

$$C = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

11.5 Les options hlines, vlines et hvlines

Dans les environnements de nicematrix, on peut bien entendu ajouter des filets horizontaux entre les rangées avec la commande `\hline` et des filets verticaux avec le spécificateur "`|`" dans le préambule de l'environnement. Par souci de commodité, l'extension nicematrix fournit aussi l'option `hlines` (resp. `vlines`) qui impose directement que tous les filets horizontaux (resp. verticaux) soient tracés (à l'exception, très naturelle, des filets extérieurs aux rangées et colonnes extérieures). L'option `hvlines` est la conjonction des options `hlines` et `vlines`.

Dans l'exemple suivant, on utilise la commande `\arrayrulecolor` de colortbl.

```
\arrayrulecolor{blue}
$\begin{NiceArray}{CCCC}%
 [hvlines,first-row,first-col]
  & e & a & b & c \\
e & e & a & b & c \\
a & a & e & c & b \\
b & b & c & e & a \\
c & c & b & a & e
\end{NiceArray}$
\arrayrulecolor{black}
```

	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

23. On rappelle que le numéro de la « première rangée » (si elle existe) est 0 et que le numéro de la « première colonne » (si elle existe) est 0 également.

Il y a néanmoins une différence entre l'utilisation de l'option `vlines` et du spécificateur "`|`" dans le préambule de l'environnement : les filets tracés par `vlines` traversent les double-filets horizontaux tracés par `\hline\hline`.

```
$\begin{NiceArray}{CCCC}[vlines] \hline
a & b & c & d \\ \hline \hline
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\ \hline
\end{NiceArray}$
```

a	b	c	d
1	2	3	4
1	2	3	4

Dans le cas d'un environnement avec délimiteurs (par exemple `{pNiceArray}` ou `{pNiceMatrix}`), l'option `vlines` ne trace pas de filets verticaux au niveau des deux délimiteurs (bien entendu).

```
\setlength{\arrayrulewidth}{0.2pt}
$\begin{pNiceMatrix}[vlines]
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{pNiceMatrix}$
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

11.6 L'option light-syntax

L'option `light-syntax`[24] permet d'alléger la saisie des matrices, ainsi que leur lisibilité dans le source TeX. Lorsque cette option est activée, on doit utiliser le point-virgule comme marqueur de fin de rangée et séparer les colonnes par des espaces ou des tabulations. On remarquera toutefois que, comme souvent dans le monde TeX, les espaces après les séquences de contrôle ne sont pas comptées et que les éléments entre accolades sont considérés comme un tout.

```
$\begin{bNiceMatrix}[light-syntax,first-row,first-col]
{} a             b                 ;
a  2\cos a       {\cos a + \cos b} ;
b \cos a+\cos b  { 2 \cos b }
\end{bNiceMatrix}$
```

$$\begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} 2\cos a & \cos a + \cos b \\ \cos a + \cos b & 2\cos b \end{bmatrix} \end{array}$$

On peut changer le caractère utilisé pour indiquer les fins de rangées avec l'option `end-of-row`. Comme dit précédemment, la valeur initiale de ce paramètre est un point-virgule.

Lorsque l'option `light-syntax` est utilisée, il n'est pas possible de mettre d'éléments en verbatim (avec par exemple la commande `\verb`) dans les cases du tableau.[25]

11.7 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 aucune macro privée de siunitx.

```
$\begin{pNiceArray}{SCWc{1cm}C}[nullify-dots,first-row]
{C_1} & \Cdots &  & C_n \\
2.3  & 0 & \Cdots & 0 \\
12.4 & \Vdots & & \Vdots \\
1.45 \\
7.2  & 0 & \Cdots & 0
\end{pNiceArray}$
```

$$\begin{array}{c} \begin{array}{cc} C_1 & \cdots\cdots\cdots\cdots\cdots C_n \end{array} \\ \begin{pmatrix} 2.3 & 0 \cdots\cdots\cdots\cdots 0 \\ 12.4 & \vdots \qquad\qquad\quad \vdots \\ 1.45 & \vdots \qquad\qquad\quad \vdots \\ 7.2 & 0 \cdots\cdots\cdots\cdots 0 \end{pmatrix} \end{array}$$

En revanche, les colonnes `d` de l'extension dcolumn ne sont pas prises en charge par nicematrix.

24. Cette option est inspirée de l'extension spalign de Joseph Rabinoff.

25. La raison en est que lorsque l'option `light-syntax` est utilisée, le contenu complet de l'environnement est chargé comme un argument de commande TeX. L'environnement ne se comporte plus comme un « vrai » environnement de LaTeX qui se contente d'insérer des commandes avant et après.

12 Remarques techniques

Première remarque : l'extension nicematrix doit être chargée après l'extension underscore. Si ce n'est pas le cas, une erreur est levée.

12.1 Pour définir de nouveaux types de colonnes

L'extension nicematrix fournit la commande `\OnlyMainNiceMatrix` qui est destinée à être utilisée dans des définitions de nouveaux types de colonnes. Son argument n'est exécuté que si on se place dans la partie principale du tableau, c'est-à-dire que l'on n'est pas dans l'une des éventuelles rangées extérieures.
Par exemple, si on souhaite définir un type de colonne `?` pour tracer un trait fort (noir) d'épaisseur 1 pt, on pourra écrire [26] :

```
\newcolumntype{?}{!{\OnlyMainNiceMatrix{\vrule width 1 pt}}}
```

Le trait fort correspondant ne s'étendra pas dans les rangées extérieures :

```
$\begin{pNiceArray}{CC?CC}[first-row,last-row=3]
C_1 & C_2 & C_3 & C_4 \\
a & b & c & d \\
e & f & g & h \\
C_1 & C_2 & C_3 & C_4
\end{pNiceArray}$
```

$$\begin{pmatrix} C_1 & C_2 & C_3 & C_4 \\ a & b & c & d \\ e & f & g & h \\ C_1 & C_2 & C_3 & C_4 \end{pmatrix}$$

Le spécificateur `?` ainsi créé est aussi utilisable dans les environnements `{array}` (de array) et, dans ce cas, `\OnlyMainNiceMatrix` est sans effet.

12.2 Le nom des nœuds PGF créés par nicematrix

Les nœuds PGF-Tikz créés par nicematrix peuvent être utilisés hors des environnements de nicematrix après avoir nommé l'environnement concerné avec l'option `name` (cf. p. 6). Il s'agit là de la méthode conseillée mais on décrit néanmoins maintenant le nom Tikz interne de ces nœuds.
Les environnements créés par nicematrix sont numérotés par un compteur global interne. La commande `\NiceMatrixLastEnv` donne le numéro du dernier de ces environnements (pour LaTeX, il s'agit d'une commande — complètement développable — et non d'un compteur).
Si l'environnement concerné a le numéro n, alors le nœud de la rangée i et de la colonne j a pour nom `nm-`n`-`i`-`j. Les noms des nœuds `medium` et `large` correspondants s'obtiennent en suffixant par `-medium` et `-large`.

12.3 Lignes diagonales

Par défaut, toutes les lignes diagonales [27] 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) :

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

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

26. La commande `\vrule` est une commande de TeX (et non de LaTeX).
27. On parle des lignes créées par `\Ddots` et non des lignes créées par une commande `\line` dans le `code-after`.

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

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

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

Le même exemple sans parallélisation :

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

12.4 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

```
\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 nulle est vide.

— 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.

12.5 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 [28]. L'environnement `{matrix}` et ses variantes (`{pmatrix}`, `{vmatrix}`, etc.) de amsmath préfèrent supprimer ces espaces avec des instructions explicites `\hskip -\arraycolsep` [29]. L'extension nicematrix fait de même dans *tous* ses environnements y compris l'environnement `{NiceArray}`. Néanmoins, si l'utilisateur souhaite que l'environnement `{NiceArray}` se comporte par défaut comme l'environnement `{array}` de array (par exemple pour faciliter l'adaptation d'un document existant), il peut contrôler ce comportement avec l'option `exterior-arraycolsep` accessible via la commande `\NiceMatrixOptions`. Avec cette option, des espaces extérieurs de longueur `\arraycolsep` seront insérés dans les environnements `{NiceArray}` (les autres environnements de l'extension nicematrix ne sont pas affectés).

28. 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).*

29. Et non en insérant `@{}` de part et d'autre du préambule, ce qui fait que la longueur des `\hline` n'est pas modifiée et elle peut paraître trop longue, surtout avec des crochets.

12.6 L'option de classe draft

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

12.7 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 AB \\[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.

```
\begin{pNiceMatrix}
a & \frac AB \\
\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 & \multicolumn1C{\frac AB} \\[2mm]
b & c
\end{pNiceMatrix}
```

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

12.8 Environnements obsolètes

La version 3.0 de nicematrix a introduit l'environnement `{pNiceArray}` (et ses variantes) avec les options `first-row`, `last-row`, `first-col` et `last-col`.
Par conséquent, les environnements suivants, présents dans les versions précédentes de nicematrix sont devenus obsolètes :

— `{NiceArrayCwithDelims}` ;

— `{pNiceArrayC}`, `{bNiceArrayC}`, `{BNiceArrayC}`, `{vNiceArrayC}`, `{VNiceArrayC}` ;

— `{NiceArrayRCwithDelims}` ;

— `{pNiceArrayRC}`, `{bNiceArrayRC}`, `{BNiceArrayRC}`, `{vNiceArrayRC}`, `{VNiceArrayRC}`.

Depuis la version 3.12 de nicematrix, on ne peut utiliser ces environnements que si on a chargé l'extension nicematrix avec l'option `obsolete-environments`. Il faut toutefois avoir conscience que ces environnements seront certainement supprimés dans une prochaine version de nicematrix.

13 Exemples

13.1 Lignes en pointillés

Une matrice de permutation.
À titre d'exemple, on a augmenté la valeur du paramètre `xdots/shorten`.

```
$\begin{pNiceMatrix}[xdots/shorten=6em]
0       & 1 & 0 &        & \Cdots &   0    \\
\Vdots  &   &   & \Ddots &        & \Vdots \\
        &   &   & \Ddots &        &        \\
        &   &   & \Ddots &        &   0    \\
0       & 0 &   &        &        &   1    \\
1       & 0 &   & \Cdots &        &   0
\end{pNiceMatrix}$
```

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

Un exemple avec `\Iddots`. On a augmenté encore davantage la valeur de `xdots/shorten`.

```
$\begin{pNiceMatrix}[xdots/shorten = 0.9em]
1       & \Cdots  &         & 1      \\
\Vdots  &         &         & 0      \\
        & \Iddots & \Iddots & \Vdots \\
1       & 0       & \Cdots  & 0
\end{pNiceMatrix}$
```

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

Un exemple avec `\multicolumn` :

```
\begin{BNiceMatrix}[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{BNiceMatrix}
```

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

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}
```

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

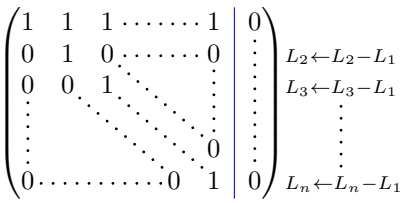
Un exemple pour le résultant de deux polynômes :

```
\setlength{\extrarowheight}{1mm}
\begin{vNiceArray}{CCCC:CCC}[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{vNiceArray}
```

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

Un exemple avec un système linéaire (le trait vertical a été tracé en bleu avec les outils de colortbl) :

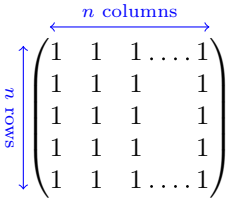
```
\arrayrulecolor{blue}
$\begin{pNiceArray}{*6C|C}[nullify-dots,last-col,code-for-last-col=\scriptstyle]
1      & 1 & 1 &\Cdots &   & 1      & 0      & \\
0      & 1 & 0 &\Cdots &   & 0      &        & L_2 \gets L_2-L_1 \\
0      & 0 & 1 &\Ddots &   & \Vdots &        & L_3 \gets L_3-L_1 \\
       &   &   &\Ddots &   &        & \Vdots & \Vdots \\
\Vdots &   &   &\Ddots &   & 0      & \\
0      &   &   &\Cdots & 0 & 1      & 0      & L_n \gets L_n-L_1
\end{pNiceArray}$
\arrayrulecolor{black}
```



13.2 Des lignes pointillées qui ne sont plus pointillées

L'option `line-style` permet de changer le style des lignes tracées par `\Ldots`, `\Cdots`, etc. On peut de ce fait tracer des lignes qui ne sont plus pointillées.

```
\NiceMatrixOptions
  {nullify-dots,code-for-first-col = \color{blue},code-for-first-col=\color{blue}}
$\begin{pNiceMatrix}[first-row,first-col]
       &   & \Ldots[line-style={solid,<->},shorten=0pt]^{n \text{ columns}} \\
       & 1 & 1 & 1 & \Ldots & 1 \\
       & 1 & 1 & 1 &  & 1 \\
\Vdots[line-style={solid,<->}]_{n \text{ rows}}
       & 1 & 1 & 1 &  & 1 \\
       & 1 & 1 & 1 &  & 1 \\
       & 1 & 1 & 1 & \Ldots & 1 \\
\end{pNiceMatrix}$
```



13.3 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,light-syntax}
\setlength{\extrarowheight}{1mm}
$\begin{pNiceArray}{CCCC:C}[last-col]
  1  1  1   1   1 ;
  2  4  8  16   9 ;
  3  9 27  81  36 ;
  4 16 64 256 100
\end{pNiceArray}$
\medskip
$\begin{pNiceArray}{CCCC:C}[last-col]
  1  1  1   1  1 ;
  0  2  6  14  7        { L_2 \gets -2 L_1 + L_2 } ;
  0  6 24  78 33        { L_3 \gets -3 L_1 + L_3 } ;
  0 12 60 252 96        { L_4 \gets -4 L_1 + L_4 }
\end{pNiceArray}$
...
\end{NiceMatrixBlock}
```

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 9 \\ 3 & 9 & 27 & 81 & 36 \\ 4 & 16 & 64 & 256 & 100 \end{array}\right)$$

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 6 & 14 & 7 \\ 0 & 6 & 24 & 78 & 33 \\ 0 & 12 & 60 & 252 & 96 \end{array}\right) \begin{array}{l} \\ L_2 \leftarrow -2L_1+L_2 \\ L_3 \leftarrow -3L_1+L_3 \\ L_4 \leftarrow -4L_1+L_4 \end{array}$$

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 7 & \frac{7}{2} \\ 0 & 3 & 12 & 39 & \frac{33}{2} \\ 0 & 1 & 5 & 21 & 8 \end{array}\right) \begin{array}{l} \\ L_2 \leftarrow \frac{1}{2}L_2 \\ L_3 \leftarrow \frac{1}{2}L_3 \\ L_4 \leftarrow \frac{1}{12}L_4 \end{array}$$

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 7 & \frac{7}{2} \\ 0 & 0 & 3 & 18 & 6 \\ 0 & 0 & -2 & -14 & -\frac{9}{2} \end{array}\right) \begin{array}{l} \\ \\ L_3 \leftarrow -3L_2+L_3 \\ L_4 \leftarrow L_2-L_4 \end{array}$$

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 7 & \frac{7}{2} \\ 0 & 0 & 1 & 6 & 2 \\ 0 & 0 & -2 & -14 & -\frac{9}{2} \end{array}\right) \begin{array}{l} \\ \\ L_3 \leftarrow \frac{1}{3}L_3 \\ \\ \end{array}$$

$$\left(\begin{array}{cccc:c} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 7 & \frac{7}{2} \\ 0 & 0 & 1 & 6 & 2 \\ 0 & 0 & 0 & -2 & -\frac{1}{2} \end{array}\right) \begin{array}{l} \\ \\ \\ L_4 \leftarrow L_3+L_4 \end{array}$$

13.4 Comment surligner les cases

Les exemples suivants nécessitent d'avoir chargé Tikz (nicematrix ne charge que PGF) ainsi que la bibliothèque Tikz `fit`, ce qui peut se faire avec les deux instructions suivantes dans le préambule du document :

```
\usepackage{tikz}
\usetikzlibrary{fit}
```

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 »).
Nous redessinons les nœuds avec de nouveaux nœuds en utilisant la bibliothèque fit de Tikz. Comme nous voulons recréer des nœuds identiques aux premiers, nous devons fixer `inner sep = 0pt` (si on ne fait pas cela, les nouveaux nœuds seront plus grands que les nœuds d'origine créés par nicematrix).

```
$\begin{pNiceArray}{>{\strut}CCCC}[create-large-nodes,margin,extra-margin = 2pt]
  a_{11} & a_{12} & a_{13} & a_{14} \\
  a_{21} & a_{22} & a_{23} & a_{24} \\
  a_{31} & a_{32} & a_{33} & a_{34} \\
  a_{41} & a_{42} & a_{43} & a_{44}
\CodeAfter
  \begin{tikzpicture}[name suffix = -large,
                      every node/.style = {draw,inner sep = 0 pt}]
     \node [fit = (1-1)] {} ;
     \node [fit = (2-2)] {} ;
     \node [fit = (3-3)] {} ;
     \node [fit = (4-4)] {} ;
  \end{tikzpicture}
\end{pNiceArray}$
```

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

On remarquera que les traits que l'on vient de tracer sont dessinés *après* la matrice sans modifier la position des composantes de celle-ci. En revanche, les traits tracés par `\hline`, le spécificateur « | » ou les options `hlines` et `vlines` « écartent » les composantes de la matrice (quand l'extension array est chargée, ce qui est toujours le cas avec nicematrix). [30]

30. En revanche les traits tracés par `\ncline` n'écartent pas les lignes de la matrice.

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. Les possibilités de réglage de colortbl sont néanmoins assez limitées. 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`.

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

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

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

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 rappelle que dans le cas d'un ensemble de cases fusionnées (avec la commande `\Block`), un nœud Tikz est créé pour l'ensemble des cases avec pour nom *i*`-`*j*`-block` où *i* et *j* sont les numéros de ligne et de colonne de la case en haut à gauche (où a été utilisée la commande `\Block`). Si on a demandé la création des nœuds `medium`, alors un nœud de ce type est aussi créé pour ce bloc avec un nom suffixé par `-medium`.

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

```
$\begin{pNiceArray}{CCC}[name=exemple,last-col,create-medium-nodes]
a & a + b & a + b + c & L_1 \\
a & a     & a + b     & L_2 \\
a & a     & a         & L_3
\end{pNiceArray}$
```

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

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-,
                             highlight/.style = {fill = red!15,
                                                 blend mode = multiply,
                                                 inner sep = 0pt,
                                                 fit = #1}}}

\begin{tikzpicture}[mes-options]
\node [highlight = (1-1) (1-3)] {} ;
\node [highlight = (2-1) (2-3)] {} ;
\node [highlight = (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 [highlight = (1-1) (1-3)] {} ;
\node [highlight = (2-1) (2-3)] {} ;
\node [highlight = (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.

```
\begin{pNiceArray}{>{\strut}CCCC}[create-large-nodes,margin,extra-margin=2pt]
  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}
\CodeAfter
  \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) ;
\end{pNiceArray}
```

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

13.5 Utilisation directe des nœuds Tikz

Dans l'exemple suivant, on souhaite illustrer le produit mathématique de deux matrices.

L'utilisation de `{NiceMatrixBlock}` avec l'option `auto-columns-width` va permettre que toutes les colonnes aient la même largeur ce qui permettra un alignement des deux matrices superposées.

```
\begin{NiceMatrixBlock}[auto-columns-width]


\NiceMatrixOptions{nullify-dots}
```

Les trois matrices vont être disposées les unes par rapport aux autres grâce à un tableau de LaTeX.

```
$\begin{array}{cc}
&
```

La matrice B a une « première rangée » (pour C_j) d'où l'option `first-row`.

```
\begin{bNiceArray}{C>{\strut}CCCC}[name=B,first-row]
       &          & C_j                       \\
b_{11} & \Cdots   & b_{1j} & \Cdots & b_{1n} \\
\Vdots &          & \Vdots &        & \Vdots \\
       &          & b_{kj}                    \\
       &          & \Vdots                    \\
 b_{n1}  & \Cdots & b_{nj} & \Cdots & b_{nn}
\end{bNiceArray} \\ \\
```

La matrice A a une « première colonne » (pour L_i) d'où l'option `first-col`.

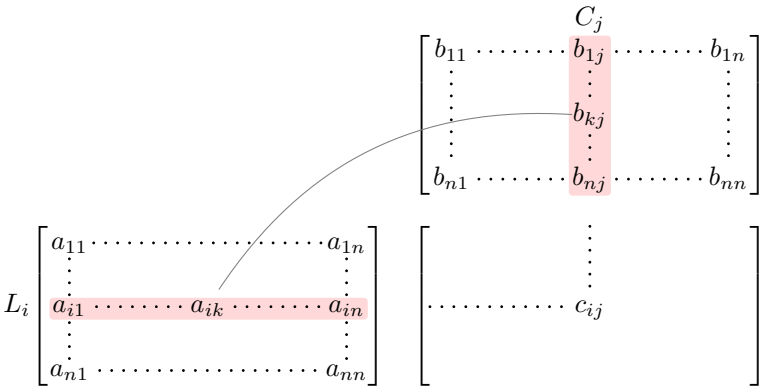
```
\begin{bNiceArray}{CC>{\strut}CCC}[name=A,first-col]
    & a_{11} & \Cdots &        &        & a_{1n} \\
    & \Vdots &        &        &        & \Vdots \\
L_i & a_{i1} & \Cdots & a_{ik} & \Cdots & a_{in} \\
    & \Vdots &        &        &        & \Vdots \\
    & a_{n1} & \Cdots &        &        & a_{nn} \\
\end{bNiceArray}
&
```

Dans la matrice produit, on remarquera que les lignes en pointillés sont « semi-ouvertes ».

```
\begin{bNiceArray}{CC>{\strut}CCC}
       & &        & & \\
       & & \Vdots     \\
\Cdots & & c_{ij}     \\
\\
\\
\end{bNiceArray}
\end{array}$

\end{NiceMatrixBlock}


\begin{tikzpicture}[remember picture, overlay]
 \node [highlight = (A-3-1) (A-3-5) ] {} ;
 \node [highlight = (B-1-3) (B-5-3) ] {} ;
 \draw [color = gray] (A-3-3) to [bend left] (B-3-3) ;
\end{tikzpicture}
```



Autre documentation

Le document `nicematrix.pdf` (fourni avec l'extension nicematrix) contient une traduction anglaise de la documentation ici présente, ainsi que le code source commenté et un historique des versions.