Séance n° 12-ter

Compléments : raffinements mathématiques

12.1 Délimiteurs à taille spécifiée

On a vu à la séance n° 2 que pour obtenir des grands délimiteurs, on pouvait utiliser `\left` et `\right`. En fait, il y a d'autres commandes, `\big`, `\Big`, `\bigg` et `\Bigg` qui permettent de choisir la taille exacte. Ces commandes ont des versions ouvrantes `\bigl`, `\Bigl`, `\biggl` et `\Biggl` et des versions fermantes `\bigr`, `\Bigr`, `\biggr` et `\Biggr`. Voici un exemple d'utilisation :

```
\[\biggl(\sum_{n=0}^{+\infty} \frac{1}{n^2}\biggr)^2\]
```

donne

$$\biggl(\sum_{n=0}^{+\infty} \frac{1}{n^2}\biggr)^2$$

Toujours utiliser une commande se terminant par `l` si c'est un délimiteur ouvrant et une commande se terminant par `r` pour un délimiteur fermant. Pour les rares délimiteurs médiants, utiliser la version sans rien :

```
\[\biggl\{u_n \bigg| \sum u_n \text{ converge} \biggr\}\]
```

donne

$$\biggl\{u_n \bigg| \sum u_n \text{ converge} \biggr\}$$

En règle générale, il vaut mieux utiliser `\left` et `\right`, mais dans certains cas, la taille choisie automatiquement n'est pas optimale. Voici quelques exemples, tous en `\displaystyle`.

ACCEPTABLE	**MEILLEUR**
$\left(\sum_{n=0}^{+\infty} \frac{1}{n^2}\right)^2$	$\left(\sum_{n=0}^{+\infty} \frac{1}{n^2}\right)^2$
$\left\{x \in \mathbb{Z} \,\middle|\, x^2 < 2\right\}$	$\{x \in \mathbb{Z} \mid x^2 < 2\}$
$\ln(1+x) = x + O\left(x^2\right)$	$\ln(1+x) = x + O(x^2)$
$\left\|\frac{1}{2}x_i\right\|$	$\|\frac{1}{2}x_i\|$

Une commande utile pour ce genre de choses est `\DeclarePairedDelimiter` du package `mathtools`. Par exemple,

```
\DeclarePairedDelimiter{\abs}{\lvert}{\rvert}
```

On peut ensuite utiliser la commande `\abs` de la manière suivante avec le résultat suivant :

CODE	RÉSULTAT
`$\abs{\dfrac{1}{2}}$`	$\lvert\frac{1}{2}\rvert$
`$\abs*{\dfrac{1}{2}}$`	$\left\lvert\frac{1}{2}\right\rvert$
`$\abs[\big]{\dfrac{1}{2}}$`	$\big\lvert\frac{1}{2}\big\rvert$
`$\abs[\Big]{\dfrac{1}{2}}$`	$\Big\lvert\frac{1}{2}\Big\rvert$
`$\abs[\bigg]{\dfrac{1}{2}}$`	$\bigg\lvert\frac{1}{2}\bigg\rvert$
`$\abs[\Bigg]{\dfrac{1}{2}}$`	$\Bigg\lvert\frac{1}{2}\Bigg\rvert$

La version étoilée est équivalente à un couple `\left \right`, tandis que dans les autres cas, on choisit la taille. Cela fait une commande très agréable à utiliser pour des petits ajustements.

Exercice 1. — Utiliser la commande `\abs` précédente pour taper la formule suivante.

$$\left|\int_a^b (f(t)+g(t))\,\mathrm{d}t\right| \leq \int_a^b |f(t)+g(t)|\,\mathrm{d}t \leq \int_a^b |f(t)|\,\mathrm{d}t + \int_a^b |g(t)|\,\mathrm{d}t = \left|\frac{1}{x}\right| + \left|\sum_{n=0}^{+\infty} x^{n^2}\right|$$

Solution de l'exercice 1. — Voici le code de la formule précédente :

```
\[\abs*{\int_a^b (f(t)+g(t)) \diff t} \leq \int_a^b \abs{f(t)+g(t)} \diff t \leq
\int_a^b \abs{f(t)} \diff t + \int_a^b \abs{g(t)} \diff t = \abs*{\frac{1}{x}} +
\abs[\bigg]{\sum_{n=0}^{+\infty}{x^{n^2}}}\]
```

Exercice 2 (plus diffile). — Définir des commandes `\bigo` et `\littleo` pour faire les grand O et les petits o en faisant appel à un `\DeclarePaireDelimiter` bien choisi afin que l'on puisse choisir la taille du délimiteur avec `*` ou un `\bigg` explicite.

Solution de l'exercice 2. — On définit d'abord une commande de parenthésage :

```
\DeclarePairedDelimiter{\parenthesage}{(}{)}
```

puis on utilise cette commande pour lui passer l'argument de `\bigo` et `\littleo` :

```
\newcommand{\bigo}{O\parenthesage}
\newcommand{\littleo}{o\parenthesage}
```

Noter qu'il n'y a pas besoin de mettre de `[1]` ou de `#1` car `\parenthesage` sera appelée après `\bigo` et `\littleo` et elle prend déjà un argument, donc `\bigo` et `\littleo` sont des commandes à argument.

Voici des tests :

```
\[\bigo*{\frac{1}{x}} \quad \bigo{\tfrac{1}{x}} \quad \bigo[\bigg]{\tfrac{1}{x}}\]
```

donne :

$$O\left(\frac{1}{x}\right) \quad O(\tfrac{1}{x}) \quad O\left(\tfrac{1}{x}\right)$$

Noter qu'il y a trop d'espace entre le grand O et la parenthèse ouvrante dans la première formule. Nous verrons comment résoudre se problème dans la prochaine section, à savoir :

```
\newcommand{\bigo}{O\mathopen{}\parenthesage}
\newcommand{\littleo}{o\mathopen{}\parenthesage}
```

Compléments. — Voir la séance n° 12-bis pour comment faire des commandes du genre de celles définies par `\DeclarePaireDelimiter`.

12.2 Repérer et régler les problèmes d'espacement

Par défaut, TeX espace automatiquement les formules. L'algorithme utilisé donne de plutôt bons résultats, mais dans certains cas, il faut rectifier l'espacement. On a en déjà vu quelques cas :

- la différentielle $\mathrm{d}x$ que l'on précède d'un `\,`
- les constructions ensemblistes $\{x \in X \mid f(x) \geq 0\}$ où l'on encadre le `\middle|` par des `\,`
- les intervalles $]-1;1[$ où l'on doit utiliser `\mathopen` et `\mathclose` pour spécifier à TeX que] est ouvrant et que [est fermant.

On va voir un certain nombre de cas où il faut corriger l'espacement, et comment le faire de façon optimale.

En règle générale, corriger l'espacement de TeX est une mauvaise idée, car l'espacement est souvent correct. Il ne faut donc utiliser ce qu'on va voir aujourd'hui que dans des cas spéciaux où c'est justifié, pas les utiliser à tout bout de champ. Le mieux est de confiner ces changements dans des macros personnelles, comme ça, en cas d'erreur, il est facile de rectifier.

Les caractères mathématiques sont divisés en les catégories suivantes :

- les caractères ordinaires (`\mathord`), comme a, b, α, Γ, $\mathbb{R}$, etc.
- les opérateurs (`\mathop`), comme $\sum$, $\prod$, $\int$, $\bigoplus$, $\cos$, $\max$, etc.
- les opérations binaires (`\mathbin`), comme $+$, $-$, $\times$, $\otimes$, $\oplus$, $\to$, etc.
- les relations (`\mathrel`), comme $=$, $\in$, $\subset$, $\sim$, $\leq$, $:$, etc.
- les délimiteurs ouvrants (`\mathopen`), comme $($, $[$, $\{$, $\langle$, etc.
- les délimiteurs fermants (`\mathclose`), comme $)$, $]$, $\}$, $\rangle$, etc.
- les ponctuations (`\mathpunct`) comme ,, ;
- les éléments internes (`\mathinner`) comme $\ldots$ et une formule entre `\left` et `\right`.

Les exemples donnés sont les styles par défaut. Certaines fois, il faut forcer le style, comme dans $]-1;1[$ où l'on doit forcer le style de] à être `\mathopen` et celui de [à être `\mathclose`. Pour cela, on écrit

```
$\mathopen{]}-1;1\mathclose{[}$
```

Si on ne le fait pas, l'espacement sera incorrect, comme on le voit en comparant $]-1;1[$ (qui est correct) et `]-1;1[` qui donne $]-1;1[$ (incorrect, il y a trop d'espace avant et après le signe $-$).

Bien sûr, il vaut mieux définir une commande pour les intervalles :

```
\newcommand{\intervalleoo}[2]{\mathopen{]}#1;#2\mathclose{[}}
```

et ensuite utiliser

```
$\intervalleoo{-1}{1}$
```

Exercice 3. — Utiliser ce qu'on vient de voir pour répondre aux questions suivantes.

a. Comment sont définis `\lvert`, `\rvert`, `\mid` à partir de `|` ? Tester sur les formules suivantes :

$$|-1| \qquad d \mid n$$

b. Comment sont définis `\lVert`, `\rVert`, `\parallel` à partir de `\Vert` ? Tester sur les formules suivantes :

$$\|-1\| \qquad (AB) \parallel (CD)$$

c. Comment est défini `\colon` à partir de `:` ? Tester sur la formule suivante :

$$f\colon X \to Y$$

Solution de l'exercice 3. —

a. La commande `\lvert` est `\mathopen{|}`, la commande `\rvert` est `\mathclose{|}` et la commande `\mid` est `\mathrel{|}`.

b. La commande `\lVert` est `\mathopen{\Vert}`, la commande `\rVert` est `\mathclose{\Vert}` et la commande `\parallel` est `\mathrel{\Vert}`.

c. La commande `\colon` est `\mathpunct{:}`.

Exercice 4. — Définir des commandes `\sha` et `\cupproduct` telles que

`A \sha B` donne $A \text{Ш} B$ et `Z_1 \cupproduct Z_2` donne $Z_1 \smile Z_2$

sachant que $\smile$ s'obtient par `\smile` et que Ш s'obtient par

```
\text{\usefont{T1}{wncyr}{m}{n}\symbol{120}}
```

une fois qu'on a rajouté au préambule les lignes suivantes :

```
\DeclareFontFamily{T1}{wncyr}{}
\DeclareFontShape{T1}{wncyr}{m}{n}{%
   <5><6><7><8><9>gen*wncyr%
   <10><10.95><12><14.4><17.28><20.74><24.88>wncyr10}{}
```

Solution de l'exercice 4. — Les définitions sont les suivantes :

```
\newcommand{\sha}{\mathbin{\text{\usefont{T1}{wncyr}{m}{n}\symbol{120}}}}
\newcommand{\cupproduct}{\mathbin{\smile}}
```

On va maintenant voir comment rectifier l'espacement dans certains situations. Une première situation problématique est une fonction de type f, `\cos` ou `\ln` suivie d'une parenthèse introduite par un `\left` ; l'espacement n'est pas correct :

$$Z_X(t) = \exp\left(\sum_{r=1}^{+\infty} |X_r| \frac{t^r}{r}\right)$$

Noter qu'il y a un peu trop d'espace entre le `\exp` et la parenthèse. Pour régler ce problème, il suffit de mettre un `\mathopen{}` (ne pas oublier les accolades vides) entre le `\exp` et le `\left` pour obtenir

$$Z_X(t) = \exp\left(\sum_{r=1}^{+\infty} |X_r| \frac{t^r}{r}\right)$$

qui est correct.

Exercice 5. — Définir des commandes `\littleo` et `\bigo` qui permettent de taper facilement les développements limités suivants :

$$\pi(x) \underset{x\to+\infty}{=} \frac{x}{\ln x} + o\left(\frac{x}{\ln x}\right) \quad \text{et} \quad p_n \underset{n\to+\infty}{\in} O(n \ln n)$$

Solution de l'exercice 5. — Les commandes `\littleo` et `\bigo` sont définies de la façon suivante :

```
\newcommand{\littleo}[1]{o\mathopen{}\left(#1\right)}
\newcommand{\bigo}[1]{O\mathopen{}\left(#1\right)}
```

On tape ensuite les deux développements limités précédents de la façon suivante :

```
\[\pi(x) \underset{x \to +\infty}{=} \frac{x}{\ln x} + \littleo{\frac{x}{\ln x}}
\quad \text{et} \quad p_n \underset{n \to +\infty}{\in} \bigo{n \ln n}\]
```

Remarquons qu'une autre façon de faire (moins robuste, mais plus simple) pour définir `\littleo` et `\bigo` est d'utiliser

```
\newcommand{\littleo}[1]{o{\left(#1\right)}}
\newcommand{\bigo}[1]{O{\left(#1\right)}}
```

Voyons maintenant quelques problèmes avec la position de certains indices ou exposants ; pour cela, on utilisera la commande `\!` qui permet d'introduire un petite espacement négatif. Certaines lettres, comme Γ ou Δ nécessitent des corrections lorsqu'elles sont suivies d'indices ou d'exposants ; par exemple, `$\Gamma_1$` donne Γ_1 est incorrect ; pour obtenir le résultat correct, à savoir Γ_1, il faut utiliser `$\Gamma_{\!1}$`. De même, on utilisera `$\Delta^{\!2}$` pour obtenir Δ^2.

La commande `\!`, tout comme `\,` ne doit pas être utilisé à tort et à travers. Il faut essayer de ne l'utiliser que dans des commandes personnelles où il n'y a aucun risque qu'elles aboutissent à un mauvais espacement et où on peut l'ôter très facilement si le besoin se fait sentir.

Exercice 6. — Définir des commandes `\varaff` (pour les variétés affines) et `\corpsfini` (pour les corps finis) telles que :

`\varaff{n}` donne $\mathbb{A}^n$ et `\corpsfini{q^r}` donne $\mathbb{F}_{q^r}$

Solution de l'exercice 6. — Les commandes sont définies de la façon suivante :

```
\newcommand{\varaff}[1]{\mathbb{A}^{\!#1}}
\newcommand{\corpsfini}[1]{\mathbb{F}_{\!#1}}
```

Un autre cas problématique est la transposée d'une matrice. On peut penser à définir une commande générale du type

```
\newcommand{\transpose}[1]{{}^{t\!}#1}
```

Voici un test :

$$ {}^tA\ {}^tB\ {}^tC\ {}^tD\ {}^tE\ {}^tF\ {}^tG\ {}^tH\ {}^tI\ {}^tJ\ {}^tK\ {}^tL\ {}^tM\ {}^tN\ {}^tO\ {}^tP\ {}^tQ\ {}^tR\ {}^tS\ {}^tT\ {}^tU\ {}^tV\ {}^tW\ {}^tX\ {}^tY\ {}^tZ $$

On voit clairement que l'espacement n'est pas uniforme, par exemple, le t est collé au V, mais pas au J. Le seul moyen [1] de rectifier ce genre de problème est de définir des commandes du type `\tJ` pour rectifier à la main l'espacement.

Exercice 7. — Définir des commandes `\tJ` et `\tV` dont l'espacement est le même que dans `\tA`.

Solution de l'exercice 7. — Voici une définition possible :

```
\newcommand{\tJ}{{}^{t\!\!}J}
\newcommand{\tV}{{}^{t}V}
```

Comparons ces commandes avec `\transpose{A}` :

$$ {}^tA\ {}^tJ\ {}^tV $$

Si on est perfectionniste, il faudrait bien sur répéter la procédure pour toutes les lettres majuscules. Si on n'a pas envie de s'embêter avec ce genre de choses, il suffit de toujours utiliser `\transpose`, qui donne des résultats relativement corrects.

Pour finir, voici quelques exemples où il y a des problèmes d'espacement.

Exercice 8. — Dans les formules suivantes, repérer les problèmes d'espacement et trouver une méthode pour les rectifier.

a. $\lambda \operatorname{Id} + u$
b. $\det \circ \rho$
c. $\sin|_{[-1;1]}$
d. $x/\cos x$
e. $E/\sim$
f. $G/\operatorname{Im}\phi$
g. $\pi \approx 3,14159$
h. $[x_1 : \ldots : x_n]$
i. $[1 : -1 : 0]$
j. $-1 \leq x \leq 1 \quad \text{et} \quad -1 \leq y \leq 1$
k. $1+? = 3 \implies ? = 2$
l. $\mathbb{L}\hat{\otimes}_{\mathbb{K}} A(D(a,r),\mathbb{K})$

Solution de l'exercice 8. — Voici les réponse. On verra ci-dessous (§ 12.3) une méthode pour automatiser ce genre d'ajustements.

a. Dans la formule $\lambda \operatorname{Id} + u$, il n'y a pas assez d'espace entre le $+$ et le u. Rappelons que l'on définit `\Id` par

```
\DeclareMathOperator{\Id}{Id}
```

Pour rectifier le problème, on peut taper :

`\mathord{\lambda \Id} + u` ce qui donne $\lambda \operatorname{Id} + u$

En fait, au lieu de taper `\mathord{\lambda \Id}`, il suffit de taper `{\lambda \Id}`, ça fait la même chose (un groupe entre accolades qui n'est pas un argument de commandes est toujours de type `\mathord`).

1. Dans un futur plus ou moins proche, LaTeX pourra utiliser des polices Opentype mathématiques qui pourront faire ce genre de choses automatiquement.

b. Dans la formule $\det \circ \rho$, le ρ ne devrait pas être collé au $\det$. Pour rectifier ce problème, il suffit d'écrire

```
\mathord{\det} \circ \rho
```
ce qui donne $\det \circ \rho$

(Comme dans l'exemple précédent, on peut utiliser `{\det}` au lieu de `\mathord{det}`.)

c. Dans la formule $\sin|_{[-1;1]}$, il y a trop d'espace avant le $|$. Pour régler ce problème, on utilise

```
{\sin}|_{[-1;1]}
```
ce qui donne $\sin|_{[-1;1]}$

d. Dans la formule $x/\cos x$, il y a trop d'espace entre le $/$ et le $\cos$. Pour rectifier ce problème, on peut taper

```
x/\mathord{\cos x}
```
ce qui donne $x/\cos x$

e. Dans la formule $E/\sim$, le `\sim` est trop éloigné du $/$, ce que l'on rectifie en tapant

```
E/\mathord{\sim}
```
ce qui donne $E/\sim$

f. Dans la formule $G/\operatorname{Im}\phi$, il y a trop d'espace entre le $/$ et le Im. Rappelons que Im est définie, dans le préambule, par

```
\DeclareMathOperator{\Img}{Im}
```
Pour rectifier le problème, on utilise

```
G/\mathord{\Img\phi}
```
ce qui donne $G/\operatorname{Im}\phi$

g. Dans la formule $\pi \approx 3,14159$, il y a trop d'espace entre la virgule et le 1 qui suit. Pour rectifier ce problème, on utilise

```
\pi \approx 3{,}14159
```
ce qui donne $\pi \approx 3{,}14159$

h. Dans la formule $[x_1 : \ldots : x_n]$, le : a trop d'espace avant et après. Pour rectifier le problème, on peut utiliser

```
[x_1\,{:}\dots{:}\,x_n]
```
ce qui donne $[x_1 : \ldots : x_n]$

i. Dans la formule $[1 : -1 : 0]$, outre le même problème d'espacement que dans l'exemples précédent, il y a aussi un problème avec le signe $-$, beaucoup trop espacé avant et après. Pour rectifier le problème, on écrit

```
[1\,{:}\,-1\,{:}\,0]
```
ce qui donne $[1 : -1 : 0]$

Bien sûr, comme pour l'exemple précédent, c'est très lourd à utiliser ; on verra dans le § 12.3 comment automatiser tout cela, ce qui rendra ces changements utilisables (et lisibles !) en pratique.

j. Dans la formule $-1 \leq x \leq 1 \quad \text{et} \quad -1 \leq y \leq 1$, il y a trop d'espace avant et après le signe $-$ après le et. Pour rectifier ce problème, il suffit d'écrire

```
-1 \leq x \leq 1 \quad\text{et}\quad {-1} \leq y \leq 1
```
ce qui donne

$-1 \leq x \leq 1 \quad \text{et} \quad -1 \leq y \leq 1$

k. Dans la formule $1+?=3 \implies ?=2$, il y a des problèmes d'interaction entre le signe $-$ et le ?. Pour rectifier le problème, il suffit d'utiliser

```
1 + {?} = 3 \implies ? = 2
```
ce qui donne $1+?=3 \implies ?=2$

l. Dans la formule $\mathbb{L}\hat{\otimes}_{\mathbb{K}}A(D(a,r),\mathbb{K})$, le $\hat{\otimes}$ n'a pas suffisamment d'espace avant et après. C'est à cause de l'accent qui détruit son appartenance au type `\mathbin` ; il faut donc la restituer :

```
\mathbb{L} \mathbin{\hat{\otimes}}_\mathbb{K} A(D(a,r),\mathbb{K})
```
ce qui donne

$$\mathbb{L} \mathbin{\hat{\otimes}}_{\mathbb{K}} A(D(a,r),\mathbb{K})$$

Bien sur, la bonne méthode est de définir une commande `\hatotimes` qui a le bon espacement incorporé :

```
\newcommand{\hatotimes}{\mathbin{\hat{\otimes}}}
```

12.3 Maîtriser l'espacement mathématique de TeX

Pour finir, voyons des choses un peu plus complexes qui permettent d'automatiser certains espacements. Voici la table décrivant l'algorithme d'espacement de TeX.

	Ord	Op	Bin	Rel	Open	Close	Punct	Inner
Ord	0	1	(2)	(3)	0	0	0	(1)
Op	1	1	*	(3)	0	0	0	(1)
Bin	(2)	(2)	*	*	(2)	*	*	(2)
Rel	(3)	(3)	*	0	(3)	0	0	(3)
Open	0	0	*	0	0	0	0	0
Close	0	1	(2)	(3)	0	0	0	(1)
Punct	(1)	(1)	*	(1)	(1)	(1)	(1)	(1)
Inner	(1)	1	(2)	(3)	(1)	0	(1)	(1)

Légende :
- 0 = rien, 1 = espace fine, 2 = espace moyenne, 3 = espace forte.
- (i) = pas d'espacement si en indice/exposant.
- $*$: le cas ne peut pas se produire.

On va maintenant utiliser cette table pour faire des commandes plus évoluées. La première chose à bien comprendre est que l'on peut facilement gérer différemment l'espacement avant et après une commande. Par exemple, prenons la différentielle `\diff`, déjà vue auparavant. On avait utilisé

```
\newcommand{\diff}{\,\mathrm{d}}
```

Le problème est évident : si on utilise `\diff`, on aura quand même l'espace fine, ce qui est indésirable, par exemple dans la formule

```
\frac{\diff f}{\diff x}
```

qui donne

$$\frac{\mathrm{d}f}{\mathrm{d}x} \quad \text{au lieu de} \quad \frac{\mathrm{d}f}{\mathrm{d}x}$$

On pourrait bien sûr définir une autre commande pour cette situation, mais on va voir que c'est inutile. En fait, si on regarde la table d'espacement, on veut que `\diff` se comporte à gauche comme un `\mathop`. On peut donc penser à utiliser

```
\newcommand{\diff}{\mathop{\mathrm{d}}}
```

Testons le résultat :

$$\int_a^b f(t)\,\mathrm{d}\,t \quad \text{avec } \mathrm{d}\,u = 2t\,\mathrm{d}\,t/\sqrt{t}$$

Le problème est évident : le d n'est pas aligné correctement. Pour résoudre ce problème (qui vient d'un automatisme de TeX utile dans d'autres cas de figure), il suffit de rajouter des accolades vides :

```
\newcommand{\diff}{\mathop{\mathrm{{}d}}}
```

Retestons le résultat :

$$\int_a^b f(t)\,\mathrm{d}\,t \quad \text{avec } \mathrm{d}\,u = 2t\,\mathrm{d}\,t/\sqrt{t}$$

Il y a un autre problème, déjà présent dans le cas précédent : il ne devrait pas y avoir d'espace après le d. Pour supprimer cet espace dans tous les cas, on regarde le tableau d'espacement pour y trouver le type à utiliser pour annuler l'espace. À la ligne Op, il y a des zéros pour les types `\mathopen`, `\mathclose` et `\mathpunct`. Une fois qu'on rajoute ce type, notre `\diff` est, à droite, ce type là, donc il faut choisir le type qui n'est jamais suivi d'espaces. Si on regarde les lignes Open, Close et Punct, on voit que le style `\mathopen` est le seul possible. Finalement, la bonne définition est

```
\newcommand{\diff}{\mathop{\mathrm{{}d}}\mathopen{}}
```

De manière équivalente, on peut utiliser

```
\newcommand{\diff}{\mathop{}\mathopen{}\mathrm{d}}
```

Testons le résultat :

$$\int_a^b f(t)\,\mathrm{d}t \quad \text{avec } \mathrm{d}u = 2t\,\mathrm{d}t/\sqrt{t}$$

Exercice 9. — Utiliser le principe précédent d'espacement différent à gauche et à droite pour faire les commandes suivantes :

a. Une commande `\factorielle` qui ajoute une espace fine après le ! selon le schéma suivant :

$$n!\,m! \qquad (n!)^2 \quad n!!$$

b. Une commande `\Id` qui se comporte ainsi :

$$\lambda\,\mathrm{Id} + u \qquad \mathrm{Id}(x)$$

c. Une commande `\mathslash` qui n'a jamais aucun espace avant ou après :

$$n/\ln n$$

d. Même question pour l'antislash ou la barre verticale :

$$\sin|_{[-1;1]}$$

Solution de l'exercice 9. —

a. Il suffit de prendre

```
\newcommand{\factorielle}[1]{#1!\mathopen{}\mathinner{}}
```

Testons la commande :

```
\[\factorielle{n}\factorielle{m} \qquad (\factorielle{n})^2 \quad
\factorielle{\factorielle{n}}\]
```

$$n!\,m! \qquad (n!)^2 \quad n!!$$

b. Il suffit de prendre

```
\newcommand{\Id}{\mathop{\mathrm{Id}}\mathopen{}\mathord{}}
```

Testons-la commande :

```
\[\lambda \Id + u \qquad \Id(x)\]
```

$$\lambda \operatorname{Id} + u \qquad \operatorname{Id}(x)$$

c. Il suffit de définir

```
\newcommand{\mathslash}{\mathclose{}/\mathopen{}}
```

Testons-la commande :

```
\[n\mathslash\ln n\]
```

$$n/\ln n$$

d. De même :

```
\newcommand{\mathbackslash}{\mathclose{}\backslash\mathopen{}}
\newcommand{\mathvert}{\mathclose{}|\mathopen{}}
```

On a bien :

```
\[\sin\mathvert_{[-1;1]}\]
```

$$\sin|_{[-1;1]}$$

12.4 Récapitulatif de commandes utiles

Petits o et Grands o. Avec le package mathtools :

```
\DeclarePairedDelimiter{\parenthesage}{(}{)}
\newcommand{\bigo}{O\mathopen{}\parenthesage}
\newcommand{\littleo}{o\mathopen{}\parenthesage}
```

Différentielles.

```
\newcommand{\diff}{\mathop{}\mathopen{}\mathrm{d}}
```

Intervalles.

```
\newcommand{\intervalle}[2]{\mathopen{[}#1\,;#2\mathclose{]}}
\newcommand{\intervalleof}[2]{\mathopen{]}#1\,;#2\mathclose{]}}
\newcommand{\intervallefo}[2]{\mathopen{[}#1\,;#2\mathclose{[}}
\newcommand{\intervalleoo}[2]{\mathopen{]}#1\,;#2\mathclose{[}}
```

Ensembles de nombre

```
\newcommand{\ensnb}[1]{\mathbb{#1}}
\newcommand{\N}{\ensnb{R}}
\newcommand{\Z}{\ensnb{R}}
\newcommand{\Q}{\ensnb{R}}
\newcommand{\R}{\ensnb{R}}
\newcommand{\C}{\ensnb{C}}
```

Valeurs absolues, normes. Avec le package mathtools :

```
\newcommand{\card}[1]{\left\lvert#1\right\rvert}
\newcommand{\norme}[1]{\left\lVert#1\right\rVert}
```

Constructions ensemblistes.

```
\makeatletter
\newcommand{\enstq}{\@ifstar\enstq@star\enstq@nostar}
\newcommand{\enstq@star}[2]{\left\lvert#1\,\middle|\,#2\right\rvert}
\newcommand{\enstq@nostar}[3][]{#1\lvert#2\ifx#1\\\\\mid
  \else\mathclose{}\,#1|\,\mathopen{}\fi#3#1\rvert}
\makeatother
```

Coefficients binomiaux.

```
% choisir selon le style voulu
\newcommand{\coeffbinom}[2]{C_{#1}^{#2}}
%\newcommand{\coeffbinom}[2]{\binom{#1}{#2}}
```