Le package `tnsgeo` : pour la géométrie élémentaire

Code source disponible sur `https://github.com/typensee-latex/tnsgeo.git`.

Version `0.3.0-beta` développée et testée sur Mac OS X.

Christophe BAL

2020-08-25

Table des matières

I. Introduction

Le package `tnsgeo` propose des macros utiles pour une rédaction efficace de textes parlant de géométrie élémentaire via un codage sémantique simple.

II. Beta-dépendance

`tnscom` qui est disponible sur `https://github.com/typensee-latex/tnscom.git` est un package utilisé en coulisse.

III. Packages utilisés

La roue ayant déjà été inventée, le package `tnsgeo` réutilise les packages suivants sans aucun scrupule.

- `amssymb`
- `commado`
- `esvect`
- `etoolbox`
- `ifmtarg`
- `mathtools`
- `nicematrix`
- `trimspaces`
- `xstring`
- `yhmath`

IV. Ensembles géométriques

Le package `tnssets` propose la macro `\setgeo` pour indiquer des ensembles géométriques. Se rendre sur `https://github.com/typensee-latex/tnssets.git` si cela vous intéresse.

V. Utiliser des unités S.I.

Comme l'excellent package `siunitx` est chargé en coulisse, il devient facile de travailler avec des unités de mesure tout en respectant **les conventions d'écriture sont françaises** [1] **dès lors que vous aurez chargé `babel` avec l'option `french`** comme c'est le cas pour cette documentation. Notez que les espaces dans `\num{123 000}` et `\num{1230 * 100}` sont inutiles mais ils facilitent la relecture du code.

```
$\ang{180} = \SI{\pi}{\radian}$ ,
$\SI{1}{km} = \SI{1e3}{m}$ avec aussi
$\num{123 000} = \num{1230 * 100}$
```

$180^\circ = \pi\,\mathrm{rad}$, $1\,\mathrm{km} = 1\times10^{3}\,\mathrm{m}$ avec aussi $123\,000 = 1230\times100$

VI. Points et lignes

1. Points

Exemple 1 – Sans indice

```
$\pt{I}$
```

I

1. Notez dans l'exemple l'écriture de 1230 n'utilise pas d'espace contrairement à celle de 12 300.

Exemple 2 – Avec un indice

`$\pt*{I}{1}$ ou` `$\pt*{I}{2}$`	I_1 ou I_2

2. Lignes

Exemple 1 – Les droites

Dans l'exemple suivant, le préfixe `g` est pour `g`-éometrie tandis que `p` est pour `p`-oint.

`$\gline{A}{B}$ ,`
`$\gline{\pt{A}}{\pt{B}}$ ou`
`$\pgline{A}{B}$`
	(AB) , (AB) ou (AB)

Exemple 2 – Les segments

Les macros `\segment` et `\psegment` ont un comportement similaire à `\gline` et `\pgline`.

`$\segment{A}{B}$ ,`
`$\segment{\pt{A}}{\pt{B}}$ ou`
`$\psegment{A}{B}$`
	$[AB]$, [AB] ou [AB]

Exemple 3 – Les demi-droites

Dans l'exemple suivant, le préfixe `h` est pour `h`-alf soit *« moitié »* en anglais.

`$\hgline{A}{B}$ ,`
`$\hgline{\pt{A}}{\pt{B}}$ ou`
`$\phgline{A}{B}$`
	$[AB)$, [AB) ou [AB)

Exemple 4 – D'autres demi-droites

Ce qui suit nécessite d'utilise l'argument optionnel de `\gline` et `\pgline`. La valeur `OC` provient de `O`-pened – `C`-losed soit *« ouvert – fermé »* en anglais.

`$\gline[OC]{A}{B}$ ,`
`$\gline[OC]{\pt{A}}{\pt{B}}$ ou`
`$\pgline[OC]{A}{B}$`
	$(AB]$, (AB] ou (AB]

Remarque. Les segments utilisent en fait l'option `C` et les demi-droites standard l'option `CO`. La valeur par défaut est `O`.

3. Droites parallèles ou non

Les opérateurs `\parallel` et `\nparallel` utilisent des obliques au lieu de barres verticales comme le montre l'exemple qui suit où `\stdnparallel` est un alias de `\nparallel` fourni par le package `amssymb`, et `\stdparallel` est un alias de la version standard de `\parallel` proposée par LaTeX.

```
$\pgline{A}{B} \parallel \pgline{C}{D}$
au lieu de
$\pgline{A}{B}
 \stdparallel \pgline{C}{D}$

$\pgline{E}{F} \nparallel \pgline{G}{H}$
au lieu de
$\pgline{E}{F}
 \stdnparallel \pgline{G}{H}$
```

$\pgline{A}{B} \parallel \pgline{C}{D}$ au lieu de $\pgline{A}{B} \stdparallel \pgline{C}{D}$

$\pgline{E}{F} \nparallel \pgline{G}{H}$ au lieu de $\pgline{E}{F} \stdnparallel \pgline{G}{H}$

VII. Vecteurs

1. Les écrire

Exemple 1

```
$\vect{UnNomLong}$  ,
$\vect*{e}{rot}$ ou
$\vect{e_{rot}}$
```

$\vect{UnNomLong}$, $\vect*{e}{rot}$ ou $\vect{e_{rot}}$

Exemple 2 – Sans point disgracieux

```
$\vect{i}$ ou
$\vect*{j}{2}$
```

$\vect{i}$ ou $\vect*{j}{2}$

Exemple 3 – Se passer de la macro \pt

```
$\pvect{A}{B} = \vect{\pt{A}\pt{B}}$.
```

$\pvect{A}{B} = \vect{\pt{A}\pt{B}}$.

2. Norme

Ci-dessous l'argument optionnel de `\vnorm` vaut `b` par défaut pour `b`-ig soit *« gros »* en anglais mais l'on peut aussi utiliser `s` pour `s`-mall soit *« petit »*. Par contre `\vnorm` n'a pas d'option.

```
 $\norm{\vect{i}} = \vnorm{i}$

 $\norm   {\dfrac{2}{7} \vect*{e}{k}}
= \norm[s]{\dfrac{2}{7} \vect*{e}{k}}$
```

$\norm{\vect{i}} = \vnorm{i}$

$\norm {\dfrac{2}{7} \vect*{e}{k}} = \norm[s]{\dfrac{2}{7} \vect*{e}{k}}$

Remarque. Le code LaTeX pour des doubles barres extensibles ou non vient directement de ce message : `https://tex.stackexchange.com/a/43009/6880`.

3. Produit scalaire

Les 1ers exemples utilisent une syntaxe longue mais adaptables à toutes les situations. Voir l'exemple 5 un peu plus bas pour une écriture rapide utilisable dans certains cas.

Exemple 1 – Version classique

```
$\dotprod{\dfrac{1}{2} \vect{u}}%
         {\vect{v}}$
```

$\dotprod{\dfrac{1}{2} \vect{u}}{\vect{v}}$

Exemple 2 – Version « pédagogique mais pas écolo. »

Dans l'exemple suivant l'option `b` est pour `b`-ullet soit *« puce »* en anglais. Cette écriture peut être utile avec des débutants mais elle est peu pratique pour une écriture manuscrite.

```
$\dotprod[b]{\dfrac{1}{2} \vect{u}}%
            {\vect{v}}$
```

$\dotprod[b]{\dfrac{1}{2} \vect{u}}{\vect{v}}$

Exemple 3 – Écriture « universitaire »

Dans l'exemple suivant l'option `p` est pour `p`-arenthèse et dans `sp` le `s` est pour `s`-mall soit *« petit »* en anglais. On rencontre souvent cette écriture dans les cursus mathématiques universitaires.

```
$\dotprod[p]{\dfrac{1}{2} \vect{u}}%
            {\vect{v}}$

$\dotprod[sp]{\dfrac{1}{2} \vect{u}}%
             {\vect{v}}$
```

$\dotprod[p]{\dfrac{1}{2} \vect{u}}{\vect{v}}$

$\dotprod[sp]{\dfrac{1}{2} \vect{u}}{\vect{v}}$

Exemple 4 – Écriture « à la physicienne »

Dans l'exemple suivant `r` est pour `r`-after soit *« chevron »* en anglais. Les physiciens aiment bien cette notation.

```
$\dotprod[r]{\dfrac{1}{2} \vect{u}}%
            {\vect{v}}$

$\dotprod[sr]{\dfrac{1}{2} \vect{u}}%
             {\vect{v}}$
```

$\dotprod[r]{\dfrac{1}{2} \vect{u}}{\vect{v}}$

$\dotprod[sr]{\dfrac{1}{2} \vect{u}}{\vect{v}}$

Exemple 5 – Version courte mais restrictive

Dans l'exemple suivant le préfixe `v` est pour `v`-ecteur. Notons que dans ce cas les options `sp` et `sr` n'apportent rien de nouveau.

```
 $\vdotprod   {u}{v}
= \vdotprod[b]{u}{v}$

 $\vdotprod[r]{u}{v}
= \vdotprod[p]{u}{v}$
```

$\vdotprod{u}{v} = \vdotprod[b]{u}{v}$

$\vdotprod[r]{u}{v} = \vdotprod[p]{u}{v}$

4. 3D – Produit vectoriel

i. Écriture symbolique

Exemple 1 – Version classique en France

```
$\crossprod{\dfrac{1}{2} \vect{i}}%
           {\vect{j}}$
```

$\crossprod{\dfrac{1}{2} \vect{i}}{\vect{j}}$

Exemple 2 – Version alternative

La macro `\crossprod` possède un argument optionnel que l'on peut utiliser pour obtenir la mise en forme suivante.

```
$\crossprod[t]{\dfrac{1}{2} \vect{i}}%
             {\vect{j}}$
```

$\crossprod[t]{\dfrac{1}{2} \vect{i}}{\vect{j}}$

Exemple 3 – Version courte mais restrictive

```
$\vcrossprod   {i}{j}$ ou
$\vcrossprod[t]{i}{j}$
```

$\vcrossprod{i}{j}$ ou $\vcrossprod[t]{i}{j}$

ii. Explication du mode de calcul

Exemple 1

Dans l'exemple suivant, le préfixe `calc` est pour `calc`-uler quant à `v` est pour `v`-ecteur pour une rédaction raccourcie pour les vecteurs.

```
$\calccrossprod{\vect{u}}{x}{y}{z}%
               {\vect{v}}{x'}{y'}{z'}$
ou
$\vcalccrossprod{AB}{x_B - x_A}%
                    {y_B - y_A}%
                    {z_B - z_A}%
                {CD}{x_D - x_C}%
                    {y_D - y_C}%
                    {z_D - z_C}$
```

$\calccrossprod{\vect{u}}{x}{y}{z}{\vect{v}}{x'}{y'}{z'}$
ou
$\vcalccrossprod{AB}{x_B - x_A}{y_B - y_A}{z_B - z_A}{CD}{x_D - x_C}{y_D - y_C}{z_D - z_C}$

Remarque. Tous les exemples suivants se feront avec `\vcalccrossprod` mais bien entendu ils restent adaptables directement à `\calccrossprod`.

Exemple 2 – Pour un public averti

On peut juste proposer des croix fléchées ou non, voir juste les coordonnées « augmentées » sans les décorations comme ci-après via l'argument optionnel de `\calccrossprod` ou `\vcalccrossprod`.

```
$ \vcalccrossprod[arrows]{u}{x}{y}{z}%
                         {v}{x'}{y'}{z'}
= \vcalccrossprod[cross] {u}{x}{y}{z}%
                         {v}{x'}{y'}{z'}
= \vcalccrossprod[nodeco]{u}{x}{y}{z}%
                         {v}{x'}{y'}{z'}$
```

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix} = \begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix} = \begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix}$$

Exemple 3 – Sans les vecteurs

Si les vecteurs vous gênent il suffira d'utiliser l'option `novec` pour `no vec`-tor soit *« pas de vecteur »* en anglais. Voici deux cas d'utilisation [2].

```
$ \vcalccrossprod[novec]
      {u}{x}{y}{z}{v}{x'}{y'}{z'}
= \vcalccrossprod[novec,cross]
      {u}{x}{y}{z}{v}{x'}{y'}{z'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix} = \begin{vmatrix} x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix}$$

Exemple 4 – Les coordonnées « développées »

Parmi les options proposées par `\calccrossprod` et `\vcalccrossprod`, il y en existe quelques unes fournissant des coordonnées [3] détaillant les calculs. Indiquons que `exp` est pour `exp`-and soit *« développer »* en anglais, `c` pour `\cdot` et enfin `t` pour `\times` [4].

```
$\vcalccrossprod[exp]%
                {u}{\dfrac{1}{2}x}{y}{z}%
                {v}{x'}{y'}{z'}$

$\vcalccrossprod[cexp,vb]%
                {u}{\dfrac{1}{2}x}{y}{z}%
                {v}{x'}{y'}{z'}$

$\vcalccrossprod[texp,sp]%
                {u}{\dfrac{1}{2}x}{y}{z}%
                {v}{x'}{y'}{z'}$
```

$$\left(y\,z' - z\,y' \,;\, z\,x' - \frac{1}{2}x\,z' \,;\, \frac{1}{2}x\,y' - y\,x' \right)$$

$$\begin{bmatrix} y \cdot z' - z \cdot y' \\ z \cdot x' - \frac{1}{2}x \cdot z' \\ \frac{1}{2}x \cdot y' - y \cdot x' \end{bmatrix}$$

$$(y \times z' - z \times y' \,;\, z \times x' - \frac{1}{2}x \times z' \,;\, \frac{1}{2}x \times y' - y \times x')$$

Voici les options disponibles. Nous expliquons ensuite comment les utiliser.

1. `p` vient de `p`-arenthèses. Ceci donnera une écriture horizontale.
2. `b` vient de `b`-rackets soit *« crochets »* en anglais. Ceci donnera une écriture horizontale.
3. `sp` et `sb` produisent des délimiteurs non extensibles en mode horizontal. Ici `s` vient de `s`-mall soit *« petit »* en anglais.
4. `vp` et `vb` produisent des écritures verticales. Ici `v` vient de `v`-ertical.
5. `exp` tout seul demande d'utiliser un espace pour séparer les facteurs de chaque produit.
6. `texp` tout seul demande d'utiliser `\times` comme opérateur de multiplication.
7. `cexp` tout seul demande d'utiliser `\cdot` comme opérateur de multiplication.

2. Il peut sembler un peu lourd d'avoir des arguments pour des vecteurs non affichés mais ce choix permet à l'usage de faire des copier-coller redoutables d'efficacité !

3. En coulisse on utilise la macro `\coord` présentée dans la section 1. page 11 dont on reprend les options de mise en forme.

4. Même si les vecteurs ne sont pas utilisés pour la mise en forme, on obtient ici une méthode très pratique à l'usage car elle permet de faire des copier-coller.

Attention ! Les produits sont rédigés stupidement. Autrement dit ce sera à vous d'ajouter des parenthèses là où il y en aura besoin sinon vous obtiendrez des horreurs comme celle ci-dessous.

```
$\vcalccrossprod[exp,vb]%
 {AB}%
 {x_B - x_A}{y_B - y_A}{z_B - z_A}%
 {CD}%
 {x_D - x_C}{y_D - y_C}{z_D - z_C}$
```

$$\begin{bmatrix} y_B - y_A\,z_D - z_C - z_B - z_A\,y_D - y_C \\ z_B - z_A\,x_D - x_C - x_B - x_A\,z_D - z_C \\ x_B - x_A\,y_D - y_C - y_B - y_A\,x_D - x_C \end{bmatrix}$$

Ici nous n'avons pas d'autre choix que de corriger le tir nous-même. Ceci étant indiqué, ce genre de situation est très rare dans la vraie vie mathématique où l'on évite d'avoir à calculer un produit vectoriel avec des expressions compliquées.

```
$\vcalccrossprod[exp,vb]%
 {AB}%
 {(x_B - x_A)}{(y_B - y_A)}{(z_B - z_A)}%
 {CD}%
 {(x_D - x_C)}{(y_D - y_C)}{(z_D - z_C)}$
```

$$\begin{bmatrix} (y_B - y_A)\,(z_D - z_C) - (z_B - z_A)\,(y_D - y_C) \\ (z_B - z_A)\,(x_D - x_C) - (x_B - x_A)\,(z_D - z_C) \\ (x_B - x_A)\,(y_D - y_C) - (y_B - y_A)\,(x_D - x_C) \end{bmatrix}$$

5. 2D – Critère de colinéarité de deux vecteurs

Exemple 1 – Version complète

Dans l'exemple suivant, le préfixe `coli` est pour `colin`-éarité et `criteria` signifie *« critère »* en anglais.

```
$\colicriteria{\vect{u}}{x}{y}%
              {\vect{v}}{x'}{y'}$
ou
$\colicriteria{\vect{AB}}%
              {x_B - x_A}{y_B - y_A}%
              {\vect{CD}}%
              {x_D - x_C}{y_D - y_C}$
```

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix} \text{ ou } \begin{vmatrix} \overrightarrow{AB} & \overrightarrow{CD} \\ x_B - x_A & x_D - x_C \\ y_B - y_A & y_D - y_C \end{vmatrix}$$

Exemple 2 – Rédaction raccourcie pour les vecteurs

Dans l'exemple suivant, le préfixe `v` est pour `v`-ecteur.

```
$\vcolicriteria{u}{x}{y}%
               {v}{x'}{y'}$
```

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$$

Exemple 3 – Versions sans les vecteurs

Dans l'exemple suivant, on utilise la valeur `novec` pour l'argument optionnel de `\vcolicriteria` qui par défaut est `vec` pour pour `vec`-teur. À l'usage ceci permet des copier-coller très efficaces !

```
$\vcolicriteria[novec]{u}{x}{y}%
                      {v}{x'}{y'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$$

6. 2D – Déterminant de deux vecteurs

Exemple 1 – Version décorée avec une boucle

Dans l'exemple suivant, le préfixe `calc` est pour `calc`-uler quant à `v` est pour `v`-ecteur pour une rédaction raccourcie pour les vecteurs.

```
$\calcdetplane{\vect{u}}{x}{y}%
              {\vect{v}}{x'}{y'}$
ou
$\vcalcdetplane{AB}%
               {x_B - x_A}{y_B - y_A}%
               {CD}%
               {x_D - x_C}{y_D - y_C}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix} \text{ ou } \begin{vmatrix} x_B - x_A & x_D - x_C \\ y_B - y_A & y_D - y_C \end{vmatrix}$$

Remarque. Tous les exemples suivants se feront avec `\vcalcdetplane` mais bien entendu ils restent adaptables directement à `\calcdetplane`.

Exemple 2 – Version décorée avec une croix fléchée

L'argument optionnel de `\calcdetplane` ou `\vcalcdetplane` permet d'obtenir une croix fléchée au lieu d'une boucle.

```
$\vcalcdetplane[arrows]{u}{x}{y}%
                       {v}{x'}{y'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$$

Exemple 3 – Version décorée avec une croix non fléchée

```
$\vcalcdetplane[cross]{u}{x}{y}%
                      {v}{x'}{y'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$$

Exemple 4 – Version non décorée

```
$\vcalcdetplane[nodeco]{u}{x}{y}%
                      {v}{x'}{y'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$$

Exemple 5 – Sans les vecteurs

```
 $\vcalcdetplane[novec]%
                {u}{x}{y} {v}{x'}{y'}
= \vcalcdetplane[novec,cross]%
                {u}{x}{y} {v}{x'}{y'}$
```

$$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix} = \begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$$

Exemple 6 – Calcul développé

Grâce à l'argument optionnel de `\calcdetplane` ou de `\vcalcdetplane` il est aussi possible d'obtenir le résultat développé du calcul comme ci-après où `exp` est pour `exp`-and soit *« développer »* en anglais, `c` pour `\cdot` et enfin `t` pour `\times`[5].

```
$\vcalcdetplane[exp]{u}{x}{y}%
                    {v}{x'}{y'}$

$\vcalcdetplane[cexp]{u}{x}{y}%
                     {v}{x'}{y'}$

$\vcalcdetplane[texp]{u}{x}{y}%
                     {v}{x'}{y'}$
```

$x\,y' - x'\,y$
$x \cdot y' - x' \cdot y$
$x \times y' - x' \times y$

Remarque. Ce qui précède marche aussi avec les versions étoilées.

Attention ! Le développement effectué est stupide. Autrement dit ce sera à vous d'ajouter des parenthèses là où il y en aura besoin sinon vous obtiendrez des horreurs comme celle qui suit.

```
$\vcalcdetplane[exp]{AB}%
                    {x_B - x_A}%
                    {y_B - y_A}%
                    {CD}%
                    {x_D - x_C}%
                    {y_D - y_C}$
```

$x_B - x_A\,y_D - y_C - x_D - x_C\,y_B - y_A$

Ici nous n'avons pas d'autre choix que de régler le problème à la main. Ce genre de situation n'est pas rare dans la vraie vie mathématique.

```
$\vcalcdetplane[exp]{AB}%
                    {(x_B - x_A)}%
                    {(y_B - y_A)}%
                    {CD}%
                    {(x_D - x_C)}%
                    {(y_D - y_C)}$
```

$(x_B - x_A)\,(y_D - y_C) - (x_D - x_C)\,(y_B - y_A)$

VIII. Géométrie cartésienne

1. Coordonnées

Exemple 1 – Des coordonnées seules

`tnsgeo` propose, via un argument optionnel, six façons différentes de rédiger des coordonnées seules *(nous verrons après des macros pour les coordonnées d'un point et celles d'un vecteur afin de produire un code LaTeX plus sémantique)*. Commençons par les écritures horizontales où vous noterez l'utilisation de `|` pour séparer les coordonnées dont le nombre peut être quelconque.

5. Même si les vecteurs ne sont pas utilisés pour la mise en forme, on obtient ici une méthode très pratique à l'usage car elle permet de faire des copier-coller.

```
$\coord    {\dfrac{1}{3} | -4 | 0}$ ou
$\coord[sp]{\dfrac{1}{3} | -4 | 0}$

$\coord[b] {\dfrac{1}{3} | -4 | 0}$ ou
$\coord[sb]{\dfrac{1}{3} | -4 | 0}$
```

$\left(\dfrac{1}{3}\,;-4\,;0\right)$ ou $(\dfrac{1}{3}\,;-4\,;0)$
$\left[\dfrac{1}{3}\,;-4\,;0\right]$ ou $[\dfrac{1}{3}\,;-4\,;0]$

Il existe en plus deux versions verticales.

```
$\coord[vp]{3 | -4}$ ou
$\coord[vb]{3 | -4}$
```

$\begin{pmatrix}3\\-4\end{pmatrix}$ ou $\begin{bmatrix}3\\-4\end{bmatrix}$

Voici d'où viennent les noms des options.

1. `p`, qui est aussi la valeur par défaut, vient de `p`-arenthèses.
2. `b` vient de `b`-rackets soit *« crochets »* en anglais.
3. `s` pour `s`-mall soit *« petit »* en anglais permet d'avoir des délimiteurs non extensibles en mode horizontal car par défaut ils le sont.
4. `v` pour `v`-ertical demande de produire une écriture verticale.

Exemple 2 – Coordonnées d'un point

La macro `\pcoord` avec `p` pour `p`-oint prend un argument supplémentaire avant les coordonnées qui est le nom d'un point qui sera mis en forme par la macro `\pt`. Si vous ne souhaitez pas que `\pt` soit appliquée, il suffit de passer via la version étoilée `\pcoord*`.

```
$\pcoord{A}{3 | -4 | 0 | -1}$ ou
$\pcoord*{\Sigma}{7 | 9 | 8}$
```

$\mathrm{A}\,(\,3\,;-4\,;0\,;-1\,)$ ou $\Sigma\,(\,7\,;9\,;8\,)$

Toutes les options disponibles avec `\coord` le sont aussi avec `\pcoord`.

```
$\pcoord[b]{A}{3 | -4 | 0 | -1}$ ou
$\pcoord*[b]{\Sigma}{7 | 9 | 8}$
```

$\mathrm{A}\,[\,3\,;-4\,;0\,;-1\,]$ ou $\Sigma\,[\,7\,;9\,;8\,]$

Exemple 3 – Coordonnées d'un vecteur

Le fonctionnement de `\vcoord` est similaire à celui de `\pcoord` si ce n'est que c'est la macro `\vect` qui sera appliquée si besoin.

```
$\vcoord{u}{3 | -4}$ ou
$\vcoord*{\dfrac{1}{2} \vect{u}}%
         {3 | -4}$

$\vcoord[vp]{u}{3 | -4}$ ou
$\vcoord*[vp]{\dfrac{1}{2} \vect{u}}%
             {3 | -4}$
```

$\vec{u}\,(\,3\,;-4\,)$ ou $\dfrac{1}{2}\vec{u}\,(\,3\,;-4\,)$
$\vec{u}\begin{pmatrix}3\\-4\end{pmatrix}$ ou $\dfrac{1}{2}\vec{u}\begin{pmatrix}3\\-4\end{pmatrix}$

2. Nommer un repère

Exemple 1 – La méthode basique

Commençons par la manière la plus basique d'écrire un repère *(nous verrons d'autres méthodes qui peuvent être plus efficaces)*.

```
$\axes{\pt{O} %
     | \pt{I} | \pt{J}}$
```

$(\mathrm{O}\,;\mathrm{I},\mathrm{J})$

Exemple 2 – La méthode basique en version étoilée

Dans l'exemple ci-dessous, on voit que la version étoilée produit des petites parenthèses.

```
$\axes{\pt{O} %
     | \dfrac{7}{3} \vect{i} %
     | \vect{j}}$
ou
$\axes*{\pt{O} %
     | \dfrac{7}{3} \vect{i} %
     | \vect{j}}$
```

$\left(\mathrm{O}\,;\dfrac{7}{3}\vec{\imath},\vec{\jmath}\right)$ ou $\left(\mathrm{O}\,;\dfrac{7}{3}\vec{\imath},\vec{\jmath}\,\right)$

Exemple 3 – La méthode basique en dimension quelconque

Il faut au minimum deux "morceaux" séparés par des barres `|`, cas de la dimension 1, mais il n'y a pas de maximum, cas d'une dimension quelconque $n > 0$.

```
$\axes{\pt{O} %
     | \vect*{i}{1} %
     | \vect*{i}{2} %
     | \vect*{i}{3} %
     | \dots %
     | \vect*{i}{9} %
     | \vect*{i}{10} %
     | \vect*{i}{11} %
     | \vect*{i}{12}}$
```

$(\mathrm{O}\,;\vec{\imath}_1,\vec{\imath}_2,\vec{\imath}_3,\ldots,\vec{\imath}_9,\vec{\imath}_{10},\vec{\imath}_{11},\vec{\imath}_{12})$

Exemple 4 – Repère affine

Dans l'exemple suivant, le préfixe `p` est pour `p`-oint.

```
$\paxes{O | I | J | K}$
au lieu de
$\axes{\pt{O} %
     | \pt{I} | \pt{J} | \pt{K}}$
```

$(\mathrm{O}\,;\mathrm{I},\mathrm{J},\mathrm{K})$ au lieu de $(\mathrm{O}\,;\mathrm{I},\mathrm{J},\mathrm{K})$

Exemple 5 – Repère vectoriel (méthode 1)

Dans l'exemple suivant, le préfixe `v` est pour `v`-ecteur.

```
$\vaxes{\pt{O} | i | j}$
au lieu de
$\axes{\pt{O} | \vect{i} | \vect{j}}$
```

$(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$ au lieu de $(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$

Exemple 6 – Repère vectoriel (méthode 2)

Dans l'exemple suivant, le préfixe `pv` permet de combiner ensemble les fonctionnalités proposées par les préfixes `p` et `v`.

```
$\pvaxes{O | i | j}$
au lieu de
$\axes{\pt{O} | \vect{i} | \vect{j}}$
```

$(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$ au lieu de $(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$

IX. Arcs circulaires

Exemple 1

```
$\circarc{UnNomLong}$ ,
$\circarc*{A}{rot}$ ou
$\circarc{A_{rot}}$
```

$\overset{\frown}{UnNomLong}$, $\widehat{A}_{rot}$ ou $\overset{\frown}{A_{rot}}$

Exemple 2

```
$\circarc{i}$ ou
$\circarc*{j}{2}$
```

$\widehat{\imath}$ ou $\widehat{\jmath}_2$

X. Angles

1. Angles géométriques « intérieurs »

Exemple 1

```
$\anglein{UnNomLong}$ ,
$\anglein*{A}{rot}$ ou
$\anglein{A_{rot}}$
```

$\overset{\frown}{UnNomLong}$, $\widehat{A}_{rot}$ ou $\overset{\frown}{A_{rot}}$

Exemple 2 – Cacher les points du i et du j

```
$\anglein{i}$ ou
$\anglein*{j}{2}$
```

$\widehat{\imath}$ ou $\widehat{\jmath}_2$

2. Angles orientés de vecteurs

Sans chapeau - Version longue

L'option par défaut est `p` pour `p`-arenthèse. Dans `sp` le `s` est pour `s`-mall soit *« petit »* en anglais.

```
$\angleorient    {\dfrac{1}{2} \vect{i}}%
                 {\vect{j}}$

$\angleorient[sp]{\dfrac{1}{2} \vect{i}}%
                 {\vect{j}}$
```

$\angleorient {\dfrac{1}{2} \vect{i}}{\vect{j}}$

$\angleorient[sp] {\dfrac{1}{2} \vect{i}}{\vect{j}}$

Sans chapeau - Version courte mais restrictive

Dans l'exemple suivant, le préfixe `v` est pour `v`-ecteur qui permet de simplifier la saisie quand l'on a juste des vecteurs nommés avec des lettres *(notez que l'option* ***`sp`*** *n'apporte rien de nouveau).*

```
$\vangleorient    {i}{j}$ comme
$\vangleorient[sp]{i}{j}$
```

$\vangleorient {i}{j}$ comme $\vangleorient[sp] {i}{j}$

Avec un chapeau

Dans l'exemple suivant, `h` est pour `h`-at soit *« chapeau »* en anglais. Notez au passage que `sh` produit juste des parenthèses petites mais ce choix de nom simplifie l'utilisation de la macro *(c'est mieux que* ***`hsp`*** *par exemple).*

```
$\angleorient[h] {\dfrac{1}{2} \vect{i}}%
                 {\vect{j}}$

$\angleorient[sh]{\dfrac{1}{2} \vect{i}}%
                 {\vect{j}}$

$\vangleorient[h] {i}{j}$ comme
$\vangleorient[sh]{i}{j}$
```

$\angleorient[h] {\dfrac{1}{2} \vect{i}}{\vect{j}}$

$\angleorient[sh] {\dfrac{1}{2} \vect{i}}{\vect{j}}$

$\vangleorient[h] {i}{j}$ comme $\vangleorient[sh] {i}{j}$

XI. Historique

Nous ne donnons ici qu'un très bref historique récent [6] de `tnsgeo` à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier `change-log` : voir le code source de `tnsgeo` sur `github`.

2020-08-25 Nouvelle version mineure `0.3.0-beta`.

- **Vecteur :** ajout de la macro `\pvect` pour ne pas avoir à taper `\pt`.
- **Produit vectoriel et déterminant de deux vecteurs :** nouveau changement de l'API.
  - `\calccrossprod`, `\vcalccrossprod`, `\calcdetplane` et `\vcalcdetplane` permettent de tracer une croix fléchée ou non à la place de la boucle fléchée.
  - `\coordcrossprod` a été supprimé. Il faut à la place utiliser l'une des options `exp`, `texp` et `cexp` de `\vcalccrossprod` et `\calccrossprod`.

2020-07-30 Nouvelle version mineure `0.2.0-beta`.

- **Critère de colinéarité :** ajout de la macro `\colicriteria`.
- **Produit vectoriel :** changement de l'API.
  - `\vcalccrossprod*` devient `\vcalccrossprod**`.
  - `\vcalccrossprod*` dessine des produits en croix à la place des boucles.

2020-07-17 Nouvelle version mineure `0.1.0-beta`.

- **Produit scalaire :** trois nouvelles options pour `\dotprod` et `\vdotprod`.
  - `p` et `sp` donnent une écriture parenthésée.
  - `b` utilise une puce au lieu d'un point centré verticalement.
- **Produit vectoriel :** un nouvel argument optionnel pour `\crossprod` et `\vcrossprod` afin d'obtenir aussi une mise en forme avec le symbole $\times$.

2020-07-10 Première version `0.0.0-beta`.

6. On ne va pas au-delà de un an depuis la dernière version.

XII. Toutes les fiches techniques

1. Points et lignes

i. Points

`\pt{#1}`

— `Argument:` un texte donnant le nom d'un point.

`\pt*{#1..#2}`

— `Argument 1:` un texte indiquant UP dans le nom UP_{down} d'un point.

— `Argument 2:` un texte indiquant *down* dans le nom UP_{down} d'un point.

ii. Lignes

`\gline [#opt]{#1..#2}` `g = g-eometry`
`\pgline[#opt]{#1..#2}` `p = p-oint`

— `Option:` pour indiquer les parenthèses ou crochets à utiliser, les valeurs possibles étant `O`, valeur par défaut, `C`, `CO` et `OC`.

— `Argument 1:` le 1er point géométrique.

— `Argument 2:` le 2e point géométrique.

`\hgline  {#1..#2}` `h = h-alf`
`\phgline {#1..#2}` `phg = p + h + g`
`\segment {#1..#2}`
`\psegment{#1..#2}`

— `Argument 1:` le 1er point géométrique.

— `Argument 2:` le 2e point géométrique.

iii. Droites parallèles ou non

`\parallel`
`\nparallel`
`\stdparallel` (pour utiliser le symbole par défaut)
`\stdnparallel` (pour utiliser le symbole proposé par `amssymb`)

2. Vecteurs

i. Les écrire

`\vect{#1}`

— `Argument:` un texte donnant le nom d'un vecteur.

`\vect*{#1..#2}`

— `Argument 1:` un texte indiquant *up* dans le nom $\overrightarrow{up}_{down}$ d'un vecteur.

— `Argument 2:` un texte indiquant *down* dans le nom $\overrightarrow{up}_{down}$ d'un vecteur.

`\pvect{#1..#2}`

— `Argument 1:` le nom du 1er point qui sera mis en forme via la macro `\pt`.

— `Argument 2:` le nom du 2e point qui sera mis en forme via la macro `\pt`.

ii. Norme

`\norm[#opt]{#1}`

— `Option:` la valeur par défaut est `b`. Deux options disponibles.

1. `b` : des doubles barres extensibles sont utilisées.
2. `s` : des doubles barres non extensibles sont utilisées.

— `Argument:` le vecteur sur lequel appliquer la norme.

`\vnorm{#1}` `v = v-ector`

— `Argument:` le nom du vecteur sur lequel appliquer la norme.

iii. Produit scalaire

`\dotprod[#opt]{#1..#2}`

— `Option:` la valeur par défaut est `u` pour `u`-sual soit *« habituel »* en anglais. Voici les différentes valeurs possibles.

1. `u` : écriture habituelle avec un point.
2. `b` : écriture habituelle mais avec une puce.
3. `p` : écriture « universitaire » avec des parenthèses extensibles.
4. `sp` : écriture « universitaire » avec des parenthèses non extensibles.
5. `r` : écriture « à la physicienne » avec des chevrons extensibles.
6. `sr` : écriture « à la physicienne » avec des chevrons non extensibles.

— `Argument 1:` le 1er vecteur qu'il faut taper via la macro `\vect`.

— `Argument 2:` le 2e vecteur qu'il faut taper via la macro `\vect`.

`\vdotprod[#opt]{#1..#2}` `v = v-ector`

— `Option:` voir les explications précédentes données pour `\dotprod`.

— `Argument 1:` le nom du 1er vecteur sans utiliser la macro `\vect`.

— `Argument 2:` le nom du 2e vecteur sans utiliser la macro `\vect`.

iv. 3D – Produit vectoriel

`\crossprod[#opt]{#1..#2}`

— `Option:` la valeur par défaut est `w` pour `w`-edge soit *« coin »* en anglais. Voici les valeurs possibles.

1. `w` : écriture classique en France.

2. `t` : écriture alternative avec le symbole `\times`.

— `Argument 1:` le 1er vecteur qu'il faut taper via la macro `\vect`.

— `Argument 2:` le 2e vecteur qu'il faut taper via la macro `\vect`.

`\vcrossprod[#opt]{#1..#2}` `v = v-ector`

— `Option:` voir les explications précédentes données pour `\crossprod`.

— `Argument 1:` le nom du 1er vecteur sans utiliser la macro `\vect`.

— `Argument 2:` le nom du 2e vecteur sans utiliser la macro `\vect`.

`\calccrossprod[#opt]{#1..#8}` `calc = calc-ulate`

— `Option:` la valeur par défaut est `vec,loop`. Voici les différentes valeurs possibles.

1. `vec` : pour afficher les vecteurs.
2. `novec` : pour ne pas afficher les vecteurs.
3. `arrows` : des croix fléchées indiquent comment effectuer les calculs.
4. `cross` : des croix non fléchées indiquent comment effectuer les calculs.
5. `loop` : des boucles indiquent comment effectuer les calculs.
6. `nodeco` : rien n'indique comment effectuer les calculs.
7. `exp` : ceci demande d'afficher les formules développées de chaque coordonnée en utilisant un espace pour séparer les facteurs de chaque produit.
8. `cexp` : comme `exp` mais avec le symbole · obtenu via `\cdot`.
9. `texp` : comme `exp` mais avec le symbole ×.
10. `p` : écriture horizontale avec des parenthèses extensibles.
11. `sp` : écriture horizontale avec des parenthèses non extensibles.
12. `vp` : écriture verticale avec des parenthèses.
13. `b` : écriture horizontale avec des crochets extensibles.
14. `sb` : écriture horizontale avec des crochets non extensibles.
15. `vb` : écriture verticale avec des crochets.

On peut combiner deux types de choix cohérents en les séparant par une virgule comme dans `vec,loop` la valeur par défaut de l'option.

— `Argument 1:` le 1er vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 2..4:` les coordonnées du 1er vecteur.

— `Argument 5:` le 2e vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 6..8:` les coordonnées du 2e vecteur.

`\vcalccrossprod[#opt]{#1..#8}` `calc = calc-ulate` et `v = v-ector`

— `Argument 1:` le 1er vecteur sans utiliser la macro `\vect`.

— `Argument 5:` le 2e vecteur sans utiliser la macro `\vect`.

Pour le reste, voir les indications données ci-dessus pour `\calccrossprod`.

v. 2D – Critère de colinéarité de deux vecteurs

`\colicriteria[#opt]{#1..#6}` `coli = coli-nearity`

— `Option:` la valeur par défaut est `vec`. Voici les deux valeurs possibles.

1. `vec` : les vecteurs sont affichés si besoin.
2. `novec` : les vecteurs ne sont jamais affichés.

— `Argument 1:` le 1er vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 2..3:` les coordonnées du 1er vecteur.

— `Argument 4:` le 2e vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 5..6:` les coordonnées du 2e vecteur.

`\vcolicriteria[#opt]{#1..#6}` `v = v-ector`

— `Argument 1:` le 1er vecteur sans utiliser la macro `\vect`.

— `Argument 4:` le 2e vecteur sans utiliser la macro `\vect`.

Pour le reste, voir les indications données ci-dessus pour `\colicriteria`.

vi. 2D – Déterminant de deux vecteurs

`\calcdetplane[#opt]{#1..#6}` `calc = calc-ulate`

— `Option:` la valeur par défaut est `vec,loop`. Voici les différentes valeurs possibles.

1. `vec` : les vecteurs sont affichés si besoin.
2. `novec` : les vecteurs ne sont jamais affichés.
3. `arrows` : des croix fléchées indiquent comment effectuer les calculs.
4. `cross` : des croix non fléchées indiquent comment effectuer les calculs.
5. `loop` : des boucles indiquent comment effectuer les calculs.
6. `nodeco` : rien n'indique comment effectuer les calculs.
7. `exp` : ceci demande d'afficher une formule développée en utilisant un espace pour séparer les facteurs de chaque produit.
8. `cexp` : comme `exp` mais avec le symbole · obtenu via `\cdot`.
9. `texp` : comme `exp` mais avec le symbole ×.

On peut combiner deux types de choix en les séparant par une virgule comme dans `vec,loop` la valeur par défaut de l'option.

— `Argument 1:` le 1er vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 2..3:` les coordonnées du 1er vecteur.

— `Argument 4:` le 2e vecteur qu'il faut taper via la macro `\vect`.

— `Arguments 5..6:` les coordonnées du 2e vecteur.

`\vcalcdetplane[#opt]{#1..#6}` `calc = calc-ulate` et `v = v-ector`

— `Argument 1:` le 1er vecteur sans utiliser la macro `\vect`.

— `Argument 4:` le 2e vecteur sans utiliser la macro `\vect`.

Pour le reste, voir les indications données ci-dessus pour `\calcdetplane`.

3. Géométrie cartésienne

i. Coordonnées

`\coord[#opt]{#1}`

— `Option:` la valeur par défaut est `p`. Voici les différentes valeurs possibles.

1. `p` : écriture horizontale avec des parenthèses extensibles.
2. `sp` : écriture horizontale avec des parenthèses non extensibles.
3. `vp` : écriture verticale avec des parenthèses.
4. `b` : écriture horizontale avec des crochets extensibles.
5. `sb` : écriture horizontale avec des crochets non extensibles.
6. `vb` : écriture verticale avec des crochets.

— `Argument:` l'argument est une suite de "morceaux" séparés par des barres `|` , chaque morceau étant une coordonnée. Il peut n'y avoir qu'un seul morceau.

`\pcoord[#opt]{#1..#2}` `p = p-oint`

— `Option:` voir les indications données pour la macro `\coord`.

— `Argument 1:` le point auquel sera appliqué automatiquement la macro `\pt`.

— `Argument 2:` voir les indications données pour l'unique argument obligatoire de la macro `\coord`.

`\pcoord*[#opt]{#1..#2}`

— `Option:` voir les indications données pour la macro `\coord`.

— `Argument 1:` le point auquel ne sera pas appliqué automatiquement la macro `\pt`.

— `Argument 2:` voir les indications données pour l'unique argument obligatoire de la macro `\coord`.

`\vcoord[#opt]{#1..#2}` `v = v-ertical`

— `Option:` voir les indications données pour la macro `\coord`.

— `Argument 1:` le vecteur sans utiliser la macro `\vect`.

— `Argument 2:` voir les indications données pour l'unique argument obligatoire de la macro `\coord`.

`\vcoord*[#opt]{#1..#2}`

— `Option:` voir les indications données pour la macro `\coord`.

— `Argument 1:` le vecteur qu'il faut taper via la macro `\vect`.

— `Argument 2:` voir les indications données pour l'unique argument obligatoire de la macro `\coord`.

ii. Nommer un repère

```
\axes {#1}
\axes*{#1}
```

— `Argument:` l'argument est une suite de "morceaux" séparés par des barres `|`.

- Le premier morceau est l'origine du repère.

- Les morceaux suivants sont des points ou des vecteurs qui "définissent" chaque axe.

`\paxes[#opt]{#1..# }` `p = p-oint`

— `Argument:` l'argument est une suite de "morceaux" séparés par des barres `|`.
- Le premier morceau est le nom de l'origine du repère sur laquelle la macro-commande `\pt` sera automatiquement appliquée.
- Viennent ensuite les noms des points "définissant" chaque axe. Pour chacun de ces points la macro-commande `\pt` sera automatiquement appliquée.

`\vaxes[#opt]{#1..# }` `v = v-ector`

— `Argument:` l'argument est une suite de "morceaux" séparés par des barres `|`.
- Le premier morceau est l'origine du repère.
- Viennent ensuite les noms des vecteurs "définissant" chaque axe. Pour chacun de ces vecteurs la macro-commande `\vect` sera automatiquement appliquée.

`\pvaxes[#opt]{#1..# }` `pv = p + v`

— `Argument:` l'argument est une suite de "morceaux" séparés par des barres `|`.
- Le premier morceau est le nom de l'origine du repère sur laquelle la macro-commande `\pt` sera automatiquement appliquée.
- Viennent ensuite les noms des vecteurs "définissant" chaque axe. Pour chacun de ces vecteurs la macro-commande `\vect` sera automatiquement appliquée.

4. Arcs circulaires

`\circarc{#1}` `circ = circ-ular`

— `Argument:` un texte donnant le nom d'un arc circulaire.

`\circarc*{#1..#2}`

— `Argument 1:` un texte indiquant *up* dans le nom $\widehat{up}_{down}$ d'un arc circulaire.

— `Argument 2:` un texte indiquant *down* dans le nom $\widehat{up}_{down}$ d'un arc circulaire.

5. Angles

i. Angles géométriques « intérieurs »

`\anglein{#1}` `in = in-terior`

— `Argument:` un texte donnant le nom d'un angle intérieur.

`\anglein*{#1..#2}`

— `Argument 1:` un texte indiquant *up* dans le nom $\widehat{up}_{down}$ d'un angle intérieur.

— `Argument 2:` un texte indiquant *down* dans le nom $\widehat{up}_{down}$ d'un angle intérieur.

ii. Angles orientés de vecteurs

`\angleorient[#opt]{#1..#2}`

— `Option:` la valeur par défaut est `p`. Voici les différentes valeurs possibles.

1. `p` : écriture habituelle avec des parenthèses extensibles.
2. `sp` : écriture habituelle avec des parenthèses non extensibles.
3. `h` : écriture avec un chapeau et des parenthèses extensibles.
4. `sh` : écriture avec un chapeau et des parenthèses non extensibles.

— `Argument 1:` le premier vecteur qu'il faut taper via la macro `\vect`.

— `Argument 2:` le second vecteur qu'il faut taper via la macro `\vect`.

`\vangleorient[#opt]{#1..#2}` `v = v-ector`

— `Option:` voir les explications précédentes données pour `\angleorient`.

— `Argument 1:` le nom du premier vecteur sans utiliser la macro `\vect`.

— `Argument 2:` le nom du second vecteur sans utiliser la macro `\vect`.