

# Le package `tnsgeo`

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

Version 0.0.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-07-10

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Points et lignes</b>	<b>3</b>
2.1	Points . . . . .	3
2.2	Lignes . . . . .	3
2.3	Droites parallèles ou non . . . . .	4
<b>3</b>	<b>Vecteurs</b>	<b>4</b>
3.1	Les écrire . . . . .	4
3.2	Norme . . . . .	4
3.3	Produit scalaire . . . . .	5
3.4	Produit vectoriel . . . . .	5
3.5	Plan – Déterminant de deux vecteurs . . . . .	7
<b>4</b>	<b>Géométrie cartésienne</b>	<b>9</b>
4.1	Coordonnées . . . . .	9
4.2	Nommer un repère . . . . .	10
<b>5</b>	<b>Arcs circulaires</b>	<b>12</b>
<b>6</b>	<b>Angles</b>	<b>12</b>
6.1	Angles géométriques « intérieurs » . . . . .	12
6.2	Angles orientés de vecteurs . . . . .	12
<b>7</b>	<b>Historique</b>	<b>14</b>
<b>8</b>	<b>Toutes les fiches techniques</b>	<b>15</b>
8.1	Introduction . . . . .	15
8.2	Points et lignes . . . . .	15
8.2.1	Points . . . . .	15
8.2.2	Lignes . . . . .	15
8.2.3	Droites parallèles ou non . . . . .	15

8.3	Vecteurs . . . . .	15
8.3.1	Vecteurs . . . . .	15
8.3.2	Norme . . . . .	16
8.3.3	Produit scalaire . . . . .	16
8.3.4	Produit vectoriel . . . . .	16
8.3.5	Plan – Déterminant de deux vecteurs . . . . .	17
8.4	Géométrie cartésienne . . . . .	18
8.4.1	Coordonnées . . . . .	18
8.4.2	Repères . . . . .	19
8.4.3	Arcs circulaires . . . . .	19
8.4.4	Angles géométriques « intérieurs » . . . . .	20
8.4.5	Angles orientés de vecteurs . . . . .	20

---

# 1 Introduction

Le package `tnsgeo` propose des macros utiles pour une rédaction efficace de textes parlant de géométrie élémentaire.

**Remarque.** Ce package s'appuie sur `tnscom` disponible sur <https://github.com/typensee-latex/tnscom.git>.

## 2 Points et lignes

### 2.1 Points

Exemple 1 – Sans indice

<code>\$\pt{I}\$</code>	$I$
-------------------------	-----

Exemple 2 – Avec un indice

<code>\$\pt*{I}{1}\$</code> ou <code>\$\pt*{I}{2}\$</code>	$I_1$ ou $I_2$
---	----------------

### 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.

<code>\$\gline{A}{B}\$</code> , <code>\$\gline{\pt{A}}{\pt{B}}\$</code> ou <code>\$\pgline{A}{B}\$</code>	$(AB)$ , $(AB)$ ou $(AB)$
---	---------------------------

Exemple 2 – Les segments

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

<code>\$\segment{A}{B}\$</code> , <code>\$\segment{\pt{A}}{\pt{B}}\$</code> ou <code>\$\psegment{A}{B}\$</code>	$[AB]$ , $[AB]$ ou $[AB]$
---	---------------------------

Exemple 3 – Les demi-droites

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

<code>\$\hgline{A}{B}\$</code> , <code>\$\hgline{\pt{A}}{\pt{B}}\$</code> ou <code>\$\phgline{A}{B}\$</code>	$[AB)$ , $[AB)$ ou $[AB)$
--	---------------------------

## Exemple 4 – D’autres demi-droites

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

<pre><code>\gline[OC]{A}{B}\$ ,</code> <code>\gline[OC]{\pt{A}}{\pt{B}}\$</code> ou <code>\pgline[OC]{A}{B}\$</code></pre>	$(AB]$ , $(AB]$ 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`.

## 2.3 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 L<sup>A</sup>T<sub>E</sub>X.

<pre><code>\pgline{A}{B} \parallel \pgline{C}{D}\$</code> au lieu de <code>\pgline{A}{B}</code> <code>\stdparallel \pgline{C}{D}\$</code></pre>	$(AB) \parallel (CD)$ au lieu de $(AB) \parallel (CD)$
<pre><code>\pgline{E}{F} \nparallel \pgline{G}{H}\$</code> au lieu de <code>\pgline{E}{F}</code> <code>\stdnparallel \pgline{G}{H}\$</code></pre>	$(EF) \nparallel (GH)$ au lieu de $(EF) \nparallel (GH)$

# 3 Vecteurs

## 3.1 Les écrire

### Exemple 1

<pre><code>\vect{ABCDEF}\$</code> , <code>\vect*{e}{rot}\$</code> ou <code>\vect{e_{rot}}\$</code></pre>	$\overrightarrow{ABCDEF}$ , $\vec{e}_{rot}$ ou $\overrightarrow{e_{rot}}$
--	---

### Exemple 2

<pre><code>\vect{i}\$</code> ou <code>\vect*{j}{2}\$</code></pre>	$\vec{i}$ ou $\vec{j}_2$
---	--------------------------

## 3.2 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}}`

$$\|\vec{i}\| = \|\vec{i}\|$$

$$\left\|\frac{2}{7}\vec{e}_k\right\| = \left\|\frac{2}{7}\vec{e}_k\right\|$$

**Remarque.** Le code L<sup>A</sup>T<sub>E</sub>X pour des doubles barres extensibles ou non vient directement de ce message : <https://tex.stackexchange.com/a/43009/6880>.

### 3.3 Produit scalaire

#### Exemple 1 – Version longue

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

$$\frac{1}{2}\vec{u} \cdot \vec{v}$$

#### Exemple 2 – Écriture « à la physicienne »

Dans l'exemple suivant, l'option **r** est pour **r**-after soit « *chevron* » en anglais, et dans **sr** le **s** est pour **s**-mall soit « *petit* » en anglais. Les physiciens aiment bien cette notation.

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

$$\left\langle \frac{1}{2}\vec{u} \mid \vec{v} \right\rangle$$

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

$$\langle \frac{1}{2}\vec{u} \mid \vec{v} \rangle$$

#### Exemple 3 – Version courtes mais restrictive

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

`\vdotprod{u}{v}$`

$$\vec{u} \cdot \vec{v}$$

`\vdotprod[r]{u}{v}$` comme

$$\langle \vec{u} \mid \vec{v} \rangle \text{ comme } \langle \vec{u} \mid \vec{v} \rangle$$

`\vdotprod[sr]{u}{v}$`

### 3.4 Produit vectoriel

#### Exemple 1 - Écriture symbolique - Version longue

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

$$\frac{1}{2}\vec{i} \wedge \vec{j}$$

#### Exemple 2 - Écriture symbolique - Version courte mais restrictive

`\vcrossprod{i}{j}$`

$$\vec{i} \wedge \vec{j}$$

### Exemple 3 - Explication des calculs

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

<pre> <math display="block">\begin{aligned} &amp; \\$\text{calccrossprod}\{\text{vect}\{u\}\{x\}\{y\}\{z\}\% \\ &amp; \quad \{\text{vect}\{v\}\{x'\}\{y'\}\{z'\}\} \\ &amp; = \\ &amp; \text{\texttt{vcalccrossprod}} \quad \{u\}\{x\}\{y\}\{z\}\% \\ &amp; \quad \{v\}\{x'\}\{y'\}\{z'\}\\$ \\ &amp; \\$\text{vcalccrossprod}\{AB\}\{x_B - x_A\}\% \\ &amp; \quad \{y_B - y_A\}\% \\ &amp; \quad \{z_B - z_A\}\% \\ &amp; \{CD\}\{x_D - x_C\}\% \\ &amp; \quad \{y_D - y_C\}\% \\ &amp; \quad \{z_D - z_C\}\\$ \end{aligned}</math> </pre>	
---	--

Avec un public averti on peut juste proposer les coordonnées sans les décorations comme ci-après via la version étoilée de `\vcalccrossprod` mais ceci fonctionne aussi avec `\calccrossprod`.

<pre> <math display="block">\begin{aligned} &amp; \\$\text{vcalccrossprod}*\{u\}\{x\}\{y\}\{z\}\% \\ &amp; \quad \{v\}\{x'\}\{y'\}\{z'\}\\$ \end{aligned}</math> </pre>	
---	--

Enfin si les vecteurs vous gênent il suffira d'utiliser l'option `novec` pour `no vec-tor` soit « *pas de vecteur* » en anglais comme ci-après. Ceci fonctionne aussi pour la macro `\calccrossprod`. 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!

<pre> <math display="block">\begin{aligned} &amp; \\$\text{vcalccrossprod}[\text{novec}]\{u\}\{x\}\{y\}\{z\}\% \\ &amp; \quad \{v\}\{x'\}\{y'\}\{z'\}\} \\ &amp; = \\ &amp; \text{\texttt{vcalccrossprod}}*[\text{novec}]\{u\}\{x\}\{y\}\{z\}\% \\ &amp; \quad \{v\}\{x'\}\{y'\}\{z'\}\\$ \end{aligned}</math> </pre>	
---	--

### Exemple 4 - Les coordonnées « détaillées »

Pour avoir le détail directement dans des coordonnées vous pouvez faire appel à la macro `\coordcrossprod` où le préfixe `coord` fait référence à `coord-onnée`<sup>1</sup>. On peut utiliser des options pour choisir certains paramètres de mise en forme.

1. En coulisse on utilise la macro `\coord` présentée dans la section 4.1 page ??.

<code>\$\coordcrossprod{\dfrac{1}{2}x}{y}{z }%</code> <code>{x'}{y'}{z'}\$</code>	$\left( y z' - z y' ; z x' - \frac{1}{2} x z' ; \frac{1}{2} x y' - y x' \right)$
<code>\$\coordcrossprod[vb]%</code> <code>{\dfrac{1}{2}x}{y}{z }%</code> <code>{x'}{y'}{z'}\$</code>	$\begin{bmatrix} y z' - z y' \\ z x' - \frac{1}{2} x z' \\ \frac{1}{2} x y' - y x' \end{bmatrix}$
<code>\$\coordcrossprod[sp,c]%</code> <code>{\dfrac{1}{2}x}{y}{z }%</code> <code>{x'}{y'}{z'}\$</code>	$(y \cdot z' - z \cdot y' ; z \cdot x' - \frac{1}{2} x \cdot z' ; \frac{1}{2} x \cdot y' - y \cdot x')$

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. **s** tout seul demande d'utiliser un espace pour séparer les facteurs de chaque produit.
6. **t** tout seul demande d'utiliser `\times` comme opérateur de multiplication.
7. **c** tout seul demande d'utiliser `\cdot` comme opérateur de multiplication.

On peut indiquer des options vis à vis du mode vertical ou horizontal avec des délimiteurs extensibles ou non éventuellement, ou bien sur le symbole pour les produits. On peut aussi combiner deux de ces types de choix en les séparant par une virgule ce qui fait un total de  $6 \times 3 = 18$  combinaisons possibles. La valeur par défaut est **p,s**.

**Attention !** Les produits sont produits 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.

<code>\$\coordcrossprod[vb]%</code> <code>{x_B - x_A}{y_B - y_A}%</code> <code>{z_B - z_A}{x_D - x_C}%</code> <code>{y_D - y_C}{z_D - z_C}\$</code>	$\begin{bmatrix} y_B - y_A z_D - z_C - z_B - z_A y_D - y_C \\ z_B - z_A x_D - x_C - x_B - x_A z_D - z_C \\ x_B - x_A y_D - y_C - y_B - y_A x_D - x_C \end{bmatrix}$
--	---

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.

<code>\$\coordcrossprod[vb]%</code> <code>{(x_B - x_A)}{(y_B - y_A)}%</code> <code>{(z_B - z_A)}{(x_D - x_C)}%</code> <code>{(y_D - y_C)}{(z_D - z_C)}\$</code>	$\begin{bmatrix} (y_B - y_A)(z_D - z_C) - (z_B - z_A)(y_D - y_C) \\ (z_B - z_A)(x_D - x_C) - (x_B - x_A)(z_D - z_C) \\ (x_B - x_A)(y_D - y_C) - (y_B - y_A)(x_D - x_C) \end{bmatrix}$
--	---

### 3.5 Plan – Déterminant de deux vecteurs

#### Exemple 1 – Version décorée

Dans l'exemple suivant, le préfixe **calc** est pour **calc**-uler.

<pre> <math>\backslash\calcdetplane{\vect{u}}{x }{y }%</math> <math>\backslash\calcdetplane{\vect{v}}{x'}{y'}%</math> ou <math>\backslash\calcdetplane{\vect{AB}}%</math> <math>\backslash\calcdetplane{\vect{CD}}%</math> </pre>	$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix} \text{ ou } \begin{vmatrix} \overrightarrow{AB} & \overrightarrow{CD} \\ x_B - x_A & x_D - x_C \\ y_B - y_A & y_D - y_C \end{vmatrix}$
---	---

## Exemple 2 – Version non décorée

<pre> <math>\backslash\calcdetplane*{\vect{u}}{x }{y }%</math> <math>\backslash\calcdetplane*{\vect{v}}{x'}{y'}%</math> </pre>	$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$
--	---

## Exemple 3 – Rédaction raccourcie pour les vecteurs

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

<pre> <math>\backslash\vcaldetplane{u}{x }{y }%</math> <math>\backslash\vcaldetplane{v}{x'}{y'}%</math> = <math>\backslash\vcaldetplane*{u}{x }{y }%</math> <math>\backslash\vcaldetplane*{v}{x'}{y'}%</math> </pre>	$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix} = \begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$
--	---

## Exemple 4 – Versions sans les vecteurs

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

<pre> <math>\backslash\vcaldetplane[novec]{u}{x }{y }%</math> <math>\backslash\vcaldetplane[novec]{v}{x'}{y'}%</math> = <math>\backslash\vcaldetplane*{u}{x }{y }%</math> <math>\backslash\vcaldetplane*{v}{x'}{y'}%</math> </pre>	$\begin{vmatrix} x & x' \\ y & y' \end{vmatrix} = \begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$
--	---

**Remarque.** Ce qui précède marche aussi avec les macros `\calcdetplane` et `\calcdetplane*`.

## Exemple 5 – Calcul développé

Grâce à l'argument optionnel de `\calcdetplane` ou `\vcaldetplane`, 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`. Même si les vecteurs ne sont pas utilisés pour la mise en forme, on obtient ici une méthode très pratique à l'usage car permettant de faire des copier-coller.



<code>\vcalcdetplane[exp]{u}{x }{y }%</code> <code>{v}{x'}{y'}\$</code>	$xy' - x'y$
<code>\vcalcdetplane[cexp]{u}{x }{y }%</code> <code>{v}{x'}{y'}\$</code>	$x \cdot y' - x' \cdot y$
<code>\vcalcdetplane[tepx]{u}{x }{y }%</code> <code>{v}{x'}{y'}\$</code>	$x \times y' - x' \times y$

**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.

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

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

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

## 4 Géométrie cartésienne

### 4.1 Coordonnées

#### Exemple 1 – Des coordonnées seules

lymath 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 L<sup>A</sup>T<sub>E</sub>X plus sémantique*). Commençons par les écritures horizontales où vous noterez l'utilisation de | pour séparer les coordonnées dont le nombre peut être quelconque.

<code>\coord {\dfrac{1}{3}   -4   0}\$</code> ou <code>\coord[sp]{\dfrac{1}{3}   -4   0}\$</code>	$\left(\frac{1}{3}; -4; 0\right)$ ou $(\frac{1}{3}; -4; 0)$
<code>\coord[b]{\dfrac{1}{3}   -4   0}\$</code> ou <code>\coord[sb]{\dfrac{1}{3}   -4   0}\$</code>	$\left[\frac{1}{3}; -4; 0\right]$ ou $[\frac{1}{3}; -4; 0]$

Il existe en plus deux versions verticales.

<code>\coord[vp]{3   -4}\$</code> ou <code>\coord[vb]{3   -4}\$</code>	$\begin{pmatrix} 3 \\ -4 \end{pmatrix}$ 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}$` ou  
`\pcoord*{\Sigma}{7 | 9 | 8}$`

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

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

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

$A[3; -4; 0; -1]$  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  $\frac{1}{2}\vec{u}(3; -4)$   
 $\vec{u}\begin{pmatrix} 3 \\ -4 \end{pmatrix}$  ou  $\frac{1}{2}\vec{u}\begin{pmatrix} 3 \\ -4 \end{pmatrix}$

## 4.2 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}}$`

$(O; I, J)$

### Exemple 2 – La méthode basique en version étoilée

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

```

 $\backslash axes{\backslash pt{0} \%$ 
 $\backslash \dfrac{7}{3} \backslash vect{i} \%$ 
 $\backslash \vect{j}}\$$ 
ou
 $\backslash axes*{\backslash pt{0} \%$ 
 $\backslash \dfrac{7}{3} \backslash vect{i} \%$ 
 $\backslash \vect{j}}\$$ 

```

$(O; \frac{7}{3} \vec{i}, \vec{j})$  ou  $(O; \frac{7}{3} \vec{i}, \vec{j})$

### 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  $n > 0$ .

```

 $\backslash axes{\backslash pt{0} \%$ 
 $\backslash \vect*{i}{1} \%$ 
 $\backslash \vect*{i}{2} \%$ 
 $\backslash \vect*{i}{3} \%$ 
 $\backslash \dots \%$ 
 $\backslash \vect*{i}{9} \%$ 
 $\backslash \vect*{i}{10} \%$ 
 $\backslash \vect*{i}{11} \%$ 
 $\backslash \vect*{i}{12}}\$$ 

```

$(O; \vec{i}_1, \vec{i}_2, \vec{i}_3, \dots, \vec{i}_9, \vec{i}_{10}, \vec{i}_{11}, \vec{i}_{12})$

### Exemple 4 – Repère affine

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

```

 $\backslash paxes{0 | I | J | K}\$$ 
au lieu de
 $\backslash axes{\backslash pt{0} \%$ 
 $\backslash \pt{I} | \backslash \pt{J} | \backslash \pt{K}}\$$ 

```

$(O; I, J, K)$  au lieu de  $(O; I, J, K)$

### Exemple 5 – Repère vectoriel (méthode 1)

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

```

 $\backslash vaxes{\backslash pt{0} | i | j}\$$ 
au lieu de
 $\backslash axes{\backslash pt{0} | \backslash vect{i} | \backslash vect{j}}\$$ 

```

$(O; \vec{i}, \vec{j})$  au lieu de  $(O; \vec{i}, \vec{j})$

### 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.

```

 $\backslash pvaxes{0 | i | j}\$$ 
au lieu de
 $\backslash axes{\backslash pt{0} | \backslash vect{i} | \backslash vect{j}}\$$ 

```

$(O; \vec{i}, \vec{j})$  au lieu de  $(O; \vec{i}, \vec{j})$

## 5 Arcs circulaires

### Exemple 1

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

$\widehat{ABCDEF}$  ,  $\widehat{A_{rot}}$  ou  $\widehat{A_{rot}}$

### Exemple 2

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

$\widehat{i}$  ou  $\widehat{j_2}$

## 6 Angles

### 6.1 Angles géométriques « intérieurs »

#### Exemple 1

`\anglein {ABCDEF}$`  
`\anglein* {A}{rot}$`  
`\anglein {A_{rot}}$`

$\widehat{ABCDEF}$   
 $\widehat{A_{rot}}$   
 $\widehat{A_{rot}}$

#### Exemple 2 – Cacher les points du i et du j

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

$\widehat{i}$  et  $\widehat{j_2}$

### 6.2 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}}$`  
`\angleorient[sp]{\dfrac{1}{2} \vect{i}}%`  
`{\vect{j}}$`

$\left( \frac{1}{2} \vec{i} ; \vec{j} \right)$   
 $\left( \frac{1}{2} \vec{i} ; \vec{j} \right)$

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

```

$(\vec{i}; \vec{j})$  comme  $(\vec{i}; \vec{j})$

## 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
 $\vangleorient[sh]{i}{j}$ 

```

$\widehat{\left(\frac{1}{2} \vec{i}; \vec{j}\right)}$   
 $\widehat{\left(\frac{1}{2} \vec{i}; \vec{j}\right)}$   
 $\widehat{(\vec{i}; \vec{j})}$  comme  $\widehat{(\vec{i}; \vec{j})}$

## 7 Historique

Nous ne donnons ici qu'un très bref historique récent <sup>2</sup> 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**.

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

---

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

## 8 Toutes les fiches techniques

### 8.1 Introduction

### 8.2 Points et lignes

#### 8.2.1 Points

`\pt <macro> (1 Argument)`

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

---

`\pt* <macro> (2 Arguments)`

— 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.

#### 8.2.2 Lignes

`\gline <macro> [1 Option] (2 Arguments)` où  $g = g\text{-eometry}$

`\pgline <macro> [1 Option] (2 Arguments)` où  $p = p\text{-oint}$  et  $g = g\text{-eometry}$

— Option : pour indiquer les parenthèses ou crochets à utiliser, les valeurs possibles étant 0, valeur par défaut, C, CO et OC.

— Argument 1 : le 1<sup>er</sup> point géométrique.

— Argument 2 : le 2<sup>e</sup> point géométrique.

---

`\hgline <macro> (2 Arguments)` où  $h = h\text{-alf}$  et  $g = g\text{-eometry}$

`\phgline <macro> (2 Arguments)` où  $p = p\text{-oint}$ ,  $h = h\text{-alf}$  et  $g = g\text{-eometry}$

`\segment <macro> (2 Arguments)`

`\psegment <macro> (2 Arguments)` où  $p = p\text{-oint}$

— Argument 1 : le 1<sup>er</sup> point géométrique.

— Argument 2 : le 2<sup>e</sup> point géométrique.

---

#### 8.2.3 Droites parallèles ou non

`\parallel <macro> (Sans argument)`

`\nparallel <macro> (Sans argument)`

`\stdparallel <macro> (Sans argument)`

`\stdnparallel <macro> (Sans argument)`

### 8.3 Vecteurs

#### 8.3.1 Vecteurs

`\vect <macro> (1 Argument)`

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

---

`\vect*` <macro> (2 Arguments)

- 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.

### 8.3.2 Norme

`\norm` <macro> [1 Option] (1 Argument)

- 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` <macro> (1 Argument)

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

### 8.3.3 Produit scalaire

`\dotprod` <macro> [1 Option] (2 Arguments)

- 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. `r` : écriture « à la physicienne » avec des chevrons extensibles.
    3. `sr` : écriture « à la physicienne » avec des chevrons non extensibles.
  - Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
  - Argument 2: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- 

`\vdotprod` <macro> [1 Option] (2 Arguments) où `v` = v-ector

- Option: voir les explications précédentes données pour `\dotprod`.
- Argument 1: le nom du 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: le nom du 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.

### 8.3.4 Produit vectoriel

`\crossprod` <macro> (2 Arguments)

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
  - Argument 2: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- 

`\vcrossprod` <macro> (2 Arguments) où `v` = v-ector

- Argument 1: le nom du 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
  - Argument 2: le nom du 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.
-



`\calccrossprod <macro> (8 Arguments)` où `calc = calc-ulate`  
`\calccrossprod* <macro> (8 Arguments)` où `calc = calc-ulate`

- Argument 1 : le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2..4 : les coordonnées du 1<sup>er</sup> vecteur.
- Argument 5 : le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 6..8 : les coordonnées du 2<sup>e</sup> vecteur.

---

`\vcalccrossprod <macro> (8 Arguments)` où `calc = calc-ulate` et `v = v-ector`  
`\vcalccrossprod* <macro> (8 Arguments)` où `calc = calc-ulate` et `v = v-ector`

- Argument 1 : le 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2..4 : les coordonnées du 1<sup>er</sup> vecteur.
- Argument 5 : le 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.
- Argument 6..8 : les coordonnées du 2<sup>e</sup> vecteur.

---

`\coordcrossprod <macro> [1 Option] (6 Arguments)` où `coord = coord-inate`

— Option : la valeur par défaut est `p,s`. Voici les différentes valeurs possibles pour la mise en forme des coordonnées uniquement (*voir la section 8.4.1 page 18*).

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.

Pour les produits, voici ce qui est proposé.

1. `s` : un espace pour séparer les facteurs de chaque produit.
2. `t` : `\times` comme opérateur de multiplication.
3. `c` : `\cdot` comme opérateur de multiplication.

On peut combiner deux types de choix en les séparant par une virgule comme dans `p,s` la valeur par défaut.

- Argument 1..3 : les coordonnées du 1<sup>er</sup> vecteur.
- Argument 4..6 : les coordonnées du 2<sup>e</sup> vecteur.

### 8.3.5 Plan – Déterminant de deux vecteurs

`\calcdetplane <macro> [1 Option] (6 Arguments)` où `calc = calc-ulate`  
`\calcdetplane* <macro> [1 Option] (6 Arguments)` où `calc = calc-ulate`

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

1. `vec` : les vecteurs sont affichés si besoin.
2. `novect` : les vecteurs ne sont jamais affichés.
3. `exp` : ceci demande d'afficher une formule développée en utilisant un espace pour séparer les facteurs de chaque produit.
4. `cexp` : comme `exp` mais avec le symbole  $\cdot$  obtenu via `\cdot`.
5. `texp` : comme `exp` mais avec le symbole  $\times$ .

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2: la 1<sup>re</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 3: la 2<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 4: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 5: la 1<sup>re</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 6: la 2<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.

---

`\vcalcdetplane` <macro> [1 Option] (6 Arguments) où `calc` = `calc-ulate` et `v` = `v-ector`  
`\vcalcdetplane*` <macro> [1 Option] (6 Arguments) où `calc` = `calc-ulate` et `v` = `v-ector`

- Option: voir les indications données pour les macros `\calcdetplane` et `\calcdetplane*`.
- Argument 1..6: voir les indications données pour les macros `\calcdetplane` et `\calcdetplane*`.

## 8.4 Géométrie cartésienne

### 8.4.1 Coordonnées

`\coord` <macro> [1 Option] (1 Argument)

- 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` <macro> [1 Option] (2 Arguments) où `v` = `v-ertical`

- 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*` <macro> [1 Option] (2 Arguments) où `v` = `v-ertical`

- 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` <macro> [1 Option] (2 Arguments) où `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* <macro> [1 Option] (2 Arguments)` où  $v = v\text{-ertical}$

— **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`.

### 8.4.2 Repères

`\axes <macro> (1 Argument)`

`\axes* <macro> (1 Argument)`

— **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 <macro> (1 Argument)` où  $p = p\text{-oint}$

— **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 <macro> (1 Argument)` où  $v = v\text{-ector}$

— **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 <macro> (3 Arguments)` où  $p v = p + v$

— **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.

### 8.4.3 Arcs circulaires

`\circarc <macro> (1 Argument)` où  $circ = circ\text{-ular}$

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

---

`\circarc* <macro> (2 Arguments)` où  $circ = circ\text{-ular}$

— **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.

#### 8.4.4 Angles géométriques « intérieurs »

`\anglein <macro> (1 Argument)` où `in = in-terior`

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

---

`\anglein* <macro> (2 Arguments)` où `in = in-terior`

— **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.

#### 8.4.5 Angles orientés de vecteurs

`\angleorient <macro> [1 Option] (2 Arguments)`

— **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 <macro> [1 Option] (2 Arguments)` où `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`.