

# Le package `tnsgeo` : pour la géométrie élémentaire

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

Version 0.5.0-beta développée et testée sur Mac OS X.

Christophe BAL

2021-02-26

---

## Table des matières

|       |                                       |    |
|-------|---------------------------------------|----|
| I.    | Introduction                          | 3  |
| II.   | Beta-dépendance                       | 3  |
| III.  | Packages utilisés                     | 3  |
| IV.   | Ensembles géométriques                | 3  |
| V.    | Utiliser des unités S.I.              | 3  |
| VI.   | Points et lignes                      | 3  |
|       | 1. Points                             | 3  |
|       | 2. Lignes                             | 4  |
|       | 3. Droites parallèles ou non          | 4  |
| VII.  | Vecteurs                              | 5  |
|       | 1. Les écrire                         | 5  |
|       | 2. Norme                              | 5  |
|       | 3. Produit scalaire                   | 5  |
|       | 4. 3D – Produit vectoriel             | 7  |
|       | i. Écriture symbolique                | 7  |
|       | ii. Explication du mode de calcul     | 7  |
|       | 5. 2D – Déterminant de deux vecteurs  | 9  |
| VIII. | Géométrie cartésienne                 | 11 |
|       | 1. Coordonnées                        | 11 |
|       | 2. Nommer un repère                   | 12 |
| IX.   | Arcs circulaires                      | 13 |
| X.    | Angles                                | 13 |
|       | 1. Angles géométriques « intérieurs » | 13 |
|       | 2. Angles orientés de vecteurs        | 14 |
| XI.   | Historique                            | 15 |
| XII.  | Toutes les fiches techniques          | 16 |
|       | 1. Points et lignes                   | 16 |
|       | i. Points                             | 16 |
|       | ii. Lignes                            | 16 |

|                                       |    |
|---------------------------------------|----|
| iii. Droites parallèles ou non        | 16 |
| 2. Vecteurs                           | 16 |
| i. Les écrire                         | 16 |
| ii. Norme                             | 17 |
| iii. Produit scalaire                 | 17 |
| iv. 3D – Produit vectoriel            | 17 |
| v. 2D – Déterminant de deux vecteurs  | 19 |
| 3. Géométrie cartésienne              | 19 |
| i. Coordonnées                        | 19 |
| ii. Nommer un repère                  | 20 |
| 4. Arcs circulaires                   | 21 |
| 5. Angles                             | 21 |
| i. Angles géométriques « intérieurs » | 21 |
| ii. Angles orientés de vecteurs       | 21 |

---

# 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` utilise les packages suivants sans aucun scrupule.

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

## 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 qui seront françaises**<sup>1</sup> 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 qu'ils facilitent la relecture du code.

|  |   |
|--|---|
| <pre><code>\$\ang{180} = \SI{\pi}{\radian}\$ ,<br/>\$\SI{1}{km} = \SI{1e3}{m}\$ avec aussi<br/>\$\num{123 000} = \num{1230 * 100}\$</code></pre> | $180^\circ = \pi \text{ rad}$ , $1 \text{ km} = 1 \times 10^3 \text{ m}$ avec aussi<br>$123\,000 = 1230 \times 100$ |
|--|---|

## VI. Points et lignes

### 1. Points

Exemple 1 – Sans indice

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

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

## 2. Lignes

### Exemple 1 – Les droites

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

|   |                           |
|---|---------------------------|
| <code>\$\gline{A}{B}\$</code> ,<br><code>\$\gline{\pt{A}}{\pt{B}}\$</code> ou<br><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> ,<br><code>\$\segment{\pt{A}}{\pt{B}}\$</code> ou<br><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> ,<br><code>\$\hgline{\pt{A}}{\pt{B}}\$</code> ou<br><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.

|   |                           |
|---|---------------------------|
| <code>\$\gline[OC]{A}{B}\$</code> ,<br><code>\$\gline[OC]{\pt{A}}{\pt{B}}\$</code> ou<br><code>\$\pgline[OC]{A}{B}\$</code> | $(AB]$ , $(AB]$ 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 L<sup>A</sup>T<sub>E</sub>X.

```

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

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

```

$(AB) \parallel (CD)$  au lieu de  $(AB) \parallel (CD)$   
 $(EF) \nparallel (GH)$  au lieu de  $(EF) \nparallel (GH)$

## VII. Vecteurs

### 1. Les écrire

#### Exemple 1

```

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

```

$\overrightarrow{UnNomLong}$  ,  $\vec{e}_{rot}$  ou  $\overrightarrow{e_{rot}}$

#### Exemple 2 – Sans point disgracieux

```

 $\vect{i}$  ou
 $\vect*{j}_2$ 

```

$\vec{i}$  ou  $\vec{j}_2$

#### Exemple 3 – Se passer de la macro \pt

```

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

```

$\overrightarrow{AB} = \overrightarrow{AB}$ .

### 2. Norme

```

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

 $\norm{\dfrac{2}{7} \vect*{e}_k}$ 
=  $\norm*{\dfrac{2}{7} \vect*{e}_k}$ 

```

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

**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. Produit scalaire

Les 1<sup>ers</sup> 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

|  |                                     |
|--|-------------------------------------|
| <code>\$\dotprod{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code> | $\frac{1}{2} \vec{u} \cdot \vec{v}$ |
|--|-------------------------------------|

### 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.

|   |                                     |
|---|-------------------------------------|
| <code>\$\dotprod[b]{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code> | $\frac{1}{2} \vec{u} \cdot \vec{v}$ |
|---|-------------------------------------|

### 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.

|  |   |
|--|---|
| <code>\$\dotprod[p]{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code>  | $\left( \frac{1}{2} \vec{u} \mid \vec{v} \right)$ |
| <code>\$\dotprod[sp]{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code> | $( \frac{1}{2} \vec{u} \mid \vec{v} )$            |

### 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.

|  |   |
|--|---|
| <code>\$\dotprod[r]{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code>  | $\left\langle \frac{1}{2} \vec{u} \mid \vec{v} \right\rangle$ |
| <code>\$\dotprod[sr]{\dfrac{1}{2} \vect{u}}{\vect{v}}\$</code> | $\langle \frac{1}{2} \vec{u} \mid \vec{v} \rangle$            |

### 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.

|   |   |
|---|---|
| <code>\$\vdotprod {u}{v}\$<br/>= <code>\vdotprod[b]{u}{v}\$</code></code>   | $\vec{u} \cdot \vec{v} = \vec{u} \bullet \vec{v}$                 |
| <code>\$\vdotprod[r]{u}{v}\$<br/>= <code>\vdotprod[p]{u}{v}\$</code></code> | $\langle \vec{u} \mid \vec{v} \rangle = ( \vec{u} \mid \vec{v} )$ |

## 4. 3D – Produit vectoriel

### i. Écriture symbolique

#### Exemple 1 – Version classique en France

|   |                                      |
|---|--------------------------------------|
| <code><math>\backslash</math>crossprod{\dfrac{1}{2} \vect{i}}%</code><br><code><math>\backslash</math>crossprod{\vect{j}}%</code> | $\frac{1}{2} \vec{i} \wedge \vec{j}$ |
|---|--------------------------------------|

#### Exemple 2 – Version alternative

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

|   |                                      |
|---|--------------------------------------|
| <code><math>\backslash</math>crossprod[t]{\dfrac{1}{2} \vect{i}}%</code><br><code><math>\backslash</math>crossprod[t]{\vect{j}}%</code> | $\frac{1}{2} \vec{i} \times \vec{j}$ |
|---|--------------------------------------|

#### Exemple 3 – Version courte mais restrictive

|   |  |
|---|--|
| <code><math>\backslash</math>vcrossprod {i}{j}%</code> ou<br><code><math>\backslash</math>vcrossprod[t]{i}{j}%</code> | $\vec{i} \wedge \vec{j}$ ou $\vec{i} \times \vec{j}$ |
|---|--|

### 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.

|   |   |
|---|---|
| <code><math>\backslash</math>calccrossprod{\vect{u}}{x}{y}{z}%</code><br><code><math>\backslash</math>calccrossprod{\vect{v}}{x'}{y'}{z'}%</code><br>ou<br><code><math>\backslash</math>vcalccrossprod{AB}{x_B - x_A}%</code><br><code><math>\backslash</math>vcalccrossprod{AB}{y_B - y_A}%</code><br><code><math>\backslash</math>vcalccrossprod{AB}{z_B - z_A}%</code><br><code><math>\backslash</math>vcalccrossprod{CD}{x_D - x_C}%</code><br><code><math>\backslash</math>vcalccrossprod{CD}{y_D - y_C}%</code><br><code><math>\backslash</math>vcalccrossprod{CD}{z_D - z_C}%</code> | $\begin{array}{c} \vec{u} \quad \vec{v} \\ \left  \begin{array}{cc} x & x' \\ y & y' \\ z & z' \end{array} \right  \end{array}$ ou $\begin{array}{c} \overrightarrow{AB} \quad \overrightarrow{CD} \\ \left  \begin{array}{cc} x_B - x_A & x_D - x_C \\ y_B - y_A & y_D - y_C \\ z_B - z_A & z_D - z_C \end{array} \right  \end{array}$ |
|---|---|

**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'}$

```

$$\begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \\ z & z' \\ x & x' \end{array} \right| \end{array} = \begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \\ z & z' \\ x & x' \end{array} \right| \end{array} = \begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \\ z & z' \\ x & x' \end{array} \right| \end{array}$$

### 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<sup>2</sup>.

```

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

```

$$\begin{array}{c} x \quad x' \\ y \quad y' \\ z \quad z' \\ x \quad x' \end{array} = \begin{array}{c} x \quad x' \\ y \quad y' \\ z \quad z' \\ x \quad x' \end{array}$$

### Exemple 4 – Les coordonnées « développées »

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

```

$ \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'}$

```

$$\left( yz' - zy'; zx' - \frac{1}{2}xz'; \frac{1}{2}xy' - yx' \right)$$

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

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

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. De nouveau 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.

|  |   |
|--|---|
| <pre> <math>\backslash</math>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}\$ </pre> | $\begin{bmatrix} y_B - y_A z_D - z_C - z_B - z_A y_D - y_C \\ z_B - z_A x_D - x_C - x_B - x_A z_D - z_C \\ x_B - x_A y_D - y_C - y_B - y_A x_D - x_C \end{bmatrix}$ |
|--|---|

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.

|  |   |
|--|---|
| <pre> <math>\backslash</math>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)}\$ </pre> | $\begin{bmatrix} (y_B - y_A)(z_D - z_C) - (z_B - z_A)(y_D - y_C) \\ (z_B - z_A)(x_D - x_C) - (x_B - x_A)(z_D - z_C) \\ (x_B - x_A)(y_D - y_C) - (y_B - y_A)(x_D - x_C) \end{bmatrix}$ |
|--|---|

## 5. 2D – 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.

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

**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.

|   |  |
|---|--|
| <pre> <math>\backslash</math>vcalcdetplane[arrows]{u}{x}{y}%                 {v}{x'}{y'}\$ </pre> | $\begin{array}{cc} \vec{u} & \vec{v} \\ \left  \begin{array}{cc} x & x' \\ y & y' \end{array} \right  \end{array}$ |
|---|--|

### Exemple 3 – Version décorée avec une croix non fléchée

|  |  |
|--|--|
| <pre> <math>\backslash</math>vcalcdetplane[cross]{u}{x}{y}%                 {v}{x'}{y'}\$ </pre> | $\begin{array}{cc} \vec{u} & \vec{v} \\ \left  \begin{array}{cc} x & x' \\ y & y' \end{array} \right  \end{array}$ |
|--|--|

### Exemple 4 – Version non décorée

|  |   |
|--|---|
| <code>\$\vcaldetplane[nodeco]{u}{x}{y}%</code><br><code>{v}{x'}{y'}\$</code> | $\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$ |
|--|---|

### Exemple 5 – Sans les vecteurs

|  |   |
|--|---|
| <code>\$\vcaldetplane[novec]%</code><br><code>{u}{x}{y} {v}{x'}{y'}</code><br><code>= \vcaldetplane[novec,cross]%</code><br><code>{u}{x}{y} {v}{x'}{y'}\$</code> | $\begin{vmatrix} x & x' \\ y & y' \end{vmatrix} = \begin{vmatrix} x & x' \\ y & y' \end{vmatrix}$ |
|--|---|

### Exemple 6 – Calcul développé

Grâce à l'argument optionnel de `\caldetplane` ou de `\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`<sup>5</sup>.

|  |   |
|--|---|
| <code>\$\vcaldetplane[exp]{u}{x}{y}%</code><br><code>{v}{x'}{y'}\$</code>  |   |
| <code>\$\vcaldetplane[cexp]{u}{x}{y}%</code><br><code>{v}{x'}{y'}\$</code> | $\begin{aligned} & x y' - x' y \\ & x \cdot y' - x' \cdot y \\ & x \times y' - x' \times y \end{aligned}$ |
| <code>\$\vcaldetplane[texp]{u}{x}{y}%</code><br><code>{v}{x'}{y'}\$</code> |   |

**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>\$\vcaldetplane[exp]{AB}%</code><br><code>{x_B - x_A}%</code><br><code>{y_B - y_A}%</code><br><code>{CD}%</code><br><code>{x_D - x_C}%</code><br><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 main. Ce genre de situation n'est pas rare dans la vraie vie mathématique.

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

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.

## 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  $L^A T_E X$  plus sémantique*). Commençons par les écritures horizontales où vous noterez l'utilisation de `|` pour séparer les coordonnées dont le nombre peut être quelconque.

|  |   |
|--|---|
| $\$ \backslash \text{coord} \{ \backslash \text{dfrac} \{ 1 \} \{ 3 \} \mid -4 \mid 0 \} \$$ ou<br>$\$ \backslash \text{coord} [\text{sp}] \{ \backslash \text{dfrac} \{ 1 \} \{ 3 \} \mid -4 \mid 0 \} \$$            | $\left( \frac{1}{3} ; -4 ; 0 \right)$ ou $(\frac{1}{3} ; -4 ; 0)$ |
| $\$ \backslash \text{coord} [\text{b}] \{ \backslash \text{dfrac} \{ 1 \} \{ 3 \} \mid -4 \mid 0 \} \$$ ou<br>$\$ \backslash \text{coord} [\text{sb}] \{ \backslash \text{dfrac} \{ 1 \} \{ 3 \} \mid -4 \mid 0 \} \$$ | $\left[ \frac{1}{3} ; -4 ; 0 \right]$ ou $[\frac{1}{3} ; -4 ; 0]$ |

Il existe en plus deux versions verticales.

|   |  |
|---|--|
| $\$ \backslash \text{coord} [\text{vp}] \{ 3 \mid -4 \} \$$ ou<br>$\$ \backslash \text{coord} [\text{vb}] \{ 3 \mid -4 \} \$$ | $\begin{pmatrix} 3 \\ -4 \end{pmatrix}$ 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*`.

|   |   |
|---|---|
| $\$ \backslash \text{pcoord} \{ \text{A} \} \{ 3 \mid -4 \mid 0 \mid -1 \} \$$ ou<br>$\$ \backslash \text{pcoord} * \{ \backslash \text{Sigma} \} \{ 7 \mid 9 \mid 8 \} \$$ | $A(3 ; -4 ; 0 ; -1)$ ou $\Sigma(7 ; 9 ; 8)$ |
|---|---|

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

|   |   |
|---|---|
| $\$ \backslash \text{pcoord} [\text{b}] \{ \text{A} \} \{ 3 \mid -4 \mid 0 \mid -1 \} \$$ ou<br>$\$ \backslash \text{pcoord} * [\text{b}] \{ \backslash \text{Sigma} \} \{ 7 \mid 9 \mid 8 \} \$$ | $A[3 ; -4 ; 0 ; -1]$ 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.

```

 $\backslash\text{vcoord}\{u\}\{3 \mid -4\}$  ou
 $\backslash\text{vcoord}*\{\dfrac{1}{2}\} \backslash\text{vect}\{u\}\%$ 
 $\{3 \mid -4\}$ 

 $\backslash\text{vcoord}\{vp\}\{u\}\{3 \mid -4\}$  ou
 $\backslash\text{vcoord}*\{vp\}\{\dfrac{1}{2}\} \backslash\text{vect}\{u\}\%$ 
 $\{3 \mid -4\}$ 

```

$$\vec{u}(3; -4) \text{ ou } \frac{1}{2}\vec{u}(3; -4)$$

$$\vec{u}\begin{pmatrix} 3 \\ -4 \end{pmatrix} \text{ ou } \frac{1}{2}\vec{u}\begin{pmatrix} 3 \\ -4 \end{pmatrix}$$

## 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*).

```

 $\backslash\text{axes}\{\text{pt}\{0\} \%$ 
 $\mid \backslash\text{pt}\{I\} \mid \backslash\text{pt}\{J\}\}$ 

```

$$(O; I, J)$$

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

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

```

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

```

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

### 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  $n > 0$ .

```

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

```

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

### Exemple 4 – Repère affine

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

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

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

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

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

## IX. Arcs circulaires

### Exemple 1

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

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

### Exemple 2

`\circarc{i}` ou  
`\circarc*[j]{2}`

$\hat{i}$  ou  $\hat{j}_2$

## X. Angles

### 1. Angles géométriques « intérieurs »

#### Exemple 1

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

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

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

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

$\widehat{i}$  ou  $\widehat{j}_2$

## 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}}`

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

### 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  $\widehat{(\vec{i} ; \vec{j})}$

## XI. Historique

Nous ne donnons ici qu'un très bref historique récent<sup>6</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`.

**2021-02-26** Nouvelle version mineure **0.5.0-beta**.

- **VECTEUR** : suppression des macros `\colicriteria` et `\vcolicriteria` car elles sont inutiles.
- 

**2020-09-02** Nouvelle version mineure **0.4.0-beta**.

- **VECTEUR** : retour de la macro `\norm*`.
- 

**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`.
    - Plus aucune version étoilée simple ou double pour `\calccrossprod`, `\vcalccrossprod`, `\calcdetplane` et `\vcalcdetplane`.
- 

**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}`

$g = g\text{-eometry}$

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

$p = p\text{-oint}$

— 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 {#1..#2}`

$h = h\text{-alf}$

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

$phg = p + h + g$

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

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

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

— Argument 2 : le 2<sup>e</sup> 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 1<sup>er</sup> point qui sera mis en forme via la macro `\pt`.

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

## ii. Norme

`\norm{#1}`

`\norm*{#1}`

— 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 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[#opt]{#1..#2}`

`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`.

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

`v = v-ector`

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

— 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[#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. `novect` : 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  $\cdot$  obtenu via `\cdot`.
9. `texp` : comme `exp` mais avec le symbole  $\times$ .
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 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.

— Arguments 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`.

— Arguments 6..8: les coordonnées du 2<sup>e</sup> vecteur.

---

`\vcalccrossprod[#opt]{#1..#8}`

`calc = calc-ulate et v = v-ector`

— Argument 1: le 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.

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

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

## v. 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. `novect` : 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  $\cdot$  obtenu via `\cdot`.
9. `texp` : comme `exp` mais avec le symbole  $\times$ .

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 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.

— **Arguments 2..3**: les coordonnées du 1<sup>er</sup> vecteur.

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

— **Arguments 5..6**: les coordonnées du 2<sup>e</sup> vecteur.

---

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

— **Argument 1**: le 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.

— **Argument 4**: le 2<sup>e</sup> 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`.