

# Documentation de `repere.mp`

Olivier PÉAULT\*

8 juillet 2022

## Table des matières

<b>I</b>	<b>Repères et géométrie</b>	<b>2</b>	<b>8</b>	<b>Statistiques et probabilités</b>	<b>22</b>
<b>1</b>	<b>Utilisation du fichier</b>	<b>2</b>	8.1	Boite à moustache . . . . .	22
1.1	Généralités . . . . .	2	8.2	Diagrammes . . . . .	23
1.2	Figures dans un document avec L <sup>A</sup> T <sub>E</sub> X . . . . .	2	8.3	Probabilités . . . . .	24
1.3	Figures dans un document avec LuaL <sup>A</sup> T <sub>E</sub> X . . . . .	2	<b>9</b>	<b>Géométrie</b>	<b>26</b>
<b>2</b>	<b>Repère utilisateur</b>	<b>3</b>	9.1	Polygones . . . . .	26
2.1	Numérotation des figures . . . . .	3	9.2	Cercles et arcs . . . . .	27
2.2	Définition du repère . . . . .	3	9.3	Codage des segments et des angles	28
2.3	Axes . . . . .	3	9.4	Cotes . . . . .	29
2.4	Quadrillages . . . . .	7	9.5	Figures complètes . . . . .	30
2.5	Base . . . . .	8	<b>10</b>	<b>Divers</b>	<b>36</b>
<b>3</b>	<b>Points, vecteurs</b>	<b>9</b>	10.1	Composition des étiquettes . . . .	36
3.1	Points . . . . .	9	10.2	Couleurs . . . . .	37
3.2	Vecteurs . . . . .	10	10.3	Remplissage . . . . .	37
<b>4</b>	<b>Droites, courbes...</b>	<b>11</b>	10.4	Figures pour une présentation . .	38
4.1	Droites . . . . .	11	<b>11</b>	<b>Dessin à main levée avec geometriesyr</b>	<b>39</b>
4.2	Demi-droites . . . . .	11	<b>II</b>	<b>Tableaux et grilles</b>	<b>40</b>
4.3	Courbes et fonctions . . . . .	12	<b>12</b>	<b>Définition et grilles</b>	<b>40</b>
4.4	Placement automatique du nom des courbes . . . . .	14	12.1	Définition du tableau . . . . .	40
4.5	Dérivée et tangentes . . . . .	15	12.2	Grille . . . . .	40
4.6	Interpolation . . . . .	15	12.3	Grille partielle . . . . .	41
<b>5</b>	<b>Suites</b>	<b>18</b>	<b>13</b>	<b>Repérage et cases</b>	<b>41</b>
<b>6</b>	<b>Surfaces</b>	<b>19</b>	13.1	Coordonnées . . . . .	41
6.1	Calcul intégral . . . . .	19	13.2	Placements d'objets dans les cases	43
6.2	Demi-plans . . . . .	20	<b>14</b>	<b>Quelques dessins</b>	<b>44</b>
<b>7</b>	<b>Projections sur les axes</b>	<b>21</b>	<b>15</b>	<b>Exemples</b>	<b>46</b>
7.1	Projetés . . . . .	21	15.1	Mots croisés . . . . .	46
7.2	Intervalles . . . . .	22	15.2	Sudokus . . . . .	47
			15.3	Dames . . . . .	47
			15.4	Bataille navale . . . . .	48
			15.5	Algoréa . . . . .	49

---

\*E-mail : [o.peault@posteo.net](mailto:o.peault@posteo.net)

## Première partie

# Repères et géométrie

## 1 Utilisation du fichier

### 1.1 Généralités

Les macros du fichier `repere.mp` ont pour but de simplifier la création de figures dans un repère du plan avec METAPOST. L'idée de départ est de coller le plus possible aux besoins de l'enseignement secondaire français de mathématiques.

Il est possible d'utiliser `repere` et `geometriesyr` (les macros de Christophe POULAIN pour la géométrie disponibles à l'adresse <http://melusine.eu.org/syracuse/poulecl/macros/>) dans une même figure comme le montre l'exemple page 39.

Le fichier `repere.mp` doit être placé dans un répertoire accessible à METAPOST (Par ex. le répertoire `metapost` du `texmf`). De plus, la ligne `input repere;` doit apparaître dans le document contenant les figures.

Les étiquettes (noms de points, de courbes, de vecteurs...) sont composées automatiquement au format L<sup>A</sup>T<sub>E</sub>X avec le package `latexmp.mp`. Il est donc nécessaire de compiler deux fois les documents. Cette double compilation n'est pas nécessaire si on utilise LuaL<sup>A</sup>T<sub>E</sub>X avec le package `luamplib`.

### 1.2 Figures dans un document avec L<sup>A</sup>T<sub>E</sub>X

Chacun des documents peut être compilé avec la commande `mpost` puis inclus dans un document à l'aide de la commande `\includegraphics` du package `graphicx`.

Certains packages permettent d'écrire du code METAPOST directement dans un document L<sup>A</sup>T<sub>E</sub>X. `repere` est compatible avec, entre autres, `emp` et `mpgraphics`.

### 1.3 Figures dans un document avec LuaL<sup>A</sup>T<sub>E</sub>X

Il est aussi possible d'utiliser LuaL<sup>A</sup>T<sub>E</sub>X avec le package `luamplib`. Il faut alors charger les packages `siunitx` et `esvect` utilisés par `repere`.

#### Figure isolée

lualatex mfigure.tex

```
\documentclass{standalone}
\usepackage{fontspec}
\usepackage{siunitx,esvect}
\usepackage{luamplib}
\mplibnumbersystem{decimal} %Si besoin
\everymplib{input repere;}
\begin{document}
\begin{mplibcode}
  repere(-3,3,1cm,-2,2,1cm);
  draw axes(1,1);
fin;
\end{mplibcode}
\end{document}
```

#### Code embarqué

lualatex monfichier.tex

```
\documentclass{article}
\usepackage{fontspec}
\usepackage{siunitx,esvect}
\usepackage{luamplib}
\mplibnumbersystem{decimal} %Si besoin
\everymplib{verbatimtex \leavevmode etex;
             input repere;}
\begin{document}
Mon texte, mon texte, mon texte

\begin{mplibcode}
  repere(-3,3,1cm,-2,2,1cm);
  draw axes(1,1);
fin;
\end{mplibcode}

Mon texte, mon texte, mon texte
\end{document}
```

## 2 Repère utilisateur

### 2.1 Numérotation des figures

Chaque figure devra débuter par une instruction `repere()` et se terminer par `fin` (voir ci-dessous). Si ces instructions se trouvent en dehors d'un environnement `beginfig()-endfig` la numérotation est automatique :

```
beginfig(2);
repere(...);
<instructions de dessin>
fin;
endfig;
```

La figure porte le numéro 2

```
repere(...);
<instructions de dessin>
fin;
```

La numérotation est automatique. La figure porte le numéro qui suit la figure précédemment dessinée. S'il s'agit de la première, elle porte le numéro 1.

### 2.2 Définition du repère

`repere(xmin,xmax,ux,ymin,ymax,uy,theta)` débute une figure et définit le repère utilisateur : axe des abscisses de `xmin` à `xmax`, unité `ux`, axe des ordonnées de `ymin` à `ymax`, unité `uy` et `theta` est l'angle en degrés entre les axes. Le paramètre `theta` est optionnel. Il est égal à 90 par défaut.

`repere.larg(xmin,xmax,Lx,ymin,ymax,Ly,theta)` définit un repère tel que la largeur totale de la figure produite soit `Lx` et sa hauteur `Ly`.

`repere.orth(xmin,xmax,Lx,ymin,ymax)` définit un repère orthonormé de largeur totale `Lx`.

`interaxes(x,y)` définit les coordonnées du point d'intersection des axes. Par défaut ces coordonnées sont (0,0).

`cadre` chemin fermé qui fait le tour du repère.

`fin` termine la figure et la découpe pour ne garder que la partie limitée par le repère utilisateur.

### 2.3 Axes

#### 2.3.1 Généralités

`axex.pos(grad,val)` axe des abscisses gradué avec un pas de `grad` et étiqueté avec un pas de `val`.

Si `grad` est négatif ou nul, l'axe n'est pas gradué et si `val` est négatif ou nul, l'axe n'est pas étiqueté.

`pos` est un paramètre optionnel qui désigne la position (au sens de METAPOST : `rt`, `urt`, `top`, `ulft`, `lft`, `llft`, `bot` ou `lrt`) des étiquettes. `pos` peut être omis, la valeur par défaut est `bot`.

Les étiquettes qui ne sont pas entièrement à l'intérieur du cadre ne sont pas dessinées.

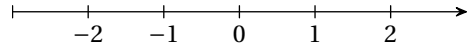
`axey.pos(grad,val)` axe des ordonnées. La valeur par défaut de `pos` est `lft`.

Au niveau de l'intersection des axes, les étiquettes sont tracées à la position `pos` si l'abscisse est différente de l'ordonnée ou si un seul axe est tracé. Dans le cas contraire, une seule étiquette est tracée pour les deux axes à une position « intermédiaire » (pour `axex.bot` et `axey.lft`, on obtient la position `llft`)

`axes.pos(grad,val)` figure formée par les deux axes gradués avec le même pas `grad` et étiquetés avec le même pas `val`. `pos` désigne la position de l'étiquette de l'intersection des axes, sa valeur par défaut est `llft`. La position des étiquettes des axes est définie à partir de `pos` (pour `urt` on obtient `top` pour l'axe des abscisses et `rt` pour l'axe des ordonnées).

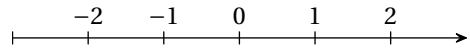
### Exemple 1

```
repere(-3,3,1cm,-1,1,1cm);  
draw axex(1,1);  
fin;
```



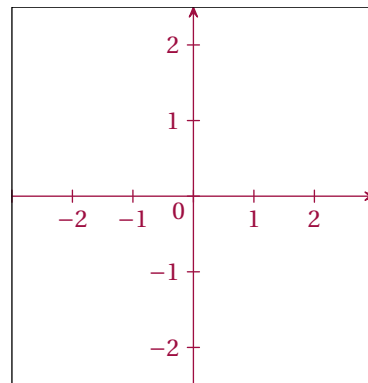
### Exemple 2

```
repere(-3,3,1cm,-1,1,1cm);  
draw axex.top(1,1);  
fin;
```



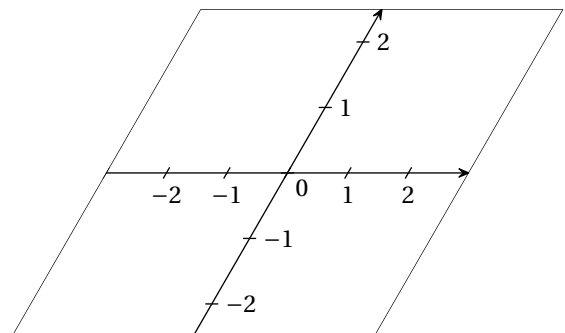
### Exemple 3

```
repere(-3,3,0.8cm,-2.5,2.5,1cm);  
draw axes(1,1) withcolor pourpre;  
draw cadre;  
fin;
```



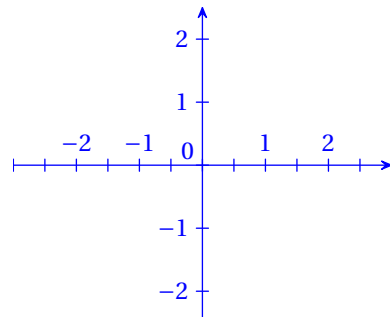
### Exemple 4

```
repere(-3,3,0.8cm,-2.5,2.5,1cm,60);  
draw axes.lrt(1,1);  
draw cadre;  
fin;
```



### Exemple 5

```
repere.orth(-3,3,5cm,-2.5,2.5);  
draw axex.top(0.5,1) withcolor bleu;  
draw axey(1,1) withcolor bleu;  
fin;
```



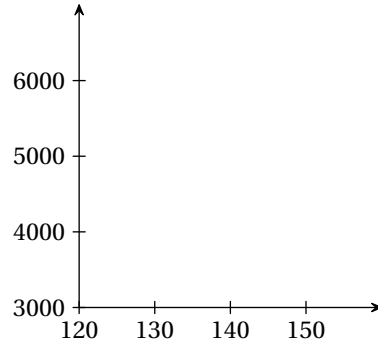
## 2.3.2 Réglages des axes

Les axes sont dessinés, gradués et étiquetés par défaut sur toute la longueur du repère utilisateur. Pour des valeurs différentes on peut utiliser les macros suivantes :

`setaxes(xmin,xmax,ymin,ymax)` définit les valeurs minimales et maximales pour les axes.  
`setgrad(xmin,xmax,ymin,ymax)` définit les valeurs minimales et maximales pour les graduations.  
`setval(xmin,xmax,ymin,ymax)` définit les valeurs minimales et maximales pour l'étiquetage.  
`flecheaxe` booléen égal à `true` par défaut qui permet de dessiner, ou non, des flèches au bout des axes.  
`axexo.pos(grad,val)`, `axeyo.pos(grad,val)`, `axeso.pos(grad,val)` macros identiques aux précédentes sauf pour l'étiquette correspondant à l'intersection des axes qui est toujours dessinée.  
`axexn.pos(grad,val)`, `axeyn.pos(grad,val)`, `axesn.pos(grad,val)` macros identiques aux précédentes sauf pour l'étiquette correspondant à l'intersection des axes qui n'est jamais dessinée.

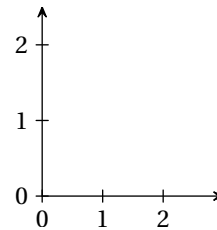
### Exemple 6

```
repere.larg(110,160,5cm,2000,7000,5cm);
interaxes(120,3000);
setaxes(120,160,3000,7000);
setgrad(120,160,3000,7000);
draw axex(10,10);
draw axey(1000,1000);
fin;
```



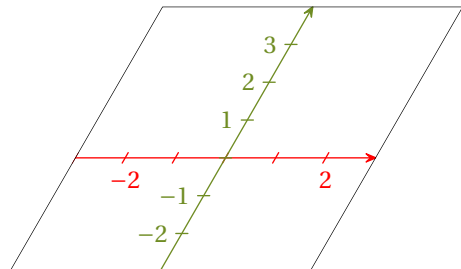
### Exemple 7

```
repere(-0.5,3,0.8cm,-0.5,2.5,1cm);
setaxes(0,3,0,2.5);
draw axeso(1,1);
fin;
```



### Exemple 8

```
repere.larg(-3,3,6cm,-3,4,3.5cm,60);
draw axexn(1,2) withcolor rouge;
draw axeyn(1,1) withcolor olive;
draw cadre;
fin;
```

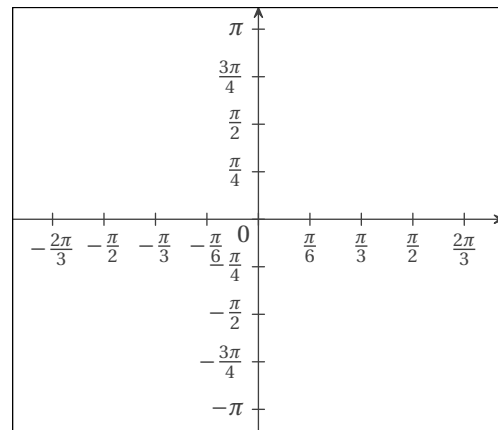


### 2.3.3 Graduations multiples de $\pi$

`axexpi.pos(n,d)` axe des abscisses gradué et étiqueté avec un pas de  $\frac{n\pi}{d}$ . Les fractions sont composées en mode normal. Pour les obtenir en mode « displaystyle », la variable de type booléen `displayfrac` doit être égale à `true`.  
`axeypoi.pos(n,d)` axe des ordonnées gradué et étiqueté avec un pas de  $\frac{n\pi}{d}$ .  
`axespi.pos(n,d)` les deux axes gradués et étiquetés avec un pas de  $\frac{n\pi}{d}$ .  
`axexpio.pos(n,d)`, `axeypio.pos(n,d)`, `axespio.pos(n,d)` même chose que précédemment sauf pour l'étiquette correspondant à l'intersection des axes qui est toujours dessinée.  
`axexpin.pos(n,d)`, `axeypin.pos(n,d)`, `axespin.pos(n,d)` même chose que précédemment sauf pour l'étiquette correspondant à l'intersection des axes qui n'est jamais dessinée.

## Exemple 9

```
repere(-2.5,2.5,1.3cm,-3.5,3.5,0.8cm);
draw axexpi(1,6) withcolor grisfonce;
draw axeypi(1,4) withcolor grisfonce;
draw cadre;
fin;
```



### 2.3.4 Graduations isolées

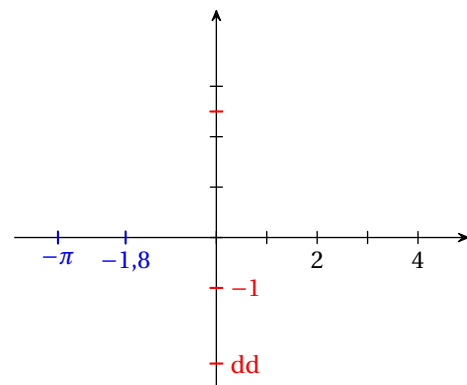
**axexpart.pos(x1,lab1,x2,lab2,...)** graduation et étiquetage partiels de l'axe des abscisses pour les valeurs  $x1$ ,  $x2...$  et les étiquettes  $lab1$ ,  $lab2...$  à la position **pos**. Si **pos** est omis, les étiquettes sont placées à la position **bot**. Les étiquettes peuvent être soit des chaînes de caractères ("aa", "bonjour"), soit des expressions du type **btex**  $\pi$  **etex** (ou **LaTeX**(" $\pi$ ") voir page 36), soit d'autres figures. Si **labn** est omis, la valeur de **xn** sera utilisée comme étiquette. Pour obtenir une graduation sans étiquette, on peut utiliser la chaîne vide "".

On peut désactiver le dessin de la graduation en donnant la valeur **false** à **boolgradxpart**.

**axeypart.pos(y1,lab1,y2,lab2,...)** même chose sur l'axe des ordonnées. Si **pos** est omis, les étiquettes sont placées à la position **lft**.

## Exemple 10

```
repere.orth(-4,5,6cm,-3,4.5);
setgrad(0,5,0,3);
setval(0,5,0,3);
draw axex(1,2);
draw axey(1,0);
draw axexpart(-1.8,-pi,LaTeX("$-\pi$"))
                                withcolor bleu;
draw axeypart.rt(-2.5,"dd",-1,2.5,"")
                                withcolor rouge;
fin;
```



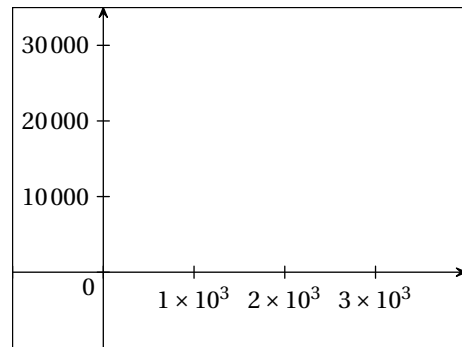
### 2.3.5 Ajout de texte sur les graduations

**extranumx** chaîne de caractères qui sera ajoutée après les valeurs des graduations sur l'axe des abscisses avant d'être composée avec la commande **\num** de **siunitx**.

**extranumy** même chose sur l'axe des ordonnées.

## Exemple 11

```
repere(-1,4,1.2cm,-1,3.5,1cm);
extranumx:="e3";
extranumy:="0000";
draw axes(1,1);
draw cadre;
fin;
```



## 2.4 Quadrillages

`setquad(xmin,xmax,ymin,ymax)` définit les valeurs minimales et maximales pour le tracé des quadrillages.<sup>1</sup>

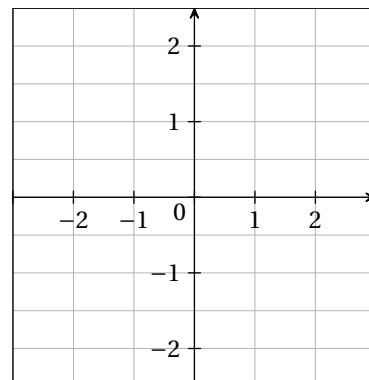
`quadrillage(x,y)` quadrillage avec un pas de  $x$  sur l'axe des abscisses et de  $y$  sur l'axe des ordonnées. L'épaisseur des traits par défaut est 0.3bp et la couleur par défaut est 0.7white.

`papiermillimetre` comme son nom l'indique... Les couleurs et les épaisseurs des traits sont stockées dans `pm_coula`, `pm_coulb`, `pm_coulc` et `pm_epa`, `pm_epb`, `pm_epc`.

`papierpointe(x,y)` quadrillage formé de points avec un pas de  $x$  sur l'axe des abscisses et de  $y$  sur l'axe des ordonnées. La taille des points par défaut est 2bp.

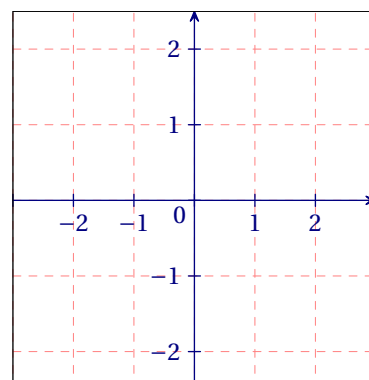
## Exemple 12

```
repere(-3,3,0.8cm,-2.5,2.5,1cm);
draw quadrillage(1,0.5);
draw axes(1,1);
draw cadre;
fin;
```



## Exemple 13

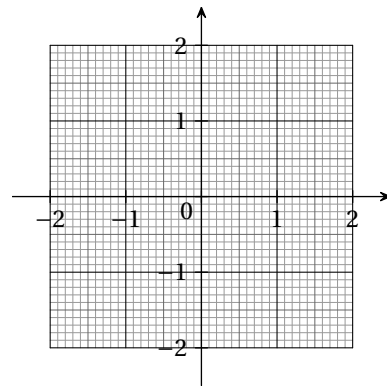
```
repere(-3,3,0.8cm,-2.5,2.5,1cm);
draw quadrillage(1,1) dashed evenly
                        withcolor (1,0.5,0.5);
draw axes(1,1) withcolor marine;
draw cadre;
fin;
```



1. Il existe une macro `settout` qui appelle successivement `setaxes`, `setgrad`, `setval` et `setquad`.

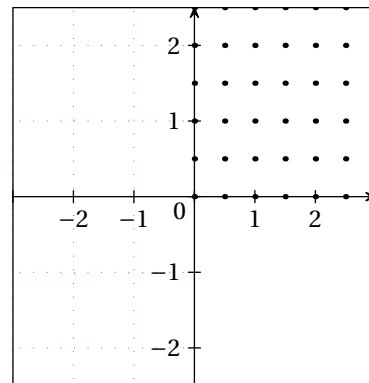
### Exemple 14

```
repere(-2.5,2.5,1cm,-2.5,2.5,1cm);
setquad(-2,2,-2,2);
draw papiermillimetre;
draw axes(1,1);
fin;
```



### Exemple 15

```
repere(-3,3,0.8cm,-2.5,2.5,1cm);
setquad(0,3,0,2.5);
draw papierpointe(0.5,0.5);
setquad(-3,0,-2.5,2.5);
draw quadrillage(1,1) dashed
      withdots withcolor magenta;
draw axes(1,1);
draw cadre;
fin;
```



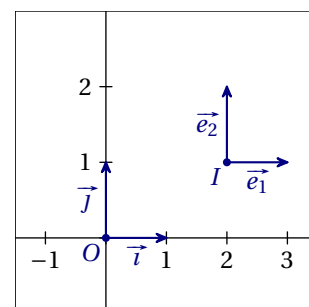
## 2.5 Base

**base(0,i,j)** figure formée par le point d'intersection des axes et son nom (0), ainsi que des deux vecteurs de la base et leurs noms (i et j surmontés d'une flèche). Si les noms sont de la forme « lettre + nombre », le nombre est affiché en indice.

**basep(0,I,J)** figure formée par le point d'intersection des axes et son nom (0), ainsi que des deux points qui définissent le repère.

### Exemple 16

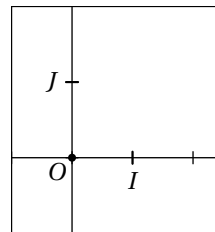
```
repere(-1.5,3.5,0.8cm,-1,3,1cm);
flecheaxe:=false;
draw axesn(1,1);
drawoptions(withcolor marine);
draw base(0,i,j);
interaxes(2,1);
draw base(I,e1,e2);
drawoptions();
draw cadre;
fin;
```





### Exemple 17

```
repere(-1,2.5,0.8cm,-1,2,1cm);  
flecheaxe:=false;  
draw axes(1,0);  
draw basep(0,I,J);  
draw cadre;  
fin;
```



## 3 Points, vecteurs

### 3.1 Points

**(x,y)** désigne le point (ou le vecteur) de coordonnées cartésiennes  $x$  et  $y$  dans le repère utilisateur.

**pol(r,t)** désigne le point (ou le vecteur) de coordonnées  $(r \cos t; r \sin t)$  dans le repère utilisateur.

**pold(r,t)** même chose avec l'angle donné en degrés.

Les macros suivantes sont directement inspirées des macros similaires de `geometriesyr16.mp`.

**MarquePoint(A)** marque le point A. Le style de marque est contrôlé par le paramètre `marque_p` qui peut prendre les valeurs "plein" (valeur par défaut), "creux" ou "croix". Une autre valeur que celles-ci ne produira aucune marque.

**pointe(A,B,C...)** permet de marquer plusieurs points.

**nomme.pos(A,nom)** marque le point et affiche son nom à la position `pos` (qui peut être `rt`, `urt`, `top`, `ulft`...). `nom` peut être soit une chaîne de caractères, soit une expression du type `btex ... etex`, soit une autre figure. Si `nom` est omis, le nom A est affiché. S'il s'agit d'un élément d'un tableau de points (A1, A2...), le nombre est affiché en indice.

**nomme[a](A,nom)** Il est possible d'obtenir un placement plus fin des étiquettes en remplaçant la position au sens de METAPOST (`rt`, `urt`...) par un nombre qui représente alors la position de l'étiquette par rapport au point en degrés.

**nommerot.pos(A,nom)(angle)** Même chose que `nomme.pos(A,nom)` mais l'étiquette est tournée d'un angle `angle`.

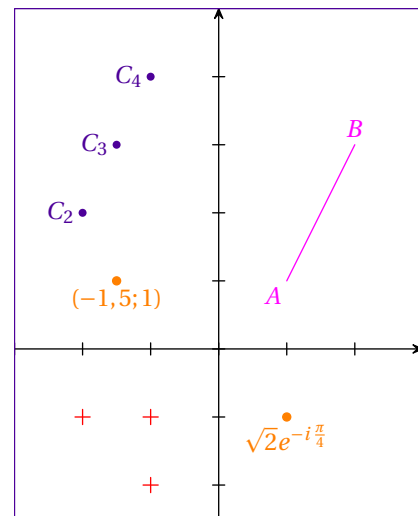
**nommerot[a](A,nom)(angle)** Même chose que `nomme[a](A,nom)` mais l'étiquette est tournée d'un angle `angle`.

### Exemple 18

```

repere(-3,3,0.9cm,-2.5,5,0.9cm);
pair A,B,C[],D,E,F;
A=(1,1);B=(2,3);
D=(-2,-1);E=(-1,-1);F=(-1,-2);
draw axes(1,0);
marque_p="";drawoptions(withcolor magenta);
nomme.llft(A);nomme.top(B);draw A--B;
marque_p="croix";drawoptions(withcolor rouge);
pointe(D,E,F);
marque_p="creux";drawoptions(withcolor orange);
nomme.bot(pol(sqrt(2),-pi/4),
"$\sqrt{2}e^{-i\frac{\pi}{4}}$");
nomme.bot((-1.5,1),"$(-1,5;1)$");
marque_p="plein";drawoptions(withcolor violet);
for i=2 upto 4:
  C[i]=(-3+i/2,i);nomme.llft(C[i]);
endfor
draw cadre;
fin;

```

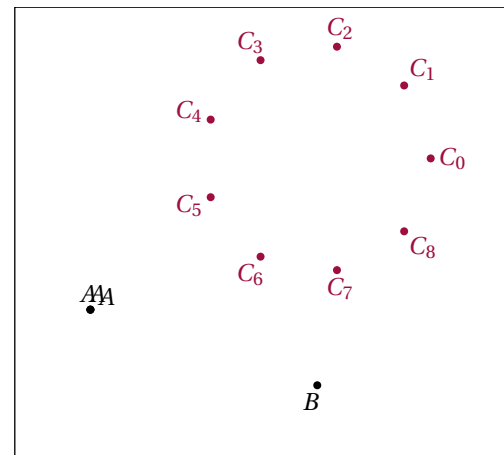


### Exemple 19

```

repere(-3,3.5,1cm,-3,3,1cm);
pair A,B,C[];
A=(-2,-1);B=(1,-2);
nomme[40](A);nomme[70](A);nomme[100](A);
nomme[-110](B);
for i=0 upto 8:
  C[i]:= (1+1.5*cosd(40i),1+1.5*sind(40i));
  nomme[40*i](C[i]) couleur pourpre;
endfor
draw cadre;
fin;

```

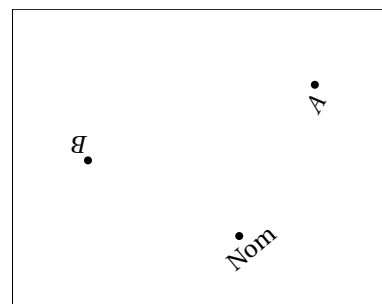


### Exemple 20

```

repere(-2,3,1cm,-1,3,1cm);
pair A,B,C;
A=(2,2);B=(-1,1);C=(1,0);
nommerot.bot(A)(60);
nommerot[120](B)(180);
nommerot[-45](C,"Nom")(40);
draw cadre;
fin;

```



## 3.2 Vecteurs

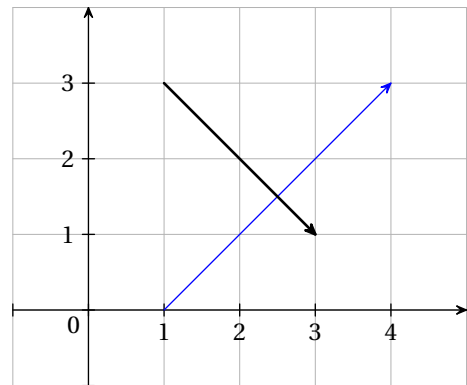
**drawvecteur(A,u)** Trace le représentant d'origine A du vecteur u. C'est l'équivalent de **drawarrow A--A+u**

## Exemple 21

```

repere(-1,5,1cm,-1,4,1cm);
pair A,u;
A=(1,0);u=(3,3);
draw quadrillage(1,1);
draw axes(1,1);
drawvecteur(A,u) couleur bleu;
drawvecteur((1,3),(2,-2)) epaisseur 1;
fin;

```



## 4 Droites, courbes...

### 4.1 Droites

**droite(A,B)** droite  $(AB)$ .

**droite(a,b,c)** droite d'équation  $ax + by + c = 0$  dans le repère utilisateur.

**droite(a,b)** droite d'équation  $y = ax + b$  dans le repère utilisateur.

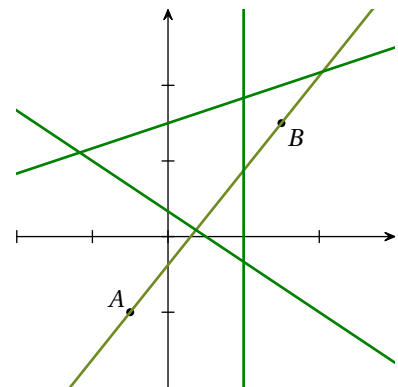
**droite(c)** droite d'équation  $x = c$  dans le repère utilisateur.

## Exemple 22

```

repere(-2,3,1cm,-2,3,1cm);
pair A,B;
A=(-0.5,-1);B=(1.5,1.5);
draw axes(1,0);
nomme.ulft(A);
nomme.lrt(B);
drawoptions(withpen pencircle scaled 1);
draw droite(A,B) withcolor olive;
draw droite(1) withcolor vertfonce;           %x=1
draw droite(1/3,3/2) withcolor vertfonce;    %y=(1/3)x+3/2
draw droite(2,3,-1) withcolor vertfonce;    %2x+3y-1=0
fin;

```



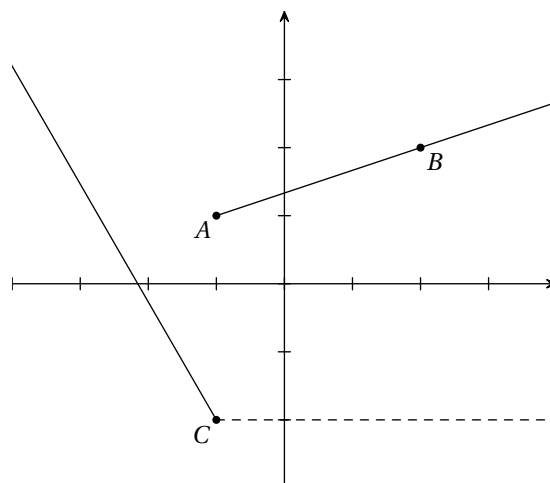
### 4.2 Demi-droites

**demidroite(A,B)** demi-droite  $[AB)$ .

**demidroite(A,a)** demi-droite d'origine  $A$  qui fait un angle  $a$  avec la direction  $(Ox)$ .

## Exemple 23

```
repere(-4,4,0.9cm,-3,4,0.9cm);
pair A,B,C;
numeric a;
A=(-1,1);B=(2,2);C=(-1,-2);
draw axes(1,0);
draw demidroite(A,B);
nomme.llft(A);nomme.lrt(B);
draw demidroite(C,0) dashed evenly;
draw demidroite(C,120);
nomme.llft(C);
fin;
```



## 4.3 Courbes et fonctions

METAPOST permet de définir simplement des fonctions (en utilisant par exemple la syntaxe suivante : `vardef f(expr x)=2x+1 enddef;`) et de définir des courbes passant par des points donnés (A..B..C). Ces possibilités sont utilisées dans les macros qui suivent.

**courbefonc(f)()** courbe représentant la fonction  $f$  sur l'intervalle définissant le repère.

**courbefonc(f)(xmin,xmax)** courbe représentant la fonction  $f$  sur l'intervalle  $[xmin;xmax]$ .

**courbefonc(f)(xmin,xmax,n)** courbe représentant la fonction  $f$  sur l'intervalle  $[xmin;xmax]$  en utilisant  $n$  points d'interpolation. La valeur par défaut de  $n$  est 60.

**courbepoints(f)(xmin,xmax,n)** ne trace que les  $n$  points sans les relier. Les points sont dessinés en fonction de la valeur de `marque_p` (voir 3.1).

**fonccourbe.p(x)** image de  $x$  par la fonction dont la courbe représentative est le chemin  $p$ . La macro renvoie 0 si la fonction n'est pas définie.

**nomme.pos(p,x,nom)** affiche `nom` au point d'abscisse  $x$  de la courbe  $p$  à la position `pos`. `nom` peut être soit une chaîne de caractères, soit une expression du type `btex ... etex`, soit une autre figure. Si `nom` est omis, le nom  $p$  est affiché. S'il s'agit d'un élément d'un tableau de points ( $p_1, p_2, \dots$ ), le nombre est affiché en indice.

**intercourbes(P,p,q)** stocke dans le tableau de points  $P$  les points d'intersection des chemins  $p$  et  $q$ .  $P_1$  est un des points d'intersection,  $P_2$  un autre etc. Il faut, avant d'utiliser cette macro, déclarer le tableau  $P$  de la façon suivante : `pair P[];`

**ptantecedents(P,y,p)** stocke dans le tableau de points  $P$  les points du chemin  $p$  d'ordonnée  $y$ . De même que précédemment, le tableau  $P$  doit être déclaré avant d'utiliser cette macro.

**antecedents(X,y,p)** stocke dans le tableau de nombres  $X$  les antécédents de  $y$  par la fonction dont la courbe représentative est le chemin  $p$ . De même que précédemment, le tableau  $X$  doit être déclaré avant d'utiliser cette macro.

**marquepointcourbe(p,x1,x2,...)** marque les points de la courbe  $p$  d'abscisses  $x_1, x_2, \dots$ . La marque dépend de la valeur de `marque_p`.

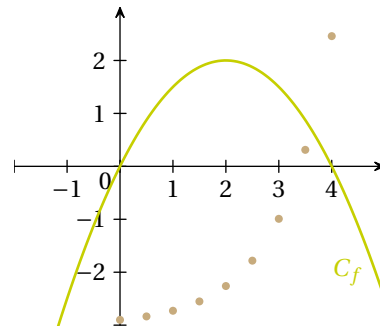
**marquepointchemin(p,n1,n2,...)** dans le cas d'un chemin défini par A..B..C., marque le  $n_1$ -ième point, le  $n_2$ -ième point... La marque dépend de la valeur de `marque_p`. Attention, le premier point est numéroté 0.

### Exemple 24

```

repere(-2,5,0.7cm,-3,3,0.7cm);
vardef f(expr x)=-0.5(x**2)+2*x enddef;
vardef g(expr x)=exp(x)/10-3 enddef;
path C_f;
draw axes(1,1);
drawoptions(withcolor moutarde);
C_f= courbefonc(f)();
draw C_f withpen pencircle scaled 1;
nomme.llft(C_f,4.7);
drawoptions(withcolor beige);
draw courbepoints(g)(0,4,9);
fin;

```

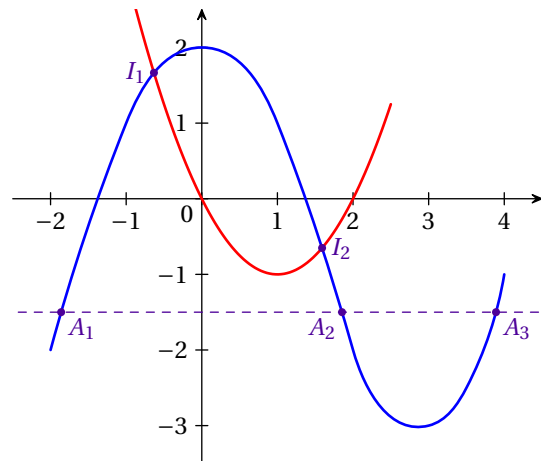


### Exemple 25

```

repere(-2.5,4.5,1cm,-3.5,2.5,1cm);
path p,C_f;
pair I[],A[];
vardef f(expr x)= x**2-2x enddef;
p=(-2,-2)..(-1,1)..(0,2)..(1,1)
  ..(2,-2)..(3,-3)..(3.5,-2.5)
  ..(4,-1);
C_f= courbefonc(f)(-2,2.5);
draw axes(1,1);
drawoptions(withpen pencircle scaled 1);
draw p withcolor bleu;
draw C_f withcolor rouge;
intercourbes(I,C_f,p);
drawoptions(withcolor violet);
nomme.lft(I1);nomme.rt(I2);
draw droite(0,-1.5) dashed evenly;
ptantecedents(A,-1.5,p);
nomme.lrt(A1);nomme.llft(A2);
nomme.lrt(A3);
fin;

```

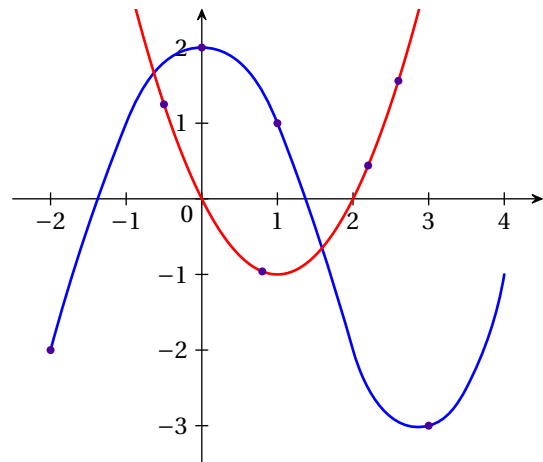


### Exemple 26

```

repere(-2.5,4.5,1cm,-3.5,2.5,1cm);
path p,C_f;
pair I[],A[];
vardef f(expr x)= x**2-2x enddef;
p=(-2,-2)..(-1,1)..(0,2)
  ..(1,1)..(2,-2)..(3,-3)
  ..(3.5,-2.5)..(4,-1);
C_f= courbefonc(f)(-2,3);
draw axes(1,1);
drawoptions(withpen pencircle scaled 1);
draw p withcolor bleu;
draw C_f withcolor rouge;
drawoptions(withcolor violet);
marquepointcourbe(C_f,-0.5,0.8,2.2,2.6);
marquepointchemin(p,0,2,3,5);
fin;

```



### 4.4 Placement automatique du nom des courbes

**nomme(p,nom)** affiche nom au niveau d'un point d'intersection de  $p$  et du contour de la figure. La position est choisie en fonction de la chaîne **prefnomme** qui peut prendre les valeurs "right" (valeur par défaut), "left", "top" ou "bottom" et de la place restante.

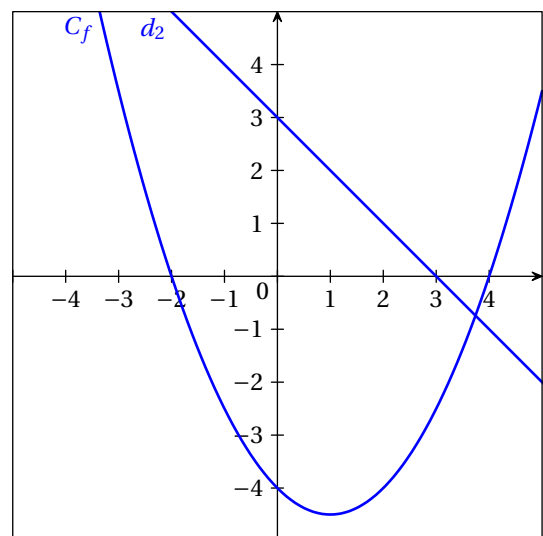
### Exemple 27

```

prefnomme:="left";

repere(-5,5,0.7cm,-5,5,0.7cm);
path d,C_f;
d=droite(-1,3);
vardef f(expr x)=0.5(x**2)-x -4 enddef;
C_f=courbefonc(f)();
draw axes(1,1);
draw d epaisseur 1 couleur bleu;
draw C_f epaisseur 1 couleur bleu;
nomme(d,"$d_2$") couleur bleu;
nomme(C_f) couleur bleu;
draw cadre;
fin;

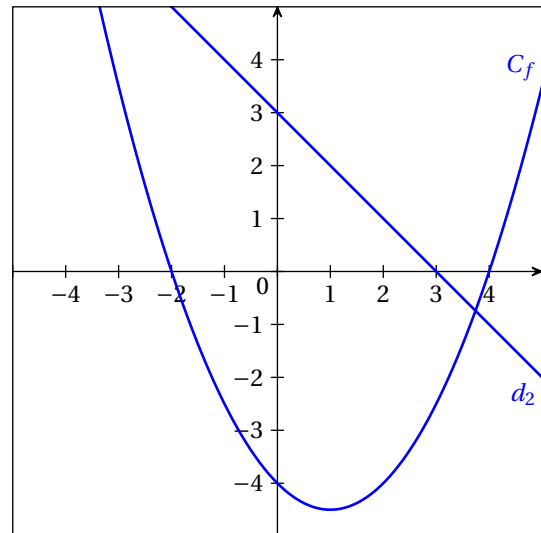
```



### Exemple 28

```
prefnomme:="bottom";

repere(-5,5,0.7cm,-5,5,0.7cm);
path d,C_f;
d=droite(-1,3);
vardef f(expr x)=0.5(x**2)-x -4 enddef;
C_f=courbefonc(f)();
draw axes(1,1);
draw d epaisseur 1 couleur bleu;
draw C_f epaisseur 1 couleur bleu;
nomme(d,"$d_2$") couleur bleu;
nomme(C_f) couleur bleu;
draw cadre;
fin;
```



## 4.5 Dérivée et tangentes

**der.p(x)** image de  $x$  par la dérivée de la fonction dont la courbe représentative est  $p$ .

**tangente(p,x)** tangente à la courbe  $p$  au point d'abscisse  $x$ .

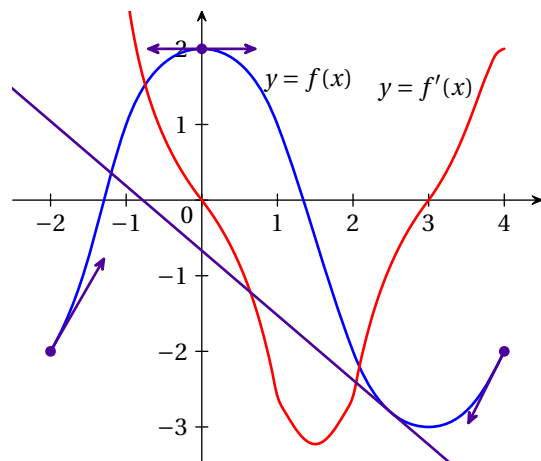
**tangente.gauche(p,x,long)** flèche de longueur  $long$  représentant la demi-tangente gauche à la courbe  $p$  au point d'abscisse  $x$ . Le paramètre  $long$  est optionnel. Sa valeur par défaut est 20bp.

**tangente.droite(p,x,long)** idem à droite.

**tangente.double(p,x,long)** idem des deux côtés.

### Exemple 29

```
repere(-2.5,4.5,1cm,-3.5,2.5,1cm);
path p,q;
p=(-2,-2){dir 60}..(-1,1)
  ..(0,2){right}..(1,1)..(2,-2)
  ..(3,-3){right}..(4,-2){(1,2)};
q= courbefonc(der.p)(-1,4);
draw axes(1,1);
drawoptions(withpen pencircle scaled 1);
draw p withcolor bleu;
nomme.rt(p,0.7,"$y=f(x)$");
draw q withcolor rouge;
nomme.lft(q,3.7,"$y=f'(x)$");
drawoptions(withpen pencircle scaled 1
  withcolor violet);
draw tangente.double(p,0);
draw tangente.droite(p,-2,40);
draw tangente.gauche(p,4,30);
draw tangente(p,2.5);
fin;
```



## 4.6 Interpolation

METAPOST propose les commandes suivantes (qui peuvent être combinées dans une même courbe) :

**A--B--C--** Ligne brisée passant par les points  $A, B, C...$

**A..B..C..** Courbe de Bézier passant par les points  $A, B, C...$

## Interpolation polynomiale

`repere.mp` propose aussi les commandes ci-dessous (pas toujours la meilleure méthode d'approximation...). À compiler avec `mpost -numbersystem="decimal" fichier.mp` pour gagner en précision.

**lagrange(A,B,C,...)()** Courbe passant par  $A, B, C...$  représentant le polynôme de degré maximal  $n - 1$  tel que  $P(x_A) = y_A, P(x_B) = y_B...$  sur l'intervalle définissant le repère.

**lagrange(A,B,C,...)(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

**lagrange(x1,y1,x2,y2,x3,y3...)** Courbe passant par les points  $(x_1; y_1), (x_2; y_2), (x_3; y_3)...$  représentant le polynôme de degré maximal  $n - 1$  tel que  $P(x_i) = y_i$  sur l'intervalle définissant le repère.

**lagrange(x1,y1,x2,y2,x3,y3...)(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

**hermite((x1,y1,y'1),(x2,y2,y'2)...)** Courbe passant par les points  $(x_1; y_1), (x_2; y_2), (x_3; y_3)...$  représentant le polynôme de degré maximal  $2n - 1$  tel que  $P(x_i) = y_i$  et  $P'(x_i) = y'_i$  sur l'intervalle définissant le repère.

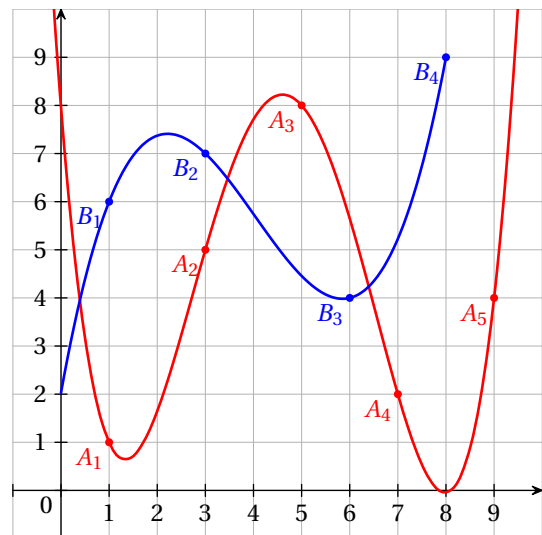
**hermite((x1,y1,y'1),(x2,y2,y'2)...)(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

**hermite(A,y'A,B,y'B,C,y'C...)** Courbe passant par les points  $A, B, C...$  représentant le polynôme de degré maximal  $2n - 1$  tel que  $P(x_A) = y_A$  et  $P'(x_A) = y'_A...$  sur l'intervalle définissant le repère.

**hermite(A,y'A,B,y'B,C,y'C...)(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

## Exemple 30

```
repere.orth(-1,10,7cm,-1,10);
pair A[],B[];
A[1]=(1,1);A[2]=(3,5);A[3]=(5,8);
A[4]=(7,2);A[5]=(9,4);
B[1]=(1,6);B[2]=(3,7);B[3]=(6,4);B[4]=(8,9);
path L;L=lagrange(A[1],A[2],A[3],A[4],A[5])();
path C;C=lagrange(1,6,3,7,6,4,8,9)(0,8);
draw quadrillage(1,1);
draw axes(1,1);
draw L epaisseur 1 couleur rouge;
draw C epaisseur 1 couleur bleu;
for i=1 upto 5: nomme.llft(A[i]) couleur rouge;
endfor
for i=1 upto 4: nomme.llft(B[i]) couleur bleu;
endfor
fin;
```



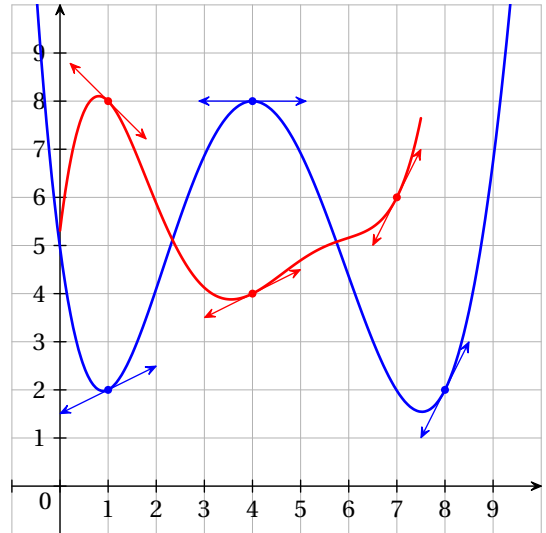


### Exemple 31

```

repere.orth(-1,10,7cm,-1,10);
draw quadrillage(1,1);
draw axes(1,1);
path H;H=hermite((1,2,0.5),(4,8,0),(8,2,2))();
draw H epaisseur 1 couleur bleu;
draw tangente.double(H,1) couleur bleu;
draw tangente.double(H,4) couleur bleu;
draw tangente.double(H,8) couleur bleu;
pair A,B,C; A:=(1,8);B:=(4,4);C:=(7,6);
path I;I=hermite(A,-1,B,0.5,C,2)(0,7.5);
draw I epaisseur 1 couleur rouge;
draw tangente.double(I,1) couleur rouge;
draw tangente.double(I,4) couleur rouge;
draw tangente.double(I,7) couleur rouge;
fin;

```



### Interpolation à l'aide de splines cubiques

**spline(A,B,C... )()** Courbe passant par les points  $A, B, C$  représentant une fonction cubique par morceaux telle que  $f(x_A) = y_A, f(x_B) = y_B \dots$  sur l'intervalle définissant le repère.

**spline(A,B,C... )(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

**spline(xA,yA,xB,yB,xC,yC,... )()** Même courbe que précédemment (sur l'intervalle définissant le repère) mais les valeurs sont données sous forme de liste.

**spline(xA,yA,xB,yB,xC,yC... )(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

**spline(A,<y'A>,B,<y'B>,C,<y'C>... )()** Courbe passant par les points  $A, B, C$  représentant une fonction cubique par morceaux telle que  $f(x_A) = y_A, f(x_B) = y_B \dots$  et, le cas échéant,  $f'(x_A) = y'_A, f'(x_B) = y'_B \dots$  sur l'intervalle définissant le repère.

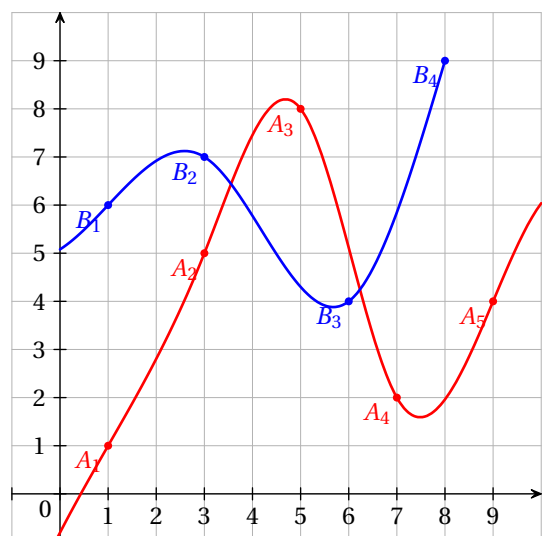
**spline(A,<y'A>,B,<y'B>,C,<y'C>... )(xmin,xmax)** Même courbe que précédemment mais sur l'intervalle  $[xmin; xmax]$ .

### Exemple 32

```

repere.orth(-1,10,7cm,-1,10);
pair A[],B[];
A[1]=(1,1);A[2]=(3,5);A[3]=(5,8);
A[4]=(7,2);A[5]=(9,4);
B[1]=(1,6);B[2]=(3,7);B[3]=(6,4);B[4]=(8,9);
path L;L=spline(A[1],A[2],A[3],A[4],A[5])();
path C;C=spline(1,6,3,7,6,4,8,9)(0,8);
draw quadrillage(1,1);
draw axes(1,1);
draw L epaisseur 1 couleur rouge;
draw C epaisseur 1 couleur bleu;
for i=1 upto 5: nomme.llft(A[i]) couleur rouge;
endfor
for i=1 upto 4: nomme.llft(B[i]) couleur bleu;
endfor
fin;

```

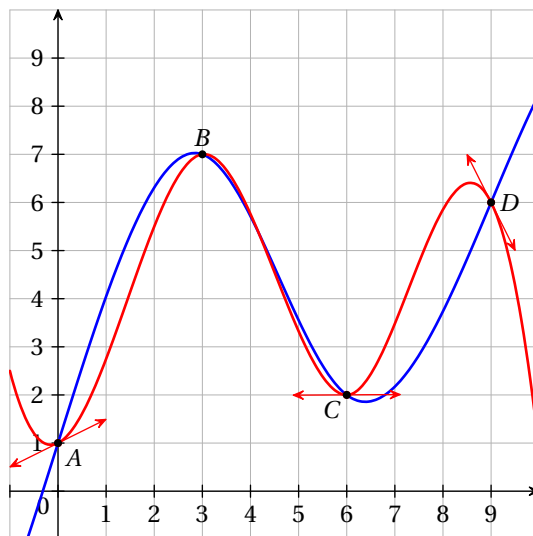


### Exemple 33

```

repere.orth(-1,10,7cm,-1,10);
pair A,B,C,D;
A=(0,1);B=(3,7);C=(6,2);D=(9,6);
path S,T;
S=spline(A,B,C,D)();
T=spline(A,0.5,B,C,0,D,-2)();
draw quadrillage(1,1);
draw axes(1,1);
draw S epaisseur 1 couleur bleu;
draw T epaisseur 1 couleur rouge;
draw tangente.double(T,0) couleur rouge;
draw tangente.double(T,6) couleur rouge;
draw tangente.double(T,9) couleur rouge;
nomme.lrt(A);nomme.top(B);
nomme.llft(C);nomme.rt(D);
fin;

```



Attention : lorsque la courbure est importante, l'utilisation de `tangente` peut donner des résultats erronés...

## 5 Suites

`suite(u,deb,fin)` figure formée des points  $(i; u_i)$  pour  $i$  variant entre `deb` et `fin`.

`suiterec(f,deb,fin,init)` ligne brisée (« escalier » ou « escargot ») permettant de visualiser les termes de la suite définie par  $u_{n+1} = f(u_n)$  de premier terme  $u_{deb} = init$  et de dernier terme  $u_{fin}$ .

`suiterecprojx.pos(lab,min,max)` figure formée des segments joignant les points  $(u_n; u_n)$  et  $(u_n; 0)$  pour  $n$  compris entre `min` et `max`. La suite  $u$  et sa valeur initiale sont définies par le dernier appel de la macro `suiterec`. `lab` désigne l'étiquette au niveau de l'axe des abscisses placée à la position `pos`. Si `lab` est la chaîne vide "", rien n'est écrit ; si `lab` est une autre chaîne de caractère (par ex. "u"), elle est utilisée comme nom de la suite (on obtiendra  $u_0, u_1, \dots$ ) ; si `lab` est un nombre, les valeurs de la suites seront affichées et arrondies à `lab` décimales. Les valeurs `min` et `max` sont facultatives et égales par défaut aux valeurs `deb` et `fin` passées à la macro `suiterec`.

`suiterecprojy.pos(lab,min,max)` même chose sur l'axe des ordonnées.

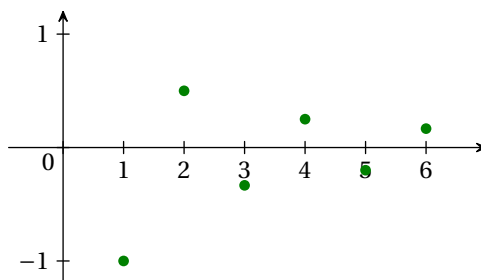
`suiterecproj(lab,min,max)` même chose sur les deux axes. Les positions sont `bot` sur l'axe des abscisses et `lft` sur l'axe des ordonnées.

### Exemple 34

```

repere(-0.9,7,0.8cm,-1.2,1.2,1.5cm);
vardef u(expr n)=(-1)**n/n enddef;
taillepoint:=4;
draw axes(1,1);
draw suite(u,1,6) withcolor vertfonce;
fin;

```

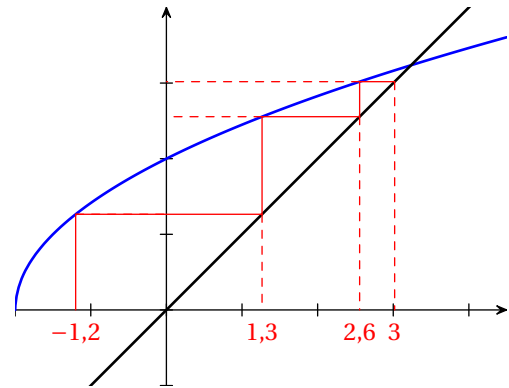


### Exemple 35

```

repere(-2,4.5,1cm,-1,4,1cm);
vardef f(expr x)=sqrt(2*x+4) enddef;
path C_f,sr;
C_f= courbefonc(f)();
sr=suiterec(f,0,3,-1.2);
draw axes(1,0);
drawoptions(withpen pencircle scaled 1);
draw C_f withcolor bleu;
draw droite(1,0);
drawoptions(withcolor rouge);
draw suiterecprojx.bot(1) dashed evenly;
draw suiterecprojy.lft("") dashed evenly;
draw sr withcolor rouge;
fin;

```

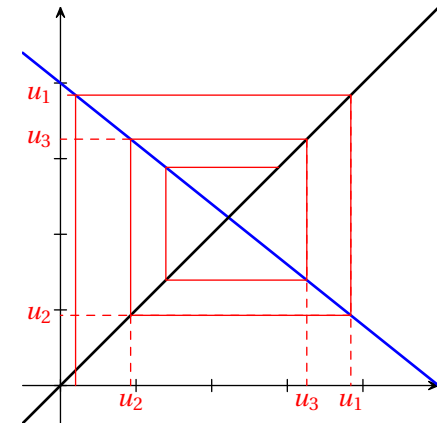


### Exemple 36

```

repere(-0.5,5,1cm,-0.5,5,1cm);
vardef f(expr x)=4-0.8*x enddef;
path C_f,sr;
C_f= courbefonc(f)();
sr=suiterec(f,0,5,0.2);
draw axes(1,0);
drawoptions(withpen pencircle scaled 1);
draw C_f withcolor bleu;
draw droite(1,0);
drawoptions(withcolor rouge);
draw suiterecproj("u",1,3) dashed evenly;
draw sr withcolor rouge;
fin;

```



## 6 Surfaces

### 6.1 Calcul intégral

**entrecourbes(p,q,xmin,xmax)** chemin fermé délimitant la zone comprise entre les courbes **p** et **q** et les droites d'équations  $y = \text{xmin}$  et  $y = \text{xmax}$ . Il peut donc être dessiné, rempli...

**souscourbe(p,xmin,xmax)** chemin fermé délimitant la zone comprise entre la courbe **p**, l'axe des abscisses et les droites d'équations  $y = \text{xmin}$  et  $y = \text{xmax}$ .

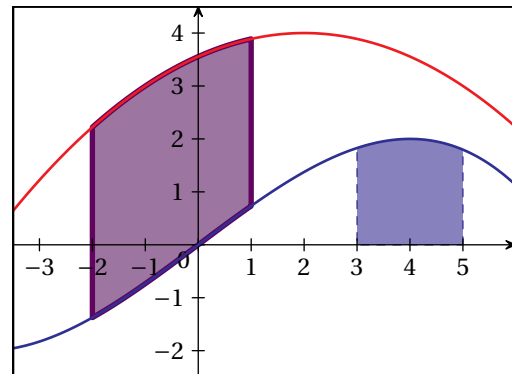
**rectangles.type(p,a,b,n)** figure formée de **n** rectangles s'appuyant sur la courbe **p** entre les abscisses **a** et **b**. **type** peut être **min**, **max**, **droite** ou **gauche**.

### Exemple 37

```

repere(-3.5,6,0.7cm,-2.5,4.5,0.7cm);
vardef f(expr x)= -(x/4)**3+0.75x enddef;
vardef g(expr x)= -((x-2)**2)/9+4 enddef;
path C_f,C_g,p,q;
C_f:= courbefonc(f)();
C_g:= courbefonc(g)();
p:=entrecourbes(C_f,C_g,-2,1);
q:=souscourbe(C_f,3,5);
fill p withcolor 0.5Violet;
draw p withpen pencircle scaled 2
      withcolor Violet;
fill q withcolor 0.5Bleu;
draw q dashed evenly withcolor Bleu;
draw axes(1,1);
drawoptions(withpen pencircle scaled 1);
draw C_f withcolor Bleu;
draw C_g withcolor Rouge;
draw cadre;
fin;

```

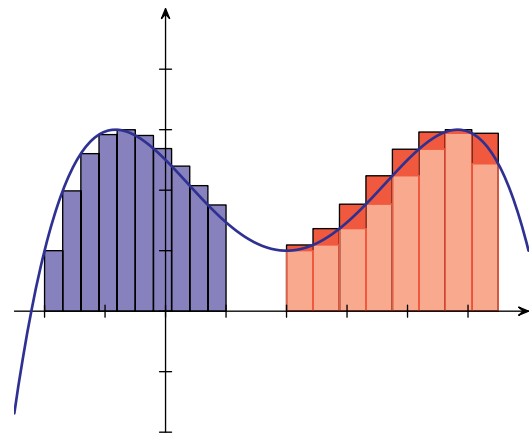


### Exemple 38

```

repere(-2.5,6,0.8cm,-2,5,0.8cm);
vardef f(expr x)=
  -((x-2)**4)/32+((x-2)**2)/2+1
enddef;
path Cf,r[];
Cf= courbefonc(f)();
r1=rectangles.max(Cf,2,5.5,8);
r2=rectangles.min(Cf,2,5.5,8);
r3=rectangles.droite(Cf,-2,1,10);
fill r1 withcolor 0.8Rouge;
fill r2 withcolor 0.4Rouge;
fill r3 withcolor 0.5Bleu;
draw r1;draw r2 withcolor 0.8Rouge;
draw r3;
draw axes(1,0);
draw Cf withcolor Bleu
      withpen pencircle scaled 1;
fin;

```



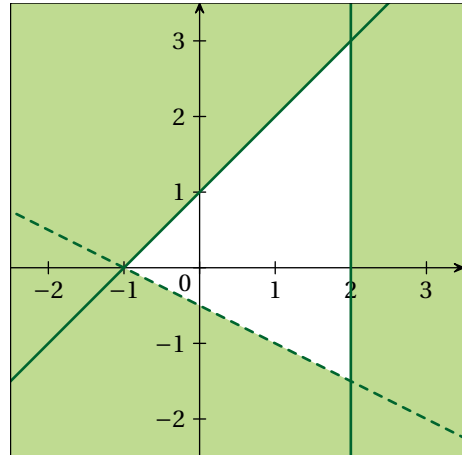
## 6.2 Demi-plans

**demiplaninf(d)** chemin fermé délimité par la droite  $d$  et par la partie inférieure de **cadre** (ou la partie gauche si  $d$  est parallèle à l'axe des ordonnées.

**demiplansup(d)** chemin fermé délimité par la droite  $d$  et par la partie supérieure de **cadre** (ou la partie droite si  $d$  est parallèle à l'axe des ordonnées.

### Exemple 39

```
repere(-2.5,3.5,1cm,-2.5,3.5,1cm);
numeric qqw;qqw=6;
path d[],dp[];
d1=droite(2);
d2=droite(1,1);
d3=droite(-0.5,-0.5);
dp1=demiplansup(d1);
dp2=demiplansup(d2);
dp3=demiplaninf(d3);
for i=1 upto 3:
fill dp[i] withcolor 0.7Lime;
endfor
draw axes(1,1);
drawoptions(withpen pencircle scaled 1
            withcolor Vertfonce);
draw d1;draw d2;
draw d3 dashed evenly;
drawoptions();
draw cadre;
fin;
```



## 7 Projections sur les axes

### 7.1 Projetés

**projetex(A)** projeté de A sur l'axe des abscisses parallèlement à l'axe des ordonnées.

**projetey(A)** projeté de A sur l'axe des ordonnées parallèlement à l'axe des abscisses.

**projectionx.pos(A,lab,dec)** figure constituée du segment joignant A à son projeté sur l'axe des abscisses ainsi que de l'étiquette **lab** placée à la position **pos** par rapport à ce projeté. La valeur **dec** indique un décalage par rapport à l'axe des abscisses. L'étiquette et le décalage sont optionnels.

**projectiony.pos(A,lab,dec)** même chose sur l'axe des ordonnées.

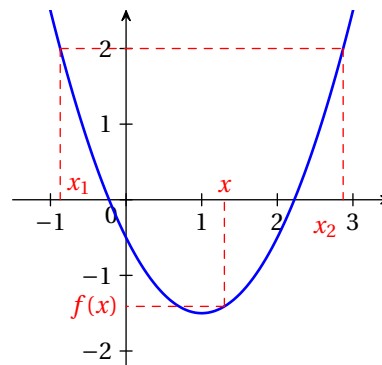
**projectionaxes(A,labx,laby,dec)** figure constituée des segments joignant A à ses projetés sur les axes ainsi que des étiquettes **labx** et **laby** positionnées automatiquement avec un décalage **dec** par rapport aux axes. Les étiquettes et le décalage sont optionnels.

## Exemple 40

```

repere(-1.5,3.5,1cm,-2.2,2.5,1cm);
path Cf; pair A[];
vardef f(expr x)= x**2-2x-0.5 enddef;
Cf= courbefonc(f)();
ptantecedents(A,2,Cf);
draw axes(1,1);
draw Cf withpen pencircle scaled 1 withcolor bleu;
drawoptions(dashed evenly withcolor rouge);
draw projectionaxes((1.3,f(1.3)),"$x$","$f(x)$");
draw projectionx.urt(A1,"$x_1$");
draw projectionx.llft(A2,"$x_2$",-6);
draw A1--A2;
fin;

```



## 7.2 Intervalles

**intervallex.bornes(a,b)** intervalle dessiné sur l'axe des abscisses entre  $a$  et  $b$  avec une épaisseur par défaut de 1.5bp. bornes peut être OO (ouvert à gauche, ouvert à droite), OF, FO ou FF.

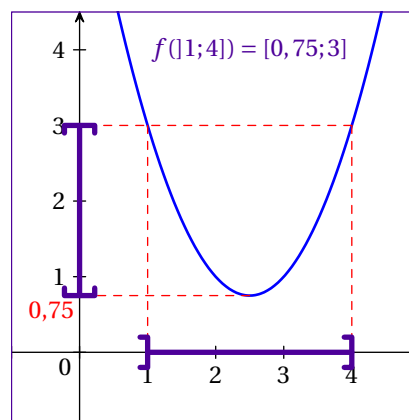
**intervalley.bornes(a,b)** même chose sur l'axe des ordonnées.

## Exemple 41

```

repere(-1,5,0.9cm,-1,4.5,1cm);
vardef f(expr x)=x**2-5x+7 enddef;
draw axes(1,1);
draw courbefonc(f)()
    withpen pencircle scaled 1 withcolor bleu;
drawoptions(dashed evenly withcolor rouge);
draw projectionx.bot((1,f(1)));
draw projectiony.llft((2.5,f(2.5)),"0,75");
draw projectionaxes((4,f(4)));
drawoptions(withcolor violet);
draw intervallex.OF(1,4);
draw intervalley.FF(0.75,3);
label("$f([1;4])=[0,75;3]$", (2.5,4));
draw cadre;
fin;

```



# 8 Statistiques et probabilités

## 8.1 Boite à moustache

**boitemoustache(min,Q1,Me,Q3,max,dec,larg)** « Boite à moustache » correspondant aux données en argument. Elle est située à une distance **dec** de l'axe des abscisses et le rectangle a une largeur de **larg**. Ces deux dernières valeurs sont optionnelles et valent par défaut 1,5 cm et 1 cm.

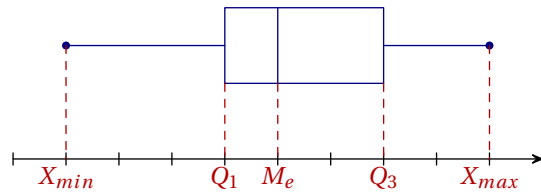
**projboitemoustache(t)** Figure formée des lignes joignant les cinq valeurs du dernier diagramme en boite dessiné à son projeté sur l'axe des abscisses ainsi que de certaines étiquettes : Si **t** est vide, les textes  $X_{min}$ ,  $Q_1$ ,  $M_e$ ,  $Q_3$  et  $X_{max}$  sont affichés ; si **t** est un entier, les valeurs arrondies à  $10^{-t}$  sont affichées ; si **t** est une liste de cinq textes (ou valeurs), ceux-ci sont affichés.

### Exemple 42

```

repere(-0.5,10,0.7cm,-1,5,0.7cm);
setaxes(0,10,0,1);
draw axex(1,0);
draw boitemoustache(1,4,5,7,9)
                        withcolor marine;
draw projboitemoustache.bot()
                        withcolor 0.7rouge dashed evenly;
fin;

```

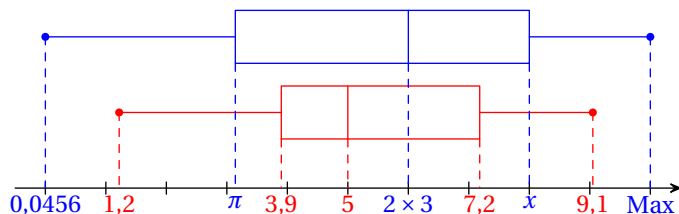


### Exemple 43

```

repere(-1,11,0.8cm,-1,5,0.7cm);
settout(-0.5,10.5,0,1);
draw axex(1,0);
drawoptions(withcolor rouge);
draw boitemoustache(1.22,3.9,5,7.18,9.05,1cm,0.7cm);
draw projboitemoustache.bot(1) dashed evenly;
drawoptions(withcolor blue);
draw boitemoustache(0,3.14,6,8,10,2cm,0.7cm);
draw projboitemoustache.bot(0.0456,"$\pi$","$\numproduct{2x3}$","$x$","Max")
                        dashed evenly;
fin;

```



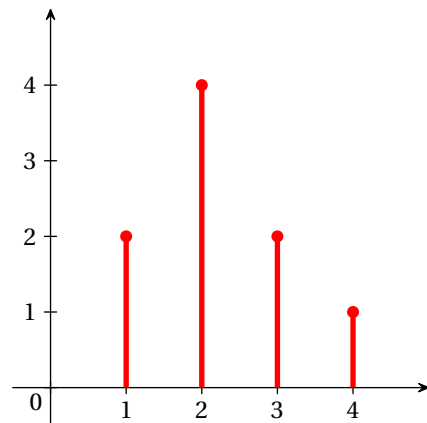
## 8.2 Diagrammes

**diagrammebatons((v1,e1),(v2,e2),... (vn,en))** Figure formée des  $n$  segments joignant les points  $(v1,e1), (v2,e2), \dots (vn,en)$  et leur projeté sur l'axe des abscisses. Les bâtons sont surmontés d'un point dont le diamètre est égal à la largeur des segments multiplié par **diampointsbatons**. **diampointsbatons** est égal à 5 par défaut. On peut lui donner la valeur 0 pour ne pas avoir ces points.

**diagrammebarres((a1,h1),(a2,h2),... (an,hn))** Figure formée de  $n$  barres rectangulaires de hauteurs  $h1 \dots hn$  aux abscisses  $a1 \dots an$ . La largeur de ces barres est le nombre **largbarres** qui vaut 20bp par défaut.

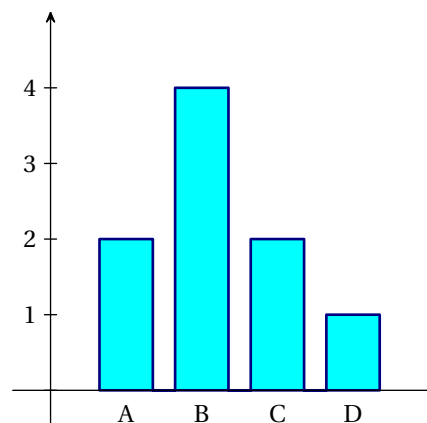
### Exemple 44

```
repere(-0.5,5,1cm,-0.5,5,1cm);
picture diag;
draw axes(1,1);
diag:=diagrammebatons((1,2),(2,4),(3,2),(4,1));
draw diag epaisseur 2 withcolor rouge;
fin
```



### Exemple 45

```
repere(-0.5,5,1cm,-0.5,5,1cm);
path diag;
draw axey(1,1);
diag:=diagrammebarres((1,2),(2,4),(3,2),(4,1));
fill diag withcolor cyan;
draw diag epaisseur 1 withcolor marine;
flecheaxe:=false;
draw axex(0,0);
boolgradxpart:=false;
draw axexpart.bot(1,"A",2,"B",3,"C",4,"D");
fin
```



## 8.3 Probabilités

Quelques fonctions mathématiques sont proposées. Pour les grandes valeurs, on dépasse rapidement les capacités de METAPOST. Il est dans ce cas conseillé de compiler en utilisant la ligne de commande `mpost -numbersystem="decimal" <fichier>.mp` ou, avec Lua $\text{\LaTeX}$  et le package `luamplib`, d'utiliser `\mplibnumbersystem{decimal}`.

**factorielle(n)** Entier égal à  $n!$ .

**binom(n,k)** Entier égal à  $\binom{n}{k}$ .

**binomiale(n,p,k)**  $P(X = k)$  pour  $X$  suivant la loi binomiale de paramètres  $n$  et  $p$ .

**diagrammebinomiale(n,p)** Diagramme en bâtons de la loi binomiale de paramètres  $n$  et  $p$ .

**diagrammeuniforme(n,m)** Diagramme en bâtons de la loi uniforme discrète sur les entiers consécutifs de  $n$  à  $m$ .

**diagrammegeometrique(p)** Diagramme en bâtons de la loi géométrique de paramètre  $p$ .

**diagrammepoisson(lambda)** Diagramme en bâtons de la loi de Poisson de moyenne  $\lambda$ .

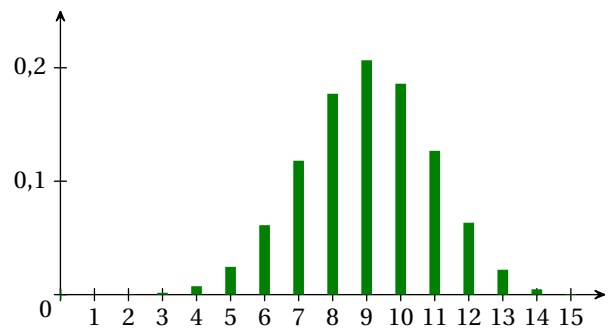
**densitenormale(mu,sigma,a,b)** Courbe représentant la densité de la loi normale de moyenne  $\mu$  et d'écart type  $\sigma$  entre  $a$  et  $b$ . Si  $a$  et  $b$  sont omis, le tracé est fait sur l'intervalle définissant le repère.

**densiteexponentielle(lambda)** Courbe représentant la densité de la loi normale de paramètre  $\lambda$ .



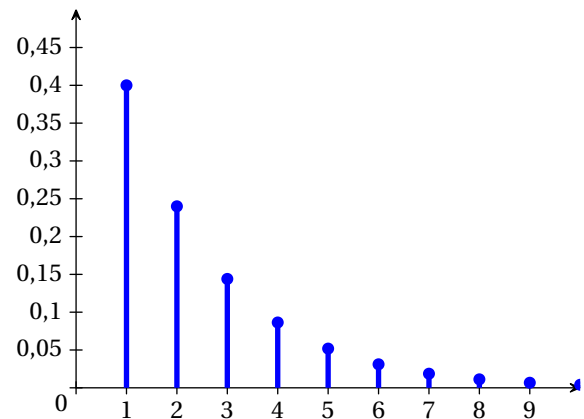
### Exemple 46

```
repere(-2,16,0.45cm,-0.1,0.25,15cm);
setall(0,16,0,0.25);
draw axex(1,1);
draw axey(0.1,0.1);
picture diag;
diampointsbatons:=0;
diag:=diagrammebinomiale(15,0.6);
draw diag withcolor vertfonce epaisseur 4;
fin;
```



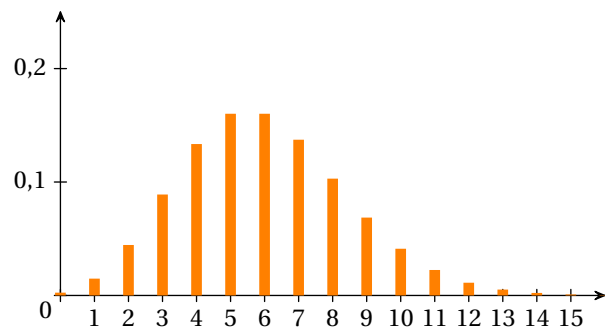
### Exemple 47

```
repere.larg(-2,10,8cm,-0.1,0.5,6cm);
setall(0,10,0,0.5);
draw axex(1,1);
draw axey(0.05,0.05);
draw diagrammegeometrique(0.4)
      epaisseur 2 couleur bleu;
fin;
```



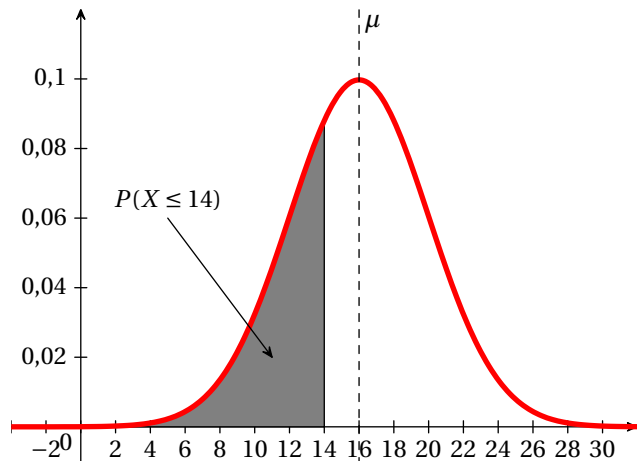
### Exemple 48

```
repere(-2,16,0.45cm,-0.1,0.25,15cm);
setall(0,16,0,0.25);
draw axex(1,1);
draw axey(0.1,0.1);
picture diag;
diampointsbatons:=0;
diag:=diagrammepoisson(6);
draw diag withcolor orange epaisseur 4;
fin;
```



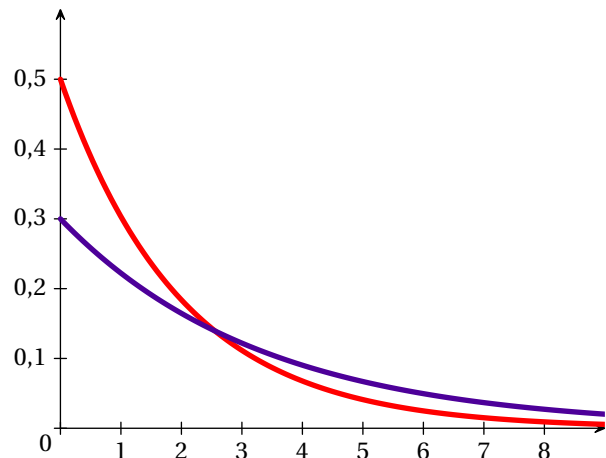
### Exemple 49

```
repere(-4,32,0.23cm,-0.01,0.12,46cm);
draw axex(2,2);
draw axey(0.02,0.02);
path C,d;
C=densitenormale(16,4);
fill souscourbe(C,0,14) couleur gris;
draw souscourbe(C,0,14);
draw C epaisseur 2 couleur rouge;
d=droite(16);
draw d dashed evenly;
drawarrow (5,0.06)--(11,0.02);
label.top("$P(X\leq 14)$",(5,0.06));
nomme(d,"$\mu$");
fin;
```



### Exemple 50

```
repere.larg(-1,9,8cm,-0.05,0.6,6cm);
setall(0,9,0,0.6);
draw axex(1,1);
draw axey(0.1,0.1);
path C,D;
C=densiteexponentielle(0.5);
D=densiteexponentielle(0.3);
draw C epaisseur 2 couleur rouge;
draw D epaisseur 2 couleur violet;
fin;
```



## 9 Géométrie

Certaines des macros suivantes sont largement inspirées des macros de `geometriesyr16.mp` de Christophe POULAIN.

### 9.1 Polygones

`polygone(A,B,C,...)` Chemin fermé représentant le polygone  $ABC\dots$

`triangle(A,B,C)` Cas particulier du précédent. Chemin fermé représentant le triangle  $ABC$ .

`parallélogramme(A,B,C)` Chemin fermé représentant  $ABCD$  où  $D$  est le quatrième point du parallélogramme.

`polygoneregulier(A,B,n)` Chemin fermé représentant le polygone régulier de sens direct à  $n$  côtés dont un des côtés est  $[AB]$ .

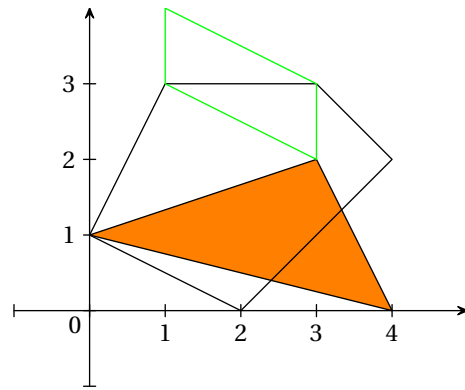
`equilateral(A,B)` Cas particulier du précédent. Triangle équilatéral de sens direct de côté  $[AB]$ .

`carre(A,B)` Autre cas particulier. Carré de sens direct de côté  $[AB]$ .

`sommetpolygoneregulier(A,B,n,i)` Sommet numéro  $i$  du polygone régulier à  $n$  côtés dont un des côtés est  $[AB]$ .  $A$  est le sommet numéro 1 et  $B$  est le sommet numéro 2.

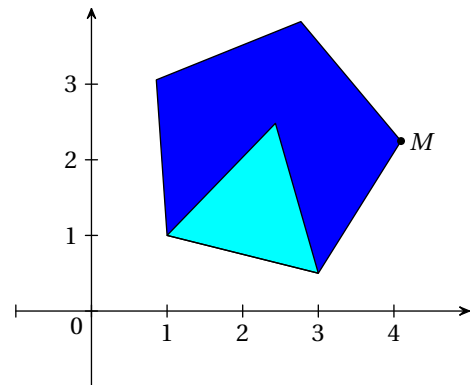
### Exemple 51

```
repere(-1,5,1cm,-1,4,1cm);
draw axes(1,1);
pair A,B,C,D,E,F,G;
A=(0,1);B=(2,0);C=(4,2);D=(3,3);E=(1,3);
F=(4,0);G=(3,2);
fill triangle(A,F,G) withcolor orange;
draw triangle(A,F,G);
draw polygon(A,B,C,D,E);
draw parallelogramme(D,G,E) withcolor vert;
fin;
```



### Exemple 52

```
repere(-1,5,1cm,-1,4,1cm);
draw axes(1,1);
pair A,B,M;
A=(1,1);B=(3,0.5);
fill polygoneregulier(A,B,5) withcolor bleu;
fill equilateral(A,B) withcolor cyan;
draw polygoneregulier(A,B,5);
M=sommetpolygoneregulier(A,B,5,3);
nomme.rt(M);
draw equilateral(A,B);
fin;
```



## 9.2 Cercles et arcs

**cercle(A,B,C)** Cercle circonscrit au triangle  $ABC$ .

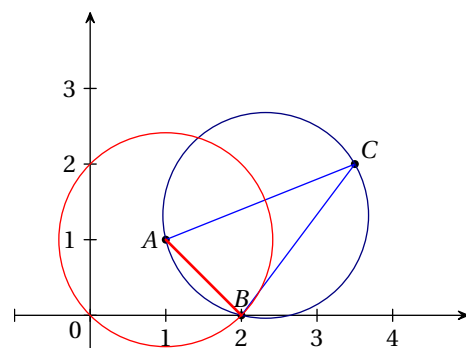
**cercle(O,A)** Cercle de centre  $O$  passant par  $A$ .

**cercle(O,r)** Cercle de centre  $O$  et de rayon  $r$ . L'unité de longueur est l'unité de l'axe des abscisses.

**arccercle(A,O,B)** Arc de cercle de sens direct de centre  $O$ , passant par  $A$  et s'appuyant sur la demi-droite  $[OB)$ .

### Exemple 53

```
repere(-1,5,1cm,-0.5,4,1cm);
draw axes(1,1);
pair A,B,C;
A=(1,1);B=(2,0);C=(3.5,2);
nomme.lft(A);nomme.urc(C);nomme.top(B);
draw triangle(A,B,C) withcolor bleu;
draw cercle(A,B,C) withcolor marine;
draw A--B withcolor rouge epaisseur 1;
draw cercle(A,B) withcolor rouge;
fin;
```

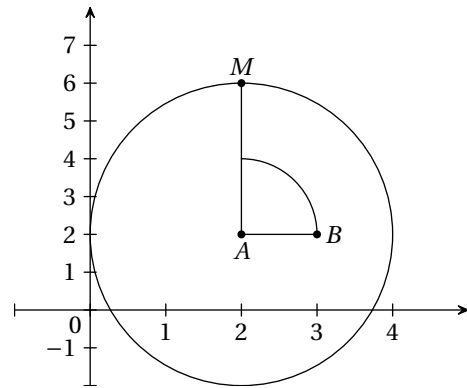


### Exemple 54

```

repere(-1,5,1cm,-2,8,0.5cm);
draw axes(1,1);
pair A,M,B;
A=(2,2);M=(2,6);B=(3,2);
nomme.bot(A);nomme.rt(B);nomme.top(M);
draw B--A--M;
draw cercle(A,2);
draw arccercle(B,A,M);
fin;

```



### 9.3 Codage des segments et des angles

**taille\_marque\_a** Valeur numérique (qui vaut par défaut 0.4cm) donnant le rayon des arcs de cercles servant à marquer les angles.

**sep\_marque\_a** Valeur numérique (qui vaut par défaut 1.5bp) donnant la différence de rayon entre les différents arcs servant à marquer les angles.

**marqueangle(A,O,B,n)** Figure formée de  $n$  arcs de cercle de centre  $O$  et de rayon moyen **taille\_marque\_a** permettant de marquer l'angle géométrique  $\widehat{AOB}$ . Les arcs (si  $n$  est supérieur à 1) sont séparés de **sep\_marque\_a**.

Il s'agit d'un chemin fermé qui peut donc être rempli.

**marqueangle(A,O,B)** Arc de cercle de centre  $O$  et de rayon **taille\_marque\_a** permettant de marquer l'angle orienté avec **drawarrow**.

**nomme.pos(A,O,B,texte)** Place le texte à la position **pos** par rapport au point central de l'arc de cercle de centre  $O$  et de rayon **taille\_marque\_a**. **pos** peut être **rt**, **urt**, **top**, etc. ou un angle donné par rapport à la direction ( $Ox$ ).

**nomme(A,O,B,texte)** Même chose que précédemment mais la position est calculée automatiquement en fonction de l'angle.

**marqueangle droit(A,O,B)** Chemin fermé permettant de marquer l'angle droit  $\widehat{AOB}$  sous forme d'un losange (il s'agit donc d'un carré si l'angle est réellement droit). Le côté du losange est **taille\_marque\_ad** et vaut 0.3cm par défaut.

**marqueselement(A,B,n)** Figure formées de  $n$  marques sur le segment  $[AB]$ . Ces marques ont une taille de **taille\_marque\_s** (0.3cm par défaut), forment un angle en degrés de **angle\_marque\_s** avec le segment (60 par défaut) et sont séparées de **sep\_marque\_s** (2 par défaut).

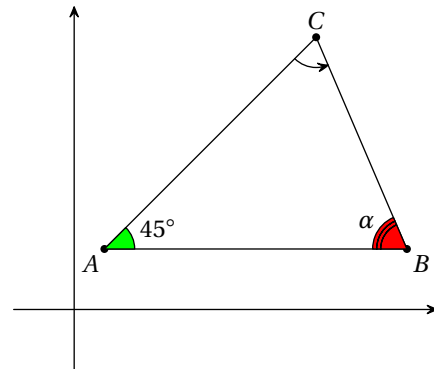
**marqueselement(A,B,C,D,...,n)** Permet de marquer plusieurs segments simultanément.

### Exemple 55

```

repere(-2,12,0.4cm,-2,10,0.4cm);
pair A,B,C;
A=(1,2);B=(11,2);C=(8,9);
draw axes(0,0);
draw triangle(A,B,C);
nomme.llft(A);nomme.lrt(B);nomme.top(C);
fill marqueangle(C,B,A,3) withcolor red;
draw marqueangle(C,B,A,3);
drawarrow marqueangle(A,C,B);
fill marqueangle(B,A,C,1) withcolor vert;
draw marqueangle(B,A,C,1);
nomme[20](B,A,C,"\ang{45}");
nomme(C,B,A,"\alpha");
fin;

```

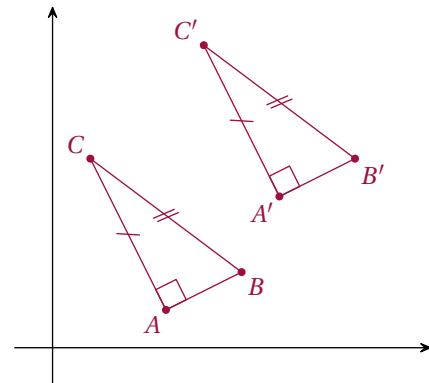


### Exemple 56

```

repere(-1,10,0.5cm,-1,9,0.5cm);
pair A,B,C,A',B',C',u;
A=(3,1);B=(5,2);C=(1,5);u=(3,3);
A'-A=B'-B=C'-C=u;
draw axes(0,0);
drawoptions(withcolor pourpre);
draw triangle(A,B,C);draw triangle(A',B',C');
nomme.llft(A);nomme.lrt(B);nomme.ulft(C);
nomme.llft(A');nomme.lrt(B');nomme.ulft(C');
draw marqueangledroit(B,A,C);
draw marqueangledroit(B',A',C');
draw marquesegment(B,C,2);
draw marquesegment(B',C',2);
draw marquesegment(A,C,A',C',1);
fin;

```



## 9.4 Cotes

**cote(A,B,texte)** Figure formée du texte orienté dans la direction du segment  $[AB]$ , placé au niveau du milieu et situé « sous » le segment.

**cote.top(A,B,texte)** Même chose mais le texte est placé au-dessus du segment.

**angle\_cote** Variable numérique qui fixe l'angle de rotation de l'étiquette. La valeur par défaut de  $-1$  indique que l'étiquette est tournée en fonction de l'angle du segment.

**cotefleche(A,B,texte)** Figure formée d'une double flèche et du texte orienté dans la direction du segment  $[AB]$ , placés au niveau du milieu et situés « sous » le segment.

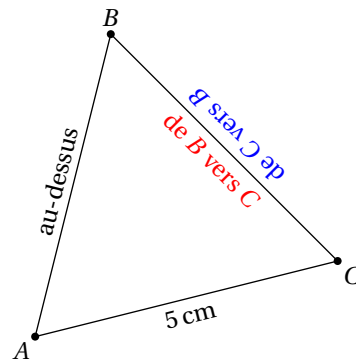
**cotefleche.top(A,B,texte)** Même chose mais la double flèche et le texte sont placés au-dessus du segment.

**dec\_cote** Variable numérique qui indique le décalage entre le segment et la double flèche. La valeur par défaut est 4 mm.

**traits\_cote** Variable booléenne qui indique si des traits délimitant la double flèche doivent être tracés. La valeur par défaut est **false**.

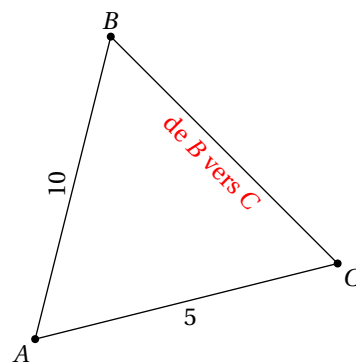
### Exemple 57

```
repere(0,6,1cm,0,6,1cm);
pair A,B,C;
A=(1,1);B=(2,5);C=(5,2);
nomme.llft(A);nomme.top(B);nomme.lrt(C);
draw triangle(A,B,C);
cote(A,C,"\SI{5}{cm}");
cote(B,C,"de B$ vers C$") couleur rouge;
cote(C,B,"de C$ vers B$") couleur bleu;
cote.top(A,B,"au-dessus");
fin;
```



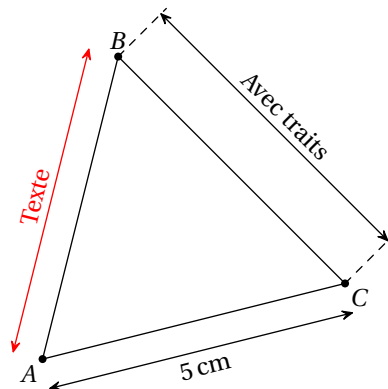
### Exemple 58

```
repere(0,6,1cm,0,6,1cm);
pair A,B,C;
A=(1,1);B=(2,5);C=(5,2);
nomme.llft(A);nomme.top(B);nomme.lrt(C);
draw triangle(A,B,C);
angle_cote:=90;
cote.top(A,B,"10");
angle_cote:=0;
cote(A,C,"5");
angle_cote:=-1;
cote(B,C,"de B$ vers C$") couleur rouge;
fin;
```



### Exemple 59

```
repere(0,6,1cm,0,6,1cm);
pair A,B,C;
A=(1,1);B=(2,5);C=(5,2);
nomme.llft(A);nomme.top(B);nomme.lrt(C);
draw triangle(A,B,C);
cotefleche(A,C,"\SI{5}{cm}");
cotefleche.top(A,B,"Texte") couleur rouge;
traits_cote:=true;
dec_cote:=8mm;
cotefleche.top(B,C,"Avec traits");
fin;
```



## 9.5 Figures complètes

Par rapport aux polygones décrits précédemment, les macros qui suivent diffèrent dans le sens où ce ne sont pas des chemins (paths) mais des figures « complètes ».

D'une part, les points sont définis (selon les données passées par l'utilisateur) puis la figure est tracée avec noms et légendes. Des macros pour ne faire que la définition des points ou que le tracé quand les points sont déjà définis sont décrites plus loin.

Pour les distinguer, leur nom commence par une majuscule.

## Figures

**Segment(A,B)(longueur,ptdepart,angle)(légende)** Figure formée du segment  $[AB]$ , des noms des points et de la cote.

Les points doivent être déclarés mais s'ils n'existent pas, il sont définis de sorte que le segment mesure `longueur`, démarre à `ptdepart` et fasse l'angle `angle` avec l'horizontale. `ptdepart` vaut (0,0) par défaut et `angle` vaut 0.

Les noms des points et la cote peuvent être éventuellement modifiés dans `légende`.

**Vecteur(u,A,B)(longueur,ptdepart,angle)(légende)** Figure formée du représentant du vecteur  $\vec{u}$  d'origine A, des noms des points et du vecteur.

Les points et le vecteur doivent être déclarés mais s'ils n'existent pas, il sont définis de sorte que le vecteur ait une norme de `longueur`, pour origine `ptdepart` et fasse l'angle `angle` avec l'horizontale. `ptdepart` vaut (0,0) par défaut et `angle` vaut 0.

Les noms des points et du vecteur peuvent être éventuellement modifiés dans `légende`.

**TriangleLLL(A,B,C)(longAB,longAC,longBC,ptdepart,angle)(points)(cotes)(angles)** Figure formée du triangle  $ABC$ , donné par trois longueurs, et des noms des points, cotes et angles. `ptdepart` vaut (0,0) par défaut et `angle` vaut 0.

Les étiquettes sont composées automatiquement mais peuvent être modifiées en passant des valeurs à `(points)`, `(cotes)` ou `(angles)`.

**TriangleLLA(A,B,C)(longAB,longAC,angleA,ptdepart,angle)(points)(cotes)(angles)** Figure formée du triangle  $ABC$ , donné par deux longueurs et un angle, et des noms des points, cotes et angles. `ptdepart` vaut (0,0) par défaut et `angle` vaut 0.

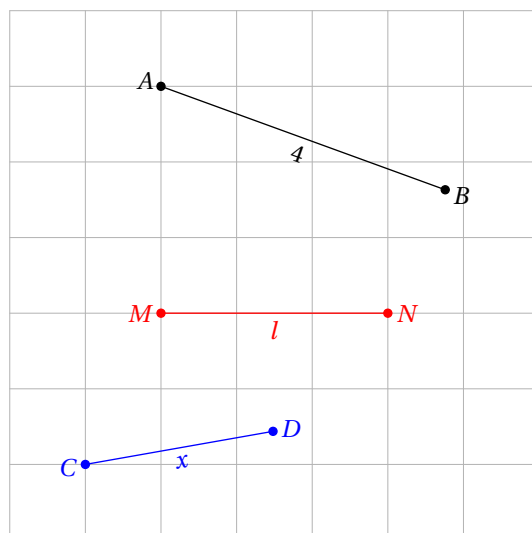
Les étiquettes sont composées automatiquement mais peuvent être modifiées en passant des valeurs à `(points)`, `(cotes)` ou `(angles)`.

**TriangleLAA(A,B,C)(longAB,angleA,angleB,ptdepart,angle)(points)(cotes)(angles)** Figure formée du triangle  $ABC$ , donné par une longueur et deux angles, et des noms des points, cotes et angles. `ptdepart` vaut (0,0) par défaut et `angle` vaut 0.

Les étiquettes sont composées automatiquement mais peuvent être modifiées en passant des valeurs à `(points)`, `(cotes)` ou `(angles)`.

## Exemple 60

```
repere(-1,6,1cm,-1,6,1cm);
pair A,B,C,D,E,F;
draw quadrillage(1,1);
draw Segment(A,B)(4,(1,5),-20)();
draw Segment(C,D)(2.52,10)("$x$")
                                couleur bleu;
E:=(1,2);
draw Segment(E,F)(3)("$l$","$M$","$N$")
                                couleur rouge;
fin;
```

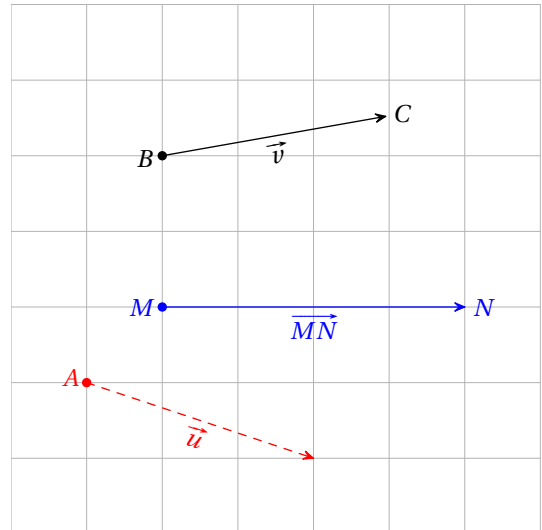


## Exemple 61

```

repere(-1,6,1cm,-1,6,1cm);
pair A,B,C,D,E,F,u,v,w;
draw quadrillage(1,1);
draw Vecteur(v,B,C)(3,(1,4),10)();
A:=(0,1);u:=(3,-1);
draw Vecteur(u,A)()() couleur rouge
                        dashed evenly;
D:=(1,2);
draw Vecteur(w,D)(4)("$\vv{MN}$","$M$","$N$")
                        couleur bleu;
fin;

```

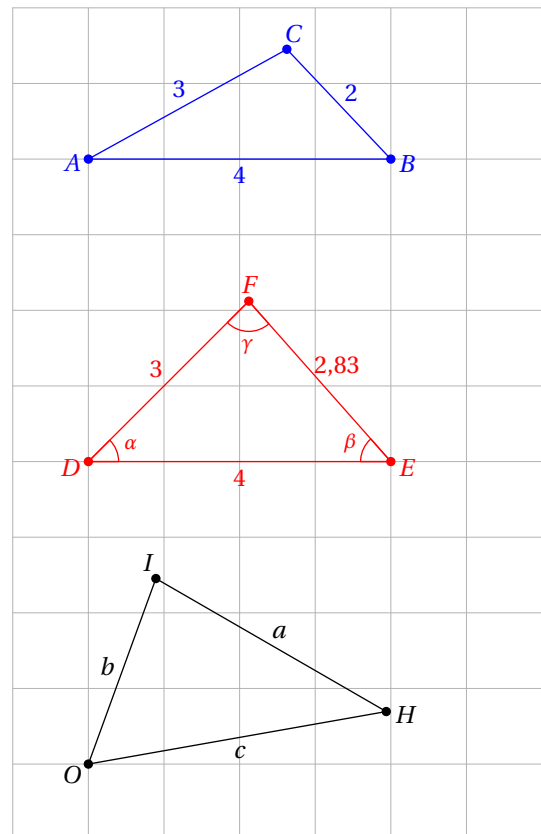


## Exemple 62

```

repere(-1,6,1cm,-5,6,1cm);
pair A,B,C,D,E,F,G,H,I;
angle_cote:=0;
draw quadrillage(1,1);
draw TriangleLLL(A,B,C)(4,3,2,(0,4))()()()
                        couleur bleu;
draw TriangleLLA(D,E,F)(4,3,45)
                        ()()("$\alpha$","$\beta$","$\gamma$")
                        couleur rouge;
G:=(0,-4);
draw TriangleLAA(G,H,I)(4,60,40,10)
                        ("0$")("$a$","$b$","$c$")();
fin;

```



## Paramètres

**AffPoints** Booléen qui indique si les noms des points doivent être affichés. La valeur par défaut est **true**. Les noms sont affichés quelle que soit la valeur de **AffPoints** si le nom est donné explicitement.

**AffCotes** Booléen qui indique si la cote des segment doit être affichée. La valeur par défaut est **true**. La cote est affichée quelle que soit la valeur de **AffCotes** si une cote est donnée explicitement.

**UnitCotes** Chaîne de caractère qui indique l'unité à rajouter à la cote lorsqu'elle est composée automatiquement. La valeur par défaut est la chaîne vide.



**ArrondiCotes** Valeur entière qui indique à quelle décimale la cote doit être arrondie lorsqu'elle est affichée automatiquement.

**AffVect** Booléen qui indique si le nom du vecteur doit être affiché. La valeur par défaut est **true**. Le nom est affiché quelle que soit la valeur de **AffVect** s'il est donné explicitement.

**AffAngles** Booléen qui indique si les angles doivent être marqués et leur nom affiché. La valeur par défaut est **false**. Les noms sont affichés quelle que soit la valeur de **AffAngles** si le nom est donné explicitement.

**ArrondiAngles** Valeur entière qui indique à quelle décimale l'angle doit être arrondi lorsqu'il est affiché automatiquement.

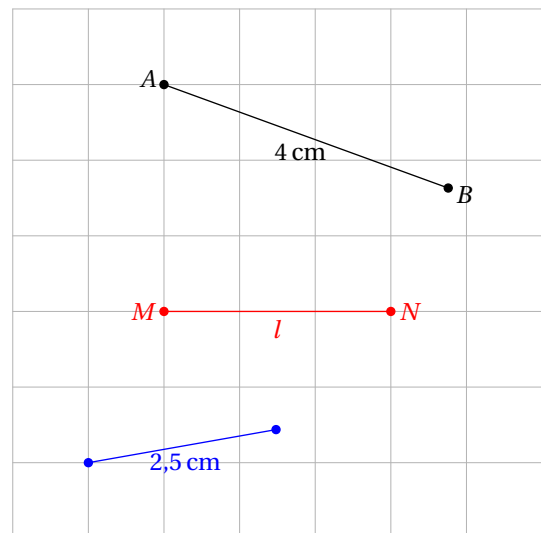
**EchelleAngles** Valeur numérique indiquant le facteur d'échelle de l'affichage des angles. La valeur par défaut est 0.8.

**AutoAngleDroit** Booléen qui indique si les angles droits doivent être marqués comme tels. La valeur par défaut est **false**.

**AffDonnees** Booléen qui permet de n'afficher que les valeurs des côtés et des angles correspondant aux données. La valeur par défaut est **false**.

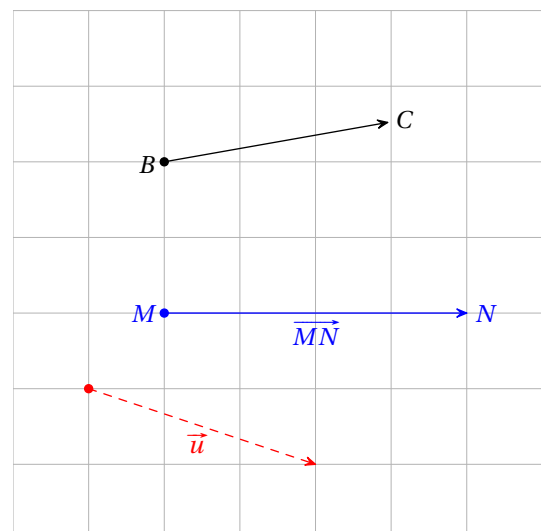
### Exemple 63

```
repere(-1,6,1cm,-1,6,1cm);
pair A,B,C,D,E,F;
draw quadrillage(1,1);
UnitCotes:="cm";
draw Segment(A,B)(4,(1,5),-20)();
AffPoints:=false;
ArrondiCotes:=1;
draw Segment(C,D)(2.52,10)() couleur bleu;
E:=(1,2);
draw Segment(E,F)(3)("$l$","$M$","$N$")
                                couleur rouge;
fin;
```



### Exemple 64

```
repere(-1,6,1cm,-1,6,1cm);
pair A,B,C,D,u,v,w;
draw quadrillage(1,1);
AffVect:=false;
draw Vecteur(v,B,C)(3,(1,4),10)();
A:=(0,1);u:=(3,-1);
AffVect:=true;
AffPoints:=false;
draw Vecteur(u,A)()() couleur rouge
                                dashed evenly;
D:=(1,2);
draw Vecteur(w,D)(4)("$\vv{MN}$","$M$","$N$")
                                couleur bleu;
fin;
```



### Exemple 65

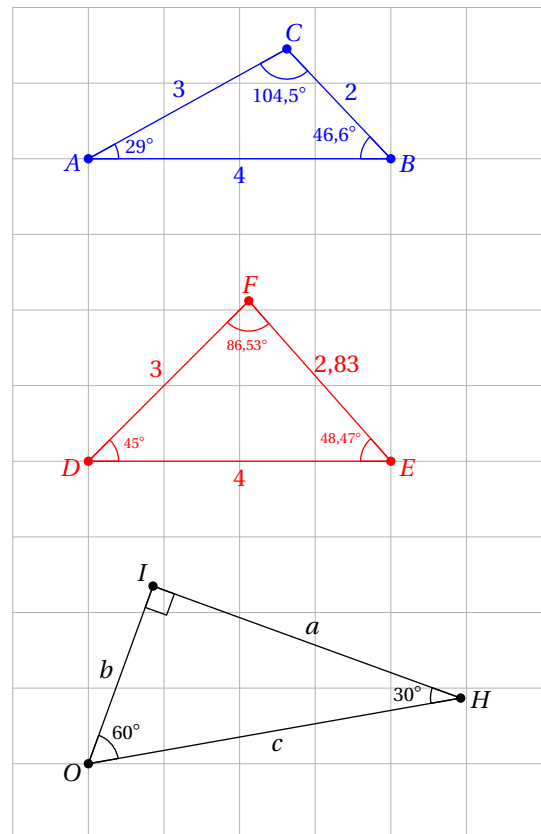
```

repere(-1,6,1cm,-5,6,1cm);
pair A,B,C,D,E,F,G,H,I;
draw quadrillage(1,1);
angle_cote:=0;
AffAngles:=true;
draw TriangleLLL(A,B,C)(4,3,2,(0,4))()()()
                                couleur bleu;

ArrondiAngles:=2;
EchelleAngles:=0.6;
draw TriangleLLA(D,E,F)(4,3,45)()()()
                                couleur rouge;

G:=(0,-4);
EchelleAngles:=0.8;
AutoAngleDroit:=true;
draw TriangleLAA(G,H,I)(5,60,30,10)
                        ("O")("$a$","$b$","$c$")();
fin;

```



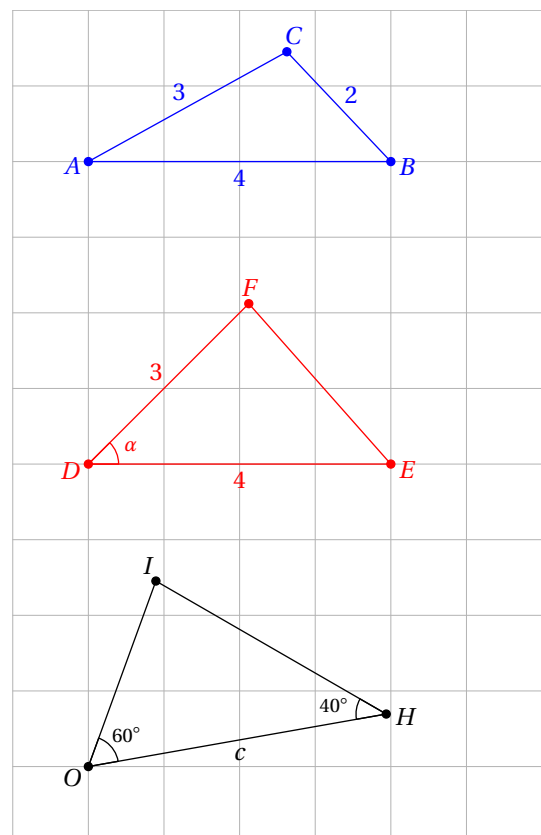
### Exemple 66

```

repere(-1,6,1cm,-5,6,1cm);
pair A,B,C,D,E,F,G,H,I;
angle_cote:=0;
AffDonnees:=true;
draw quadrillage(1,1);
draw TriangleLLL(A,B,C)(4,3,2,(0,4))()()()
                                couleur bleu;
draw TriangleLLA(D,E,F)(4,3,45)
                        ()("$\alpha$","$\beta$","$\gamma$")
                                couleur rouge;

G:=(0,-4);
draw TriangleLAA(G,H,I)(4,60,40,10)
                        ("O")("$a$","$b$","$c$")();
fin;

```



## Définitions seules

**defSegment(A,B)(longueur,ptdepart,angle)** Macro permettant de définir les points  $A$  et  $B$  sans dessiner le segment.

**defVecteur(u,A,B)(longueur,ptdepart,angle)** Macro permettant de définir les points  $A$  et  $B$  et le vecteur  $\vec{u}$  sans les dessiner.

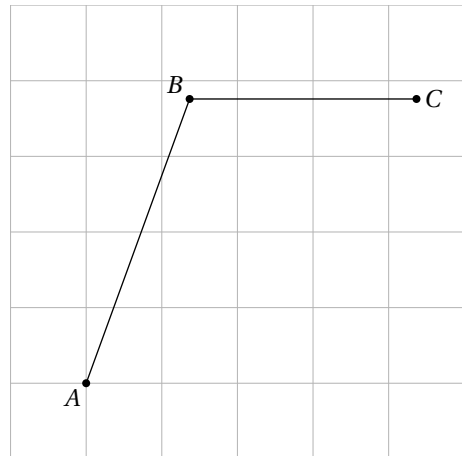
**defTriangleLLL(A,B,C)(longAB,longAC,longBC,ptdepart,angle)** Macro permettant de définir les points  $A$ ,  $B$  et  $C$  sans dessiner le triangle.

**defTriangleLLA(A,B,C)(longAB,longAC,angleA,ptdepart,angle)** Macro permettant de définir les points  $A$ ,  $B$  et  $C$  sans dessiner le triangle.

**defTriangleLAA(A,B,C)(longAB,angleA,angleB,ptdepart,angle)** Macro permettant de définir les points  $A$ ,  $B$  et  $C$  sans dessiner le triangle.

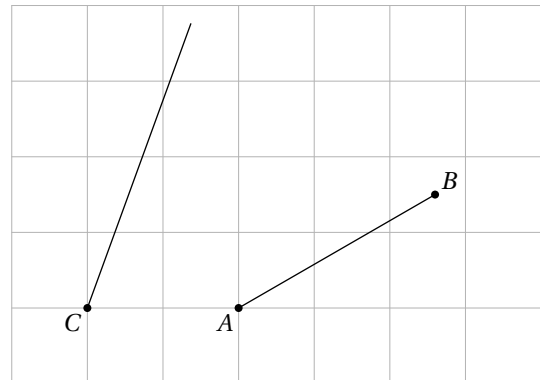
### Exemple 67

```
repere(0,6,1cm,0,6,1cm);
pair A,B,C,D,E,F;
draw quadrillage(1,1);
A:=(1,1);
defSegment(A,B)(4,70);
defSegment(B,C)(3);
draw A--B--C;
nomme.llft(A);nomme.ulft(B);
nomme.rt(C);
fin;
```



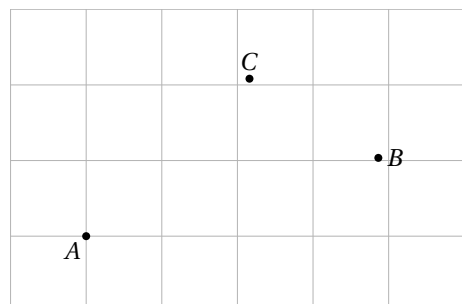
### Exemple 68

```
repere(-1,6,1cm,0,5,1cm);
pair A,B,C,D,E,F,u,v,w,ve;
draw quadrillage(1,1);
defVecteur(u,A,B)(3,(2,1),30);
draw A--B;
nomme.llft(A);nomme.urt(B);
defVecteur(v)(4,70);
C:=(0,1);
draw C--(C+v);
nomme.llft(C);
fin;
```



### Exemple 69

```
repere(-1,5,1cm,-1,3,1cm);
pair A,B,C;
draw quadrillage(1,1);
defTriangleLLL(A,B,C)(4,3,2,15);
nomme.llft(A);nomme.rt(B);
nomme.top(C);
fin;
```



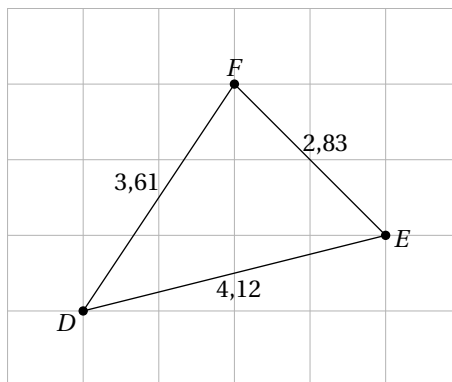
## Tracé seul

Si les points sont définis, on peut utiliser la macro de tracé ci-dessous.

**Triangle(A,B,C)(points)(cotes)(angles)** Figure formée du triangle  $ABC$  et des noms des points, cotes et éventuellement des angles.

### Exemple 70

```
repere(-1,5,1cm,-1,4,1cm);
pair D,E,F;
angle_cote:=0;
draw quadrillage(1,1);
D:=(0,0);E:=(4,1);F:=(2,3);
draw Triangle(D,E,F)()()();
fin;
```



## 10 Divers

### 10.1 Composition des étiquettes

Tous les textes et étiquettes peuvent être composés en utilisant la macro ci-dessous.

**LaTeX(ch)** Figure formée de la chaîne **ch** composée avec  $\text{\LaTeX}$  et mise à l'échelle **defaultscale**.

Cette macro utilise la commande **texttext** de **luamplib** dans le cas de l'utilisation de  $\text{\LuaLaTeX}$  et **texttext** de **latexmp** dans le cas d'une compilation **METAPOST** « standard ». Ce dernier cas nécessite alors deux compilations.

### Exemple 71

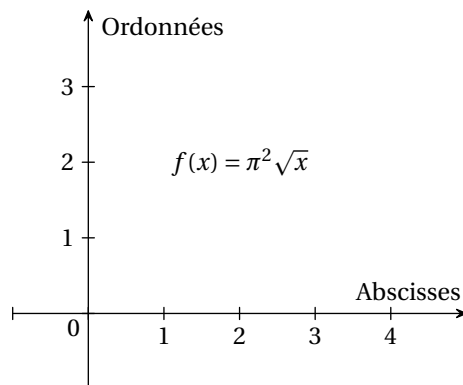
```
repere(-1,7,1cm,-1,1,1cm);
for i=2 upto 6:
label(LaTeX("\frac{1}{"&decimal(i)&"}$"),(i,0));
endfor
fin;
```

$\frac{1}{2}$     $\frac{1}{3}$     $\frac{1}{4}$     $\frac{1}{5}$     $\frac{1}{6}$

**label.pos(fig,point)** Commande de **METAPOST** qui permet de placer la figure **fig** au niveau du point **point**.

### Exemple 72

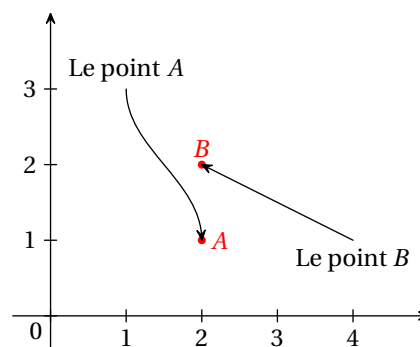
```
repere(-1,5,1cm,-1,4,1cm);
draw axes(1,1);
label.ulft("Abscisses",(5,0.1));
label.lrt("Ordonnées",(0.1,4));
label("$f(x)=\pi^2\sqrt{x}$",(2,2));
fin;
```



**legende.pos(fig,p)** Figure formée du chemin p dessiné avec une flèche et de la figure ou de la chaîne fig située à la position pos par rapport au premier point du chemin.

### Exemple 73

```
repere(-0.5,5,1cm,-0.5,4,1cm);
draw axes(1,1);
pair A,B;
A=(2,1);B=(2,2);
nomme.rt(A) couleur rouge;
nomme.top(B) couleur rouge;
legende.top("Le point $A$", (1,3){down}..{down}A);
legende.bot("Le point $B$", (4,1)--B);
fin;
```



## 10.2 Couleurs

Certaines couleurs sont définies par leur nom et peuvent être utilisées directement :

<span style="display:inline-block; width:15px; height:15px; background-color:red;"></span> rouge	<span style="display:inline-block; width:15px; height:15px; background-color:limegreen;"></span> vert	<span style="display:inline-block; width:15px; height:15px; background-color:blue;"></span> bleu	<span style="display:inline-block; width:15px; height:15px; background-color:cyan;"></span> cyan	<span style="display:inline-block; width:15px; height:15px; background-color:magenta;"></span> magenta	<span style="display:inline-block; width:15px; height:15px; background-color:yellow;"></span> jaune	<span style="display:inline-block; width:15px; height:15px; background-color:black;"></span> noir
<span style="display:inline-block; width:15px; height:15px; background-color:brown;"></span> marron	<span style="display:inline-block; width:15px; height:15px; background-color:lightgreen;"></span> lime	<span style="display:inline-block; width:15px; height:15px; background-color:orange;"></span> orange	<span style="display:inline-block; width:15px; height:15px; background-color:pink;"></span> rose	<span style="display:inline-block; width:15px; height:15px; background-color:maroon;"></span> pourpre	<span style="display:inline-block; width:15px; height:15px; background-color:olive;"></span> olive	<span style="display:inline-block; width:15px; height:15px; background-color:purple;"></span> violet
<span style="display:inline-block; width:15px; height:15px; background-color:tan;"></span> beige	<span style="display:inline-block; width:15px; height:15px; background-color:darkblue;"></span> marine	<span style="display:inline-block; width:15px; height:15px; background-color:yellowgreen;"></span> moutarde	<span style="display:inline-block; width:15px; height:15px; background-color:lightgrey;"></span> grisclair	<span style="display:inline-block; width:15px; height:15px; background-color:grey;"></span> gris	<span style="display:inline-block; width:15px; height:15px; background-color:darkgrey;"></span> grisfoncé	<span style="display:inline-block; width:15px; height:15px; background-color:green;"></span> vertfoncé

Toutes ces couleurs sont définies selon le modèle « rgb ». Pour les obtenir selon le modèle « cmyk », remplacer la première lettre par une majuscule.

## 10.3 Remplissage

Pour remplir des chemins fermés avec autre chose que de la couleur, **repere** permet l'utilisation de la syntaxe **fill p avec motif** où **motif** est un des motifs décrits ci-dessous. Cette instruction peut être complétée par des options de dessin (**withpen**, **withcolor...**).

**hachures(pas,angle)** hachures espacées de **pas** et formant un angle en degrés de **angle** avec l'horizontale. Si les valeurs sont omises, **pas** vaut 5 et **angle** vaut 60.

**briques(larg,haut,dec)** briques de largeur **larg**, de hauteur **haut** et décalées d'une ligne à l'autre de **dec**. Si les valeurs sont omises, **larg** vaut 12, **haut** vaut 6 et **dec** vaut 6.

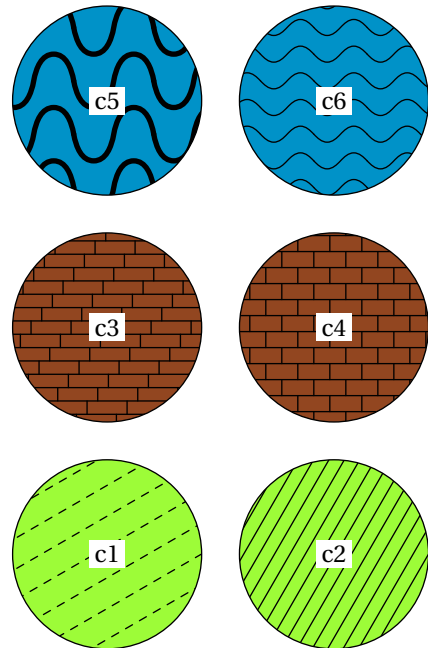
**vagues(per,amp,dec)** (d'après le manuel de l'utilisateur) « vagues » de période **per**, d'amplitude **amp** et décalées d'une ligne à l'autre de **dec**. Si les valeurs sont omises, **per** vaut 20, **amp** vaut 3 et **dec** vaut 10.

### Exemple 74

```

repere(-1.5,4.5,1cm,-1.5,7.5,1cm);
path c[];picture lab;
c1=fullcircle scaled 2.5;
for k=1 upto 6:
  i:=(k-1) mod 2;j:=(k-1) div 2;
  c[k]:=c1 shifted (3*i,3*j);
endfor;
fill c1 withcolor lime;
fill c1 avec hachures(10,30) dashed evenly;
fill c2 withcolor lime;
fill c2 avec hachures();
fill c3 withcolor (0,0.65,0.8,0.48);
fill c3 avec briques(15,5,4);
fill c4 withcolor (0,0.65,0.8,0.48);
fill c4 avec briques();
fill c5 withcolor (1,0,0,0.2);
fill c5 avec vagues(30,10,20)
      withpen pencircle scaled 2;
fill c6 withcolor (1,0,0,0.2);
fill c6 avec vagues();
for k=1 upto 6:
  i:=(k-1) mod 2;j:=(k-1) div 2;
  draw c[k];
  lab:=thelabel("c"&decimal(k),3*(i,j));
  unfill bbox lab;draw lab;
endfor;
fin;

```



## 10.4 Figures pour une présentation

**figureinter** exporte la figure telle qu'elle est au moment où cette commande apparaît. La numérotation est incrémentée et la figure peut continuer.

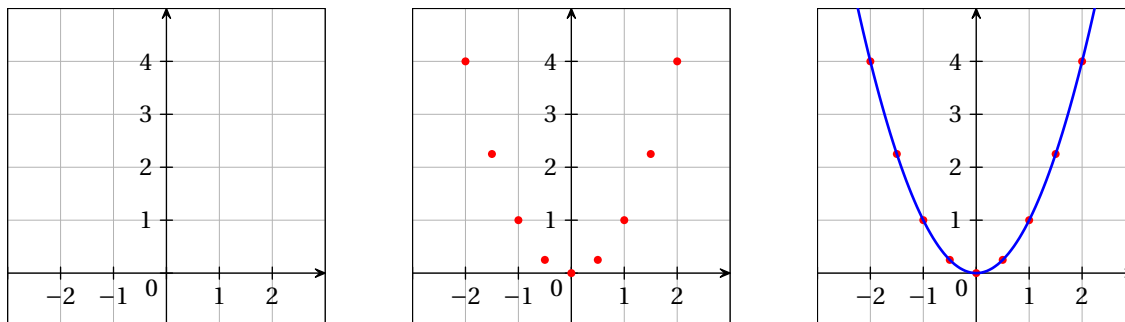
L'exemple ci-dessous crée trois figures :

### Exemple 75

```

repere(-3,3,0.7cm,-1,5,0.7cm);
path C_f;
vardef f(expr x)=x**2 enddef;
C_f= courbefonc(f)();
draw quadrillage(1,1);
draw axes(1,1);
draw cadre;
figureinter;
draw courbepoints(f)(-2,2,9) withcolor rouge;
figureinter;
draw C_f withcolor bleu withpen pencircle scaled 1;
fin;

```



Si ces trois figures s'appellent `mafigure.1`, `mafigure.2` et `mafigure.3`, elles peuvent être incluses dans un document de la classe `beamer` avec le code ci-dessous :

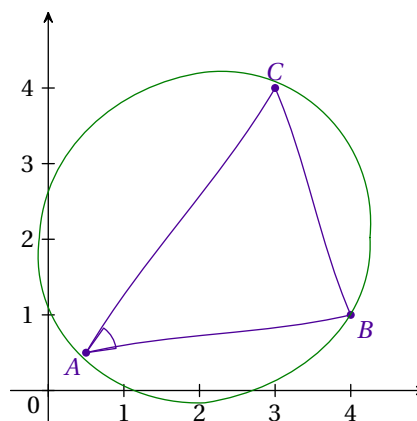
```
\documentclass{beamer}
\ifpdf % Pour utiliser pdflatex
\DeclareGraphicsRule{*}{mps}{*}{}
\fi
\begin{document}
\begin{frame}
\includegraphics<+>{mafigure.1}%
\includegraphics<+>{mafigure.2}%
\includegraphics<+>{mafigure.3}%
\end{frame}
\end{document}
```

## 11 Dessin à main levée avec `geometriesyr`

Il est possible, dans une figure créée avec `repere`, d'utiliser le « dessin à main levée » de `geometriesyr`. Il faut alors charger `geometriesyr` *avant* `repere` et utiliser les fonctions de dessin telles que `cercles`, `triangle`...

### Exemple 76

```
repere(-0.5,5,1cm,-0.5,5,1cm);
pair A,B,C,D;
A=(0.5,0.5);B=(4,1);C=(3,4);
typetrace="mainlevee";
draw axes(1,1);
drawoptions(withcolor violet);
draw triangle(A,B,C);
nomme.llft(A);nomme.lrt(B);
nomme.top(C);
draw marqueangle(B,A,C,1);
drawoptions(withcolor vertfonce);
draw cercles(CentreCercleC(A,B,C),A);
fin;
```



## Deuxième partie

# Tableaux et grilles

Il est possible d'utiliser `repere` pour représenter des tableaux ou « damiers » et placer des objets dans chacune des cases. La figure devra alors débiter par une commande `tableau` au lieu de la commande `repere` et la numérotation sera automatique.

## 12 Définition et grilles

### 12.1 Définition du tableau

`tableau(n,m,u)` débute une figure et définit un tableau de  $n$  colonnes et  $m$  lignes. La largeur des colonnes est égale à la hauteur des lignes et vaut  $u$ .

`tableau(n,m,ux,uy)` débute une figure et définit un tableau de  $n$  colonnes et  $m$  lignes. La largeur des colonnes vaut  $ux$  et la hauteur des lignes vaut  $uy$ .

`fin` termine la figure.

### 12.2 Grille

`grille(x,y)` quadrillage avec un pas de  $x$  sur les colonnes et de  $y$  sur les lignes.

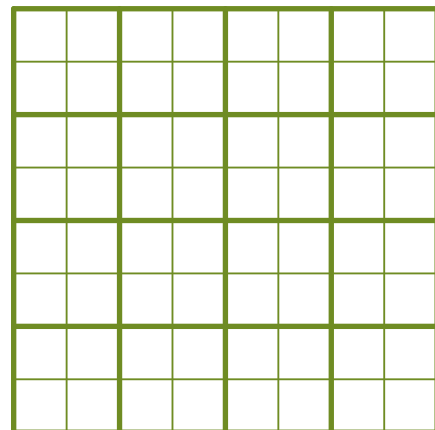
La couleur et l'épaisseur des lignes peuvent être modifiées localement mais on peut aussi les changer globalement :

`coullignes` variable contenant la couleur des lignes de la grille. La valeur par défaut est `black`.

`eplignes` variable contenant l'épaisseur des lignes de la grille. La valeur par défaut est `0.7bp`.

#### Exemple 77

```
tableau(8,8,0.7cm);
coullignes:=olive;
draw grille(1,1);
draw grille(2,2) epaisseur 2;
fin;
```



`damier(x,y)` figure formée de la grille et des cases coloriées alternativement.

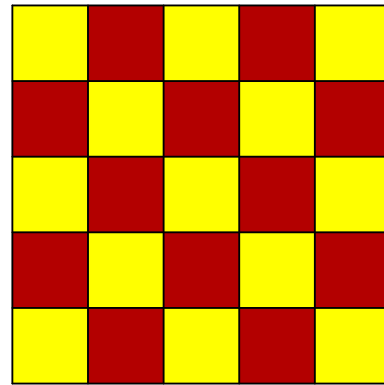
`coultableau[]` variables qui contiennent les couleurs des cases.

Les valeurs par défaut sont `coultableau[1]=0.45white` et `coultableau[2]:=0.9white`.



## Exemple 78

```
tableau(5,5,1cm);  
  coultableau[1]:=jaune;  
  coultableau[2]:=0.7rouge;  
  draw damier(1,1);  
fin;
```



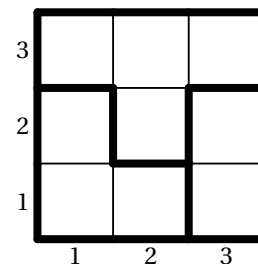
## 12.3 Grille partielle

**lignesh(<suite de 0 et 1>)** Figure qui décrit les lignes horizontales qui doivent être tracées. De gauche à droite et de haut en bas.

**lignesv(<suite de 0 et 1>)** Figure qui décrit les lignes verticales qui doivent être tracées. De gauche à droite et de haut en bas.

## Exemple 79

```
tableau(3,3,1cm);  
  draw grille(1,1);  
  draw coord;  
  draw lignesh(1,1,1,  
               1,0,1,  
               0,1,0,  
               1,1,1) epaisseur 3;  
  draw lignesv(1,0,0,1,  
               1,1,1,1,  
               1,0,1,1) epaisseur 3;  
fin;
```



## 13 Repérage et cases

### 13.1 Coordonnées

**coordx** figure formée des numéros de colonnes placée par défaut en bas.

**coordy** figure formée des numéros de lignes placée par défaut à gauche.

**coord** figure formée des deux figures précédentes.

### Exemple 80

```
tableau(5,5,1cm);  
  draw grille(1,1);  
  draw coord;  
fin;
```

5					
4					
3					
2					
1					
	1	2	3	4	5

**style\_coord\_x** variable de type **string** qui indique comment doivent être composées les coordonnées des colonnes. La valeur par défaut est "1". Les autres valeurs possibles sont "a" (lettres minuscules), "A" (lettres majuscules), "i" (chiffres romains minuscules) et "I" (chiffres romains majuscules).

**style\_coord\_y** même chose pour les lignes.

**deb\_coord\_x** variable de type **numeric** qui indique la première valeur des colonnes. La valeur par défaut est 1.

**deb\_coord\_y** même chose pour les lignes.

**align\_coord\_y** variable de type **string** qui indique comment doivent être alignés les numéros de lignes. La valeur par défaut est "c" (centré). Les valeurs possibles sont "g" ou "l" (gauche) et "d" ou "r" (droite).

**place\_coord** variable de type **string** qui indique où les coordonnées doivent être affichées. La valeur par défaut est "bg" (en bas à gauche). On peut utiliser une combinaison des lettres "b" (en bas), "h" (en haut), "g" (à gauche) et "d" (à droite).

**inverse\_coord\_x** variable de type **boolean** si le sens par défaut dans lequel les numéros de colonnes doivent être écrits (de gauche à droite) doit être inversé. La valeur par défaut est **false**.

**inverse\_coord\_y** variable de type **boolean** si le sens par défaut dans lequel les numéros de lignes doivent être écrits (de bas en haut) doit être inversé. La valeur par défaut est **false**.

### Exemple 81

```
tableau(5,5,1cm);  
  draw grille(1,1);  
  style_coord_x:="a";  
  style_coord_y:="I";  
  deb_coord_x:=5;  
  deb_coord_y:=3;  
  draw coord;  
fin;
```

VII					
VI					
V					
IV					
III					
	e	f	g	h	i

### Exemple 82

```

tableau(5,5,1cm);
  draw grille(1,1);
  style_coord_x:="A";
  style_coord_y:="I";
  place_coord:="hg";
  inverse_coord_y:=true;
  align_coord_y:="r";
  draw coord;
fin;

```

	A	B	C	D	E
I					
II					
III					
IV					
V					

[numerotationdames](#) description

### 13.2 Placements d'objets dans les cases

**case(x,y)** figure formée de la case de coordonnées (x,y) remplie et de son contour dessiné avec l'épaisseur et la couleur de la grille.

**cases((x1,y1),(x2,y2),...)** toutes les cases (x1,y1), (x2,y2)... sont coloriées.

### Exemple 83

```

tableau(5,5,1cm);
  draw damier(1,1);
  draw coord;
  draw case(2,4) couleur rouge;
  draw cases((1,2),(3,3),(5,1)) couleur bleu;
fin;

```

5					
4					
3					
2					
1					
	1	2	3	4	5

On peut utiliser une variante de **label** pour placer des objets dans les cases :

**tablabel(obj,x,y)** Place l'objet **obj** dans la case de coordonnées (x,y). L'objet peut être une chaîne de caractère, un chemin ou une figure.

**tablabels(obj)((x1,y1),(x2,y2)...)**  Place l'objet **obj** dans toutes les cases indiquées.

### Exemple 84

```

tableau(8,8,0.6cm);
  draw grille(1,1);
  draw coord;
  tablabel(fullcircle scaled 0.8,5,3);
  tablabels("X")((3,4),(6,1),(2,2)) couleur vert;
  for i=5 upto 8:
    for j=5 upto 8:
      tablabel(decimal(i*j),i,j) couleur bleu;
    endfor
  endfor
fin;

```

8					40	48	56	64
7					35	42	49	56
6					30	36	42	48
5					25	30	35	40
4			X					
3					○			
2		X						
1						X		
	1	2	3	4	5	6	7	8

### Exemple 85


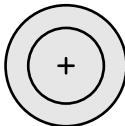



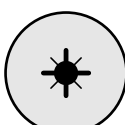

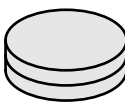
```

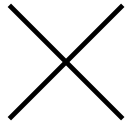
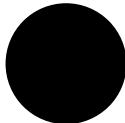


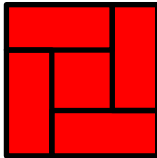





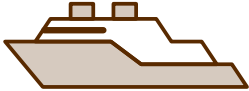
tableau(5,5,0.8cm);
draw grille(1,1);
inverse_coord_x:=true;
inverse_coord_y:=true;
deb_coord_x:=3;
deb_coord_y:=4;
draw coord;
tablables("X")((3,4),(6,7),(4,8));
fin;



```

4					X
5					
6					
7		X			
8				X	
	7	6	5	4	3

## 14 Quelques dessins

Syntaxe	Exemples	Résultat
Pion(v)(cou11,coul2)	Pion(1)(noir,0.85blanc)	
	Pion(1)(0.9blanc,noir)	
	Pion(2)(noir,0.85blanc)	
	Pion(2)(0.9blanc,noir)	
Dame(v)(cou11,coul2)	Dame(1)(noir,0.85blanc)	
	Dame(1)(0.9blanc,noir)	
	Dame(2)(noir,0.85blanc)	
	Dame(2)(0.9blanc,noir)	

Syntaxe	Exemples	Résultat
Croix		
Trou		
Marque(v)(coul)	Marque(1)((0.9,0.7,0.7))	
	Marque(2)((0.8,0.65,0.5))	
Mur(coul1,coul2)	Mur(rouge,noir)	
Plot(coul1,coul2)	Plot(0.7rouge,0.5rouge)	
Caisse(coul1,coul2)	Caisse(orange,marron)	
Robotdroite(coul1,coul2)	Robotdroite(noir,0.6vert)	
Robotgauche(coul1,coul2)	Robotgauche(marine,0.6rouge)	
Bateau(numcases,coul)	Bateau(1,violet)	
	Bateau(2,marron)	

Syntaxe	Exemples	Résultat
Eau(coul)	Eau(marine)	
Fleche(coul1,coul2)	Fleche(jaune,marine)	

## 15 Exemples

### 15.1 Mots croisés

#### Exemple 86

```
def moth(expr ch,n,m)=
  for i=1 upto length ch:
    tablabel(substring(i-1,i) of ch,n+i-1,m);
  endfor
enddef;

tableau(8,8,0.8cm);
draw grille(1,1);
inverse_coord_y:=true;
style_coord_y="I";
place_coord:="hg";
draw coord;
draw cases((3,3),(3,5),(2,7),(6,6),(7,4));
moth("OMELETTE",1,2);
moth("AMIES",4,3);
fin;
```

	1	2	3	4	5	6	7	8
I								
II	O	M	E	L	E	T	T	E
III				A	M	I	E	S
IV								
V								
VI								
VII								
VIII								

## 15.2 Sudokus

### Exemple 87

```

tableau(9,9,0.8cm);
draw grille(1,1);
draw grille(3,3) epaisseur 2;
tablabeled("1",(1,4),(4,8),(6,5),(7,3));
tablabeled("2",(3,9),(5,6),(6,2));
tablabeled("3",(1,1),(4,2),(9,6));
tablabeled("4",(3,8),(8,7));
tablabeled("5",(2,7),(4,9),(6,6),(7,1));
tablabeled("6",(2,4),(5,8));
tablabeled("7",(6,3),(7,2),(8,9),(9,5));
tablabeled("8",(1,3),(3,7),(5,5),(8,8));
tablabeled("9",(3,2));
fin;

```

		2	5				7	
		4	1	6			8	
	5	8					4	
				2	5			3
				8	1			7
1	6							
8					7	1		
		9	3		2	7		
3						5		











## 15.3 Dames

### Exemple 88

```

tableau(10,10,0.8cm);
picture pionblanc,pionnoir,dameblanche;
picture fleche;
pionblanc:=Pion(2)(0.9blanc,noir);
pionnoir:=Pion(2)(noir,0.85blanc);
dameblanche:=Dame(2)(0.9blanc,noir);
coultableau[1]:=0.7marron+0.3blanc;
coultableau[2]:=beige;
fleche:=Fleche(rouge,noir) rotated 135;
draw damier(1,1);
draw numerotationdames;
tablabeled(pionblanc,31,36,39);
tablabeled(pionnoir,3,7,16,19,23,30);
tablabeled(dameblanche,15);
tablabel(fleche,5.5,5.5);
fin;

```

	1		2				4		5
6				8		9		10	
	11		12		13		14		
		17		18				20	
	21		22				24		25
26		27		28		29			
			32		33		34		35
		37		38				40	
	41		42		43		44		45
46		47		48		49		50	

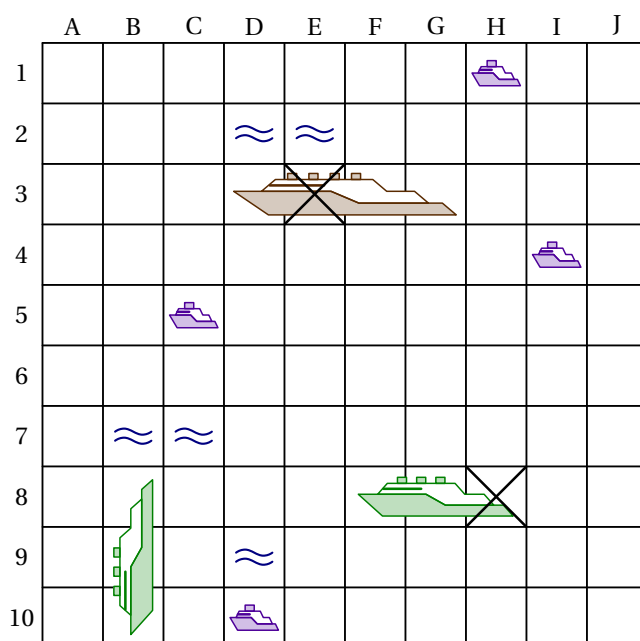
## 15.4 Bataille navale

### Exemple 89

```

tableau(10,10,0.8cm);
draw grille(1,1);
inverse_coord_y:=true;
style_coord_x="A";
place_coord:="hg";
draw coord;
tablabel(Bateau(4,marron),5.5,3);
tablabel(Bateau(3,0.5vert),7,8);
tablabel(Bateau(3,0.5vert) rotated 90,2,9);
tablabels(Bateau(1,violet),(4,10),(3,5),(8,1),(9,4));
tablabels(Eau(marine),(2,7),(3,7),(4,9),(4,2),(5,2));
tablabels(Croix,(5,3),(8,8));
fin;

```





## 15.5 Algoréa

### Exemple 90

```
tableau(13,5,1cm);  
coultableau[1] := (1,0.9,0.7);  
coultableau[2] := (1,0.9,0.7);  
draw damier(1,1);  
tablabeled(Mur(0.8rouge,noir),(1,2),(1,3),(1,4),(13,2),(13,3),(13,4));  
for i=1 upto 13:  
  tablabeled(Mur(0.8rouge,noir),(i,1),(i,5));  
endfor  
tablabel(Robotdroite(noir,0.7vert),2,2);  
for i=4,6,7,10,12:  
  tablabel(Caisse(orange,marron),i,3);  
  tablabel(Marque(2)((0.8,0.65,0.5)),i,4);  
endfor  
fin;
```

