

Mémo jps

Version 3

13 Novembre 2004

Avant tout, je tiens à remercier vivement Jean-Paul Vignault pour ces réponses nombreuses et rapides à toutes mes questions, sans qui je n'aurais jamais pu faire ce boulot. J'espère que ce guide vous servira, il y a sûrement des oublis, quelques erreurs (pas trop j'espère), n'hésitez pas à me contacter pour toute remarque : rouxn@caramail.com.

Roux Nicolas

Table des matières

1	Trucs de base	4
1.1	<u>Définition de la fenêtre du dessin</u>	4
1.2	<u>Quadrillages</u>	4
1.3	<u>Définitions des repères, axes, etc...</u>	4
2	Insertion de texte	5
2.1	<u>Définir la fonte</u>	5
2.2	<u>Définir le texte et l'emplacement</u>	5
2.3	<u>Orientation du texte</u>	5
2.4	<u>Inclure du texte en T_EX ou L^AT_EX</u>	5
3	Géométrie plane	6
3.1	<u>Éléments géométrique de base</u>	6
3.1.1	Points	6
3.1.2	Droites	6
3.1.3	Segments	6
3.1.4	Ligne brisée	6
3.1.5	Polygones	6
3.2	<u>Droites particulières</u>	6
3.2.1	Parallèle	6
3.2.2	Perpendiculaire, hauteur	6
3.2.3	Médiane	7
3.2.4	Médiatrice	7
3.2.5	Bissectrice	7
3.3	<u>Cercle</u>	7
3.3.1	Cercle	7
3.3.2	Arc de cercle	7
3.3.3	Portion de camembert	7
3.4	<u>Point particuliers</u>	7
3.4.1	Milieu d'un segment	7
3.4.2	Projeté orthogonal	8
3.4.3	Intersection de 2 droites	8
3.4.4	Intersection d'un cercle et d'une droite	8
3.4.5	Intersection de 2 cercles	8
3.4.6	Point sur le cercle	8
3.4.7	Image d'un point par rotation	8
3.5	<u>Vecteurs</u>	8
3.6	<u>Codage de figure</u>	9
3.6.1	Angles de même mesure	9
3.6.2	Côtés de même longueur	9
4	Tracés de courbes	10
4.1	<u>Définition des fonctions</u>	10
4.1.1	Notation polonaise inversée	10
4.1.2	Méthode rpn	10
4.2	<u>Tracé d'une courbe représentative d'une fonction</u>	10
4.3	<u>Tangente à une courbe</u>	10
4.4	<u>Placer des points sur une courbe et projeter les coordonnées sur les axes</u>	11
4.4.1	Placer un point sur une courbe	11
4.4.2	Projection des coordonnées des points sur les axes	11
4.5	<u>Hachurer</u>	11
4.5.1	Hachurer entre une courbe et l'axe des abscisses	11

4.5.2	Hachurer entre deux courbes	11
4.6	<u>Tracés de courbes passant par des points</u>	11
5	Statistiques	13
5.1	<u>Statistiques à une variable</u>	13
5.1.1	Diagramme en boîte	13
5.1.2	Diagramme en bâton	13
5.2	<u>Statistiques à deux variables</u>	14
5.2.1	Nuage de points	14
5.2.2	Ajustements affines	15
6	Divers	16
6.1	<u>Remplissage</u>	16
6.1.1	Définition du remplissage	16
6.1.2	Options de remplissages	16
6.1.3	Le remplissage	16
6.2	<u>Fins de lignes</u>	17

1 Trucs de base

1.1 Définition de la fenêtre du dessin

-10 10 setxrange : taille de la fenêtre en unités (ici -10; 10)

-15 15 setyrange : taille de la fenêtre en unités

30 setxunit : taille de l'unité en points

30 setyunit : taille de l'unité

.7 setlinewidth : définition de la taille des traits, attention quand c'est défini, c'est défini pour TOUT ce qu'il y a après (courbe, axes, etc...)

1.2 Quadrillages

Méthode 1 : faire des quadrillages complexes mais complets

/quadrillagewidth [.7 .4 .1] def : définition des largeurs de traits du quadrillage

[1 1 .5 .5 .1 .1] {.7 setgray} Quadrillage : définition des pas, puis de la couleur (avec l'option plus ou moins foncé)

Méthode 2 : faire des quadrillages de base :

1 1 Quadrillage : on place le pas, puis la commande Quadrillage

Méthode 3 : encore des quadrillages de base :

La première chose à faire : définir le pas : **1 1 setquadrillagestep**

Puis tout simplement taper **quadrillage**.

Si vous voulez tracer un quadrillage comme du papier millimétré il faut taper :

/quadrillagewidth [.7 .4 .1] def [1 1 .5 .5 .1 .1] orange Quadrillage

Au fait, il faut faire attention, il y a quadrillage et Quadrillage...

1.3 Définitions des repères, axes, etc...

traceaxes : comme l'indique son nom, ça trace les axes tout court, sans rien

axesarrow : ça, ça trace les flèches des axes

1 1 setsubtkstep : définit le pas (ici 1 et 1) pour les sous-divisions (par défaut les sous-divisions font 0,5 unités.)

ticks : trace les divisions sur les axes (sauf en 0)

xticks : la même chose mais seulement sur l'axe des abscisses

yticks : la même chose mais seulement sur l'axe des ordonnées

0 xtick : seulement pour la valeur 0 sur l'axe des abscisses

subticks : trace les sous-divisions (demi-unités pour être précis)

marks : ticks + valeurs marquées (sauf le 0)

xmarks : la même chose mais seulement sur l'axe des abscisses

ymarks : sans commentaires

0 xmark : seulement pour la valeur 0 sur l'axe des abscisses

unites : trace les vecteurs unités.

tracerepere : traceaxes, axesarrow, unites, ticks et subticks d'un coup!

2 Insertion de texte

2.1 Définir la fonte

`setTimes` ou `setTimesItalic` ou encore `setSymbol` pour avoir les belles lettres α π ... et aussi `15` `setfontsize...`

2.2 Définir le texte et l'emplacement

(A) `1 2 drtext` : place le texte A au point de coordonnées 1 2, plus précisément en bas (down) à droite (right) du point de coordonnées 1 2.

(B) `B text` : place le texte B au point B (si on l'a défini avant bien sûr)...

2.3 Orientation du texte

(banane) `A {30} drtext` : écrit le texte banane au point A en bas à droite et avec un angle de 30° (toujours dans le cercle trigo).

Mieux encore : (banane) `A {A C angle} urtext` : écrit "banane" en haut à droite du point A parallèlement à (AC).

2.4 Inclure du texte en T_EX ou L^AT_EX

`#tex# $ f(x)=\pi \times x $`

`A [2 2] cctextlabel` : place le texte ($f(x) = \pi \times x$) au point A agrandi 2 fois.

Attention : ce qui doit être écrit en tex doit être sur la même ligne que `#tex#`.

On peut aussi inclure des labels L^AT_EX, bien pratique pour les fractions, les ensembles de nombres...

`#latex# $\frac{1}{2}$`

`0 1 cctextlabel` : place le texte au point de coordonnées (0;1) sans agrandissement.

Attention : on utilise toujours la commande `texlabel`, pas de `latexlabel`.

2^e Attention : on précise toujours la position du texte par rapport au point (cc : center, center ; dr : down, right ; ul : up, left...).

3 Géométrie plane

3.1 Éléments géométrique de base

3.1.1 Points

La première chose à faire est de définir les points : `/A {1 2} def` définit le point A de coordonnées (1;2), de même `/B {2 3}` fait ce que tu penses.

1^{re}méthode

Puis `A point` place le point A.

Par contre si on veut placer plusieurs points il faut taper :

`[A B] points`

Il faut faire attention à la syntaxe : 1 point, on ne met pas de s à point, à partir de 2, on en met un s à points. Logique. 1 point, sans crochet, à partir de 2, on en met.

2^eméthode

Celle que j'utilise :

`[A B] {plus} plot` : pour les options, voir le guide de l'utilisateur jps2ps.

3.1.2 Droites

2 méthodes pour tracer une droite :

Méthode 1 : `A B droite` tout simplement, ça trace la droite passant par les points A et B.

Méthode 2 : On commence par définir la droite passant par les points A et B : `/d {A B} def` puis `d droite`, cette méthode permet de définir la droite pour s'en servir après.

3.1.3 Segments

`[A B] ligne` : trace le segment $[AB]$.

3.1.4 Ligne brisée

`[A B C] ligne` : trace la ligne brisée.

3.1.5 Polygones

Pour tracer un polygone, c'est tout simple, on donne les sommets du polygone (dans l'ordre bien sûr) :

`[A B C D] polygone` : trace le polygone ABCD.

`[A B C] polygone` : trace le triangle ABC.

3.2 Droites particulières

3.2.1 Parallèle

`/f {A B C paral} def` : (f) est la droite parallèle à (AB) passant par C.

3.2.2 Perpendiculaire, hauteur

`/d {B C A perp} def` : (d) est la droite perpendiculaire à (BC) passant par A.

3.2.3 Médiane

/d'' {A B C milieu} def : (d'') est la droite passant par A et le milieu de [BC]. Remarque : il est plus rapide de définir le milieu de [BC]...

3.2.4 Médiatrice

/d' {B C mediatrice} def : (d') est la médiatrice de [BC].

3.2.5 Bissectrice

/e {A B C bissectrice} def : (e) est la bissectrice de l'angle \widehat{ABC} .

3.3 Cercle

3.3.1 Cercle

Il y a 2 méthodes pour tracer un cercle :

Avec le centre et le rayon

I r cercle : trace le cercle de centre I et de rayon r.

Au fait ne pas oublier qu'on peut toujours tout définir, par exemple :

/r {A B distance} def

Cercle passant par trois points

A B C ABCercle : trace le cercle passant par les points A, B et C.

3.3.2 Arc de cercle

Méthode 1

A O B .5 (-) tripointarcarrow : trace l'arc \widehat{AOB} de rayon 0,5 avec un trait (-).

A M B .5 (=) tripointarcarrow : trace l'arc \widehat{AMB} de rayon 0,5 avec deux traits (=).

Une variante (avec flèche) :

A O B .5 (<-) tripointarcarrow...

Méthode 2

-45 0 B 2 Cercle : trace l'arc de cercle de l'angle -45 à 0, de centre B et de rayon 2.

Méthode 3

I A B ABCercle : trace l'arc de cercle de centre I passant par A, inscrit dans l'angle \widehat{AIB} .

3.3.3 Portion de camembert

0 90 A 1 wedge : portion de camembert entre les angles -90 et 0, de centre A et de rayon 1.

3.4 Point particuliers

3.4.1 Milieu d'un segment

/A' {B C milieu} def : A' est le milieu de [BC].

3.4.2 Projeté orthogonal

`/A' {I B C orthoproj}` def : on définit A comme étant le projeté orthogonal de I sur le segment $[B;C]$.

3.4.3 Intersection de 2 droites

`/C {d d' interdroite}` def : C est le point d'intersection des droites (d) et (d') (préalablement définies).

3.4.4 Intersection d'un cercle et d'une droite

`d C interdroitecercle A B` : Les points A et B sont les points d'intersection de la droite (d) et du cercle C, rangé par ordre décroissant d'ordonnée si possible, par ordre décroissant d'abscisse sinon.

3.4.5 Intersection de 2 cercles

`C C' interdroitecercle A B` : Les points A et B sont les points d'intersection du cercle C et du cercle C', rangé par ordre décroissant d'ordonnée si possible, par ordre décroissant d'abscisse sinon.

3.4.6 Point sur le cercle

Dans cet exemple, on commence par définir le cercle de centre O et de rayon 2 :

`/cerc {O 2} cercle`

Puis A le point du cercle d'angle -45

`/A {-45 cerc cpoint} def`

3.4.7 Image d'un point par rotation

`/B {A O 20 rotatpoint}` def : B est l'image du point A par la rotation de centre O et d'angle 20.

3.5 Vecteurs

On définit un vecteur exactement comme un point : `/u {1 2} def` par exemple.

Pour faire des calculs avec des vecteurs, c'est presque comme pour les fonctions, exemple :

`/AM { u 2 mulv v 3 mulv subv}` def qui correspond à $\overrightarrow{AM} = 2 \times \overrightarrow{u} - 3 \times \overrightarrow{v}$.

On peut aussi définir un point vectoriellement, exemple :

`/N {A u v 2 mulv addv addv}` def qui correspond à N défini par $\overrightarrow{AN} = \overrightarrow{u} + 2 \times \overrightarrow{v}$.

Donc maintenant qu'on sait faire des calculs avec des vecteurs, on va les tracer de 2 manières différentes :

Méthode 1 :

`A u vect` : on donne le point de départ, le vecteur et zou.

Méthode 2 :

Si on ne veut pas s'embêter à calculer les coordonnées du (ou des) vecteur(s), il y a encore plus simple :

`[A B] (->) ligne` : trace le vecteur \overrightarrow{AB} .

`/arrowscale {1.5 1.5} def` : taille des flèches.

3.6 Codage de figure

3.6.1 Angles de même mesure

A B C **angledroit** : trace l'angle droit \widehat{ABC}

A O B .5 (-) **tripointarcarrow** : trace l'arc \widehat{AOB} de rayon 0,5 avec un trait (-).

A M B .5 (=) **tripointarcarrow** : trace l'arc \widehat{AMB} de rayon 0,5 avec deux traits (=).

3.6.2 Côtés de même longueur

A B **1 marked** : place une marque sur le segment $[AB]$.

C D **2 marked** : place deux marques sur le segment $[CD]$.

[A B B C C D] **{1 marked} dapply** : $AB=BC=CD$

4 Tracés de courbes

4.1 Définition des fonctions

2 méthodes pour le faire :

4.1.1 Notation polonaise inversée

```
/f {setxvar  
1 2 x 2 exp mul x 2 exp 100 sub div sub  
} def
```

qui correspond à la fonction définie par $f(x) = 1 - \frac{2x^2}{x^2 - 100}$.

4.1.2 Méthode rpn

```
/f {setxvar  
#rpn# 1 - (2x^2)/(x^2 - 100)  
} def
```

Attention : comme pour l'insertion des labels tex, il ne doit y avoir sur la ligne avec #rpn# que l'expression mathématique de la fonction, surtout pas le } def.

4.2 Tracé d'une courbe représentative d'une fonction

{f} **courbe** : pour tracer la courbe tout simplement
1 2 {f} **Courbe** : pour tracer la courbe sur l'intervalle [1; 2]

Petite info, si la fonction n'est pas définie en un point (type $\frac{1}{x}$) l'asymptote verticale est tracée. Par contre les asymptotes horizontales, faut le faire soit même.
Je le redis, il y a aussi Courbe et courbe...

4.3 Tangente à une courbe

Pour tracer une tangente à une courbe, il y a 2 méthodes : celle du perfectionniste (comme moi) et l'autre (pas comme moi).

Méthode 1 : perfectionniste

Il faut d'abord définir la dérivée de la fonction de départ, avec l'exemple précédent, il faut faire :

```
/f' {setxvar  
#rpn# (400x)/((x^2 - 100)^2)  
} def
```

puis définir la taille de la tangente comme ceci :

2 settailletangente

En effet, quand on trace une tangente avec cette méthode, on trace en fait un vecteur tangent, et non une tangente, donc on peut demander un vecteur tangent (en gros la taille réglée à 2 (unités comme toujours)) ou une tangente, on veillera à prendre une grande valeur pour ne pas voir les flèches dans l'écran, si t'as rien compris, fais un essai avec la taille à 2, puis 5 puis 32, là ça sera plus clair.

Il suffit après de taper : **2 (f) tangente** et on a une tangente au point d'abscisse 2.

Méthode 2 : pas la mienne

Il suffit tout simplement de définir l'expression de la tangente dans une fonction (g par exemple) et de la tracer.

L'avantage de la méthode 1 : on peut tracer un paquet de tangentes rapidement, sans calculer leurs expressions exemple : **2 (f) tangente 3 (f) tangente 6 (f) tangente...** et on a les tangentes aux points d'abscisses 2, 3 et 6. Alors qu'avec l'autre méthode, il aurait fallu déterminer les équations des 3 tangentes puis les définir, puis les tracer....

4.4 Placer des points sur une courbe et projeter les coordonnées sur les axes

4.4.1 Placer un point sur une courbe

Soit f définie comme suit :

```
/f setxvar{  
#rpn# e ^ x  
}def  
\ A {1 1 f} def : définit le point A de coordonnées (1 ;f(1))
```

4.4.2 Projection des coordonnées des points sur les axes

Pour faire ce qu'on veut, il faut utiliser la commande **dashpoint** qui fonctionne comme **point**.

Exemple : **A dashpoint** ou **[A B] dashpoints**.

Un petit truc aussi, si tu ne veux pas utiliser la commande marks pour avoir la graduation mais que tu veux les valeurs des projections, il suffit de faire ça par exemple : **[A B] {ymark xmark} papply**

4.5 Hachurer

4.5.1 Hachurer entre une courbe et l'axe des abscisses

{f} hachcourbe : pour hachurer partout.

2 3 {f} Hachcourbe : pour hachurer sur l'intervalle [2; 3].

4.5.2 Hachurer entre deux courbes

{f}{g} hachcourbes : pour hachurer partout.

2 3 {f}{g} Hachcourbes : pour hachurer sur l'intervalle [2; 3] entre les courbes représentatives des fonctions f et g .

Attention : il y a hachcourbe, Hachcourbe, hachcourbes et Hachcourbes...

Remarque : On peut définir les couleurs, remplissages comme pour les remplissages... voir "remplissage" du chapitre 6.

4.6 Tracés de courbes passant par des points

Un exemple vaut 1013495934 explications donc :

On définit les points suivants :

```
/A {-4 -1} def  
/B {-2 -3} def  
/C {0 -1} def  
/D {.4 0} def
```

```
/E {1 2} def
/F {2 4} def
/G {4 2} def
/H {5 0} def
/I {6 -2} def
/J {7 -3} def
/K {9 -5} def
```

On veut tracer la courbe passant par ces points et ayant un minimum au point B et un maximum au point F. La courbe part du point A vers le point B, il faut définir l'angle de départ (en degré, par rapport au cercle trigo). On tape donc :

```
[A {300 dir} .. B {right} .. C .. D .. E .. F{right} .. G .. H .. I .. J .. K] draw
```

Tu remarques que pour les points B et F on a ajouté `{right}` (comme par hasard, ce sont les extrêmes) à côté, essaie de faire avec `{left}` tu comprendras.

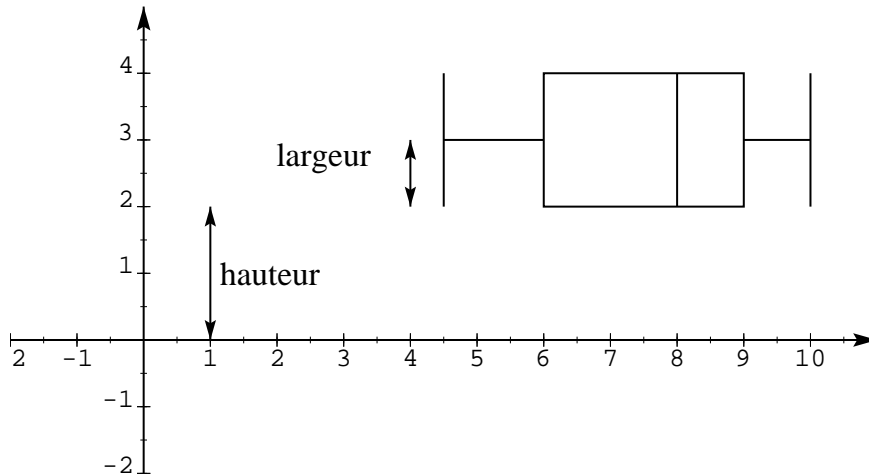
5 Statistiques

5.1 Statistiques à une variable

5.1.1 Diagramme en boîte

Pour tracer un diagramme en boîte, il faut :

1. déterminer la hauteur à laquelle on place la boîte (hauteur=ordonnée dans le repère) , en effet, on peut tracer plusieurs boîtes les unes au-dessus des autres, mais il ne faut pas qu'elles se chevauchent ;
2. déterminer la largeur de la boîte en unité, attention, la largeur est définie comme sur le schéma suivant :



3. on peut donc finir le boulot :

On commence par la hauteur, puis la largeur, la valeur minimum de la série, le premier quartile, la médiane, le troisième quartile et enfin la valeur maximum.

Dans l'exemple précédent, on a :

2 1 4.5 6 8 9 10 diag_en_boite

5.1.2 Diagramme en bâton

Pour tracer un diagramme en bâton, il faut commencer par définir la largeur des bâtons, par défaut, cette largeur vaut 1.

`/rectwidth 0.5 def` : pas grand chose à dire...

Il faut ensuite définir la liste contenant toutes les valeurs. Petit inconvénient, si dans la série statistique, une valeur apparaît 3 fois, il faut la mettre 3 fois dans la série.

`/liste [16 11.5 8.5 8 16 7 10.5 11 3 4.5 10 6 7 18.5 11 11 4.5 20 12 1] def` : tu peux remarquer que l'ordre importe peu, par contre la valeur 11 apparaît 3 fois...

Enfin pour tracer le diagramme, il suffit de faire un "traitement" sur la liste :

`liste traitement {rect1} plot` : on remarque 1 option : `rect1`, pour les autres options, voir le manuel de `jps`.

5.2 Statistiques à deux variables

5.2.1 Nuage de points

Il y a deux méthodes pour tracer un nuage de points.

Première méthode : point par point.

Deux façons de le faire :

1. Il suffit de définir tous les points, puis de les tracer.

Exemple :

```
/A{1 1} def
```

```
/B{2 4} def
```

...

```
[A B ... ] points
```

2. On peut aussi tout simplement tracer les points sans les définir :

Exemple :

```
[1 1 2 3 4 5 6 7] points : trace les points de coordonnées (1;1), (2;3), (4;5) et (6;7).
```

Deuxième méthode : avec des listes.

Cette méthode est beaucoup plus pratique, il suffit de rentrer les valeurs des deux séries dans deux listes, puis de les fusionner et enfin de les tracer.

Exemple :

```
[0 5 10 15 20 25 27] : liste correspondants aux valeurs de la première série.
```

```
[44.6 37.9 31.2 25.3 20.2 16.9 15.6] : liste correspondants aux valeurs de la deuxième série.
```

fuz : fusionne les deux listes.

points : trace le nuage de points.

Il existe plusieurs type de points pour les nuages, en général, on utilise les croix (+) pour tracer un point. Il y a un grand choix de points :

plus1, plus2, plus3

times1, times2, times3

... (voir le manuel de jps)

Si on veut utiliser un autre style de points que celui donné, il faut utiliser la syntaxe suivante à la place de **points**, par exemple :

```
{plus2} plot
```

Exemple :

```
-0.5 6 setxrange
```

```
-0.5 5 setyrange
```

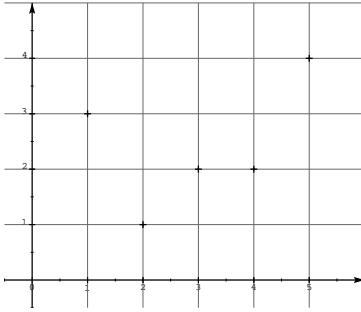
```
tracerepere
```

```
/11 [1 2 3 4 5] def
```

```
/12 [3 1 2 2 4] def
```

```
11 12 fuz {plus2} plot
```

donne le résultat suivant :



5.2.2 Ajustements affines

1. Régression de y en x

Avec l'exemple précédent :

```
/11 [1 2 3 4 5] def
```

```
/12 [3 1 2 2 4] def
```

```
-1 6 setxrange
```

```
-1 5 setyrange
```

```
tracerepere
```

nuage de points :

```
l1 l2 fuz plus2 plot
```

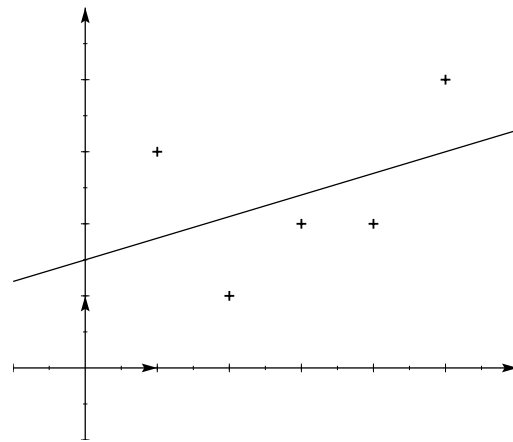
définition de la droite :

```
l1 l2 regyx /dyx defdroite
```

représentation de la droite :

```
dyx droite
```

donne le résultat suivant :



2. Régression de x en y

Toujours avec l'exemple précédent :

```
/11 [1 2 3 4 5] def
```

```
/12 [3 1 2 2 4] def
```

```
-1 6 setxrange
```

```
-1 5 setyrange
```

```
tracerepere
```

nuage de points :

```
l1 l2 fuz plus2 plot
```

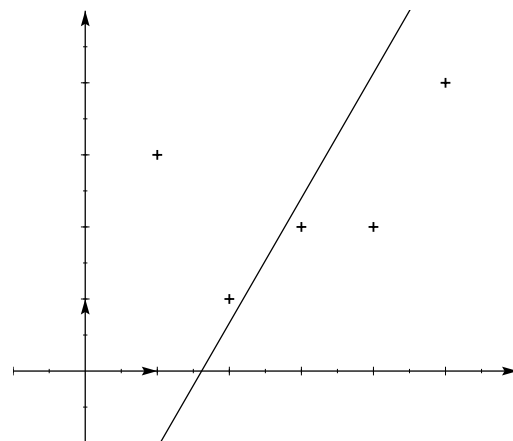
définition de la droite :

```
l1 l2 regxy /dxy defdroite
```

représentation de la droite :

```
dxy droite
```

donne le résultat suivant :



3. Droite de Mayer

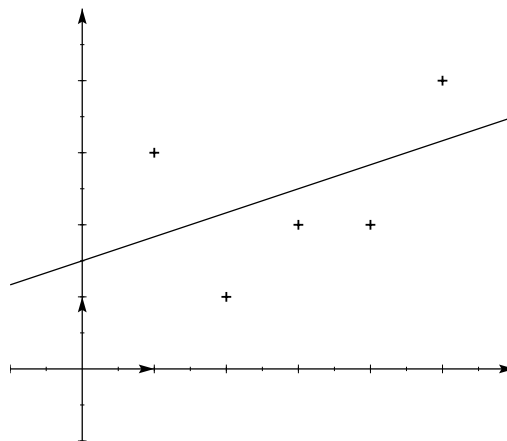
Encore avec l'exemple précédent :

```
/11 [1 2 3 4 5] def
```

```
/12 [3 1 2 2 4] def
```

```
-1 6 setxrange
```

-1 5 setyrange
tracerepere
 nuage de points :
l1 l2 fuz plus2 plot
 définition de la droite :
l1 l2 Mayer /dm defdroite
 représentation de la droite :
dm droite



donne le résultat suivant :

6 Divers

6.1 Remplissage

La plupart des objets disposent d'une version étoilée qui permet de remplir un domaine plan défini par l'objet.

6.1.1 Définition du remplissage

La première chose à faire est de définir le remplissage :

/fillstyle{hachure} def : pour hachurer
/fillstyle{fill} def : pour du remplissage plein

6.1.2 Options de remplissages

On peut ajouter des couleurs :

/fillstyle{bleu hachure} def
/fillstyle{jaune fill bleu hachure} def : hachures bleues sur fond jaune

On peut déterminer différents paramètres :

/fillstyle{/hangle 90 store hachure} def : /hangle 90 store définit l'angle de hachurage à 90°, initialement à -45°.

/fillstyle{/hstep 3 store hachure} def : définit l'espace entre deux hachures (en points), initialement à 8.

/fillstyle{/hwidth 0.1 store hachure} def : épaisseur des hachures, initialement à 0,8.

On peut cumuler les options, sans oublier store après chaque option, exemple :

/fillstyle{/hangle 90 store /hstep 14 store bleu hachure} def : hachures bleues, 14 points d'écart, angle :90°.

6.1.3 Le remplissage

[A B C] polygone* : remplit le polygone ABC avec le remplissage définit.

A 2 cercle* : remplit le disque de centre A et de rayon 2 avec le remplissage définit.

6.2 Fins de lignes

valeur	effet
-	rien
_	disques centrés aux points terminaux
o-o	cercles centrés aux points terminaux
<->	flèches
>-<	flèches inversées
\(-\)	demi-cercles rentrant
\) - \((demi-cercles sortant
[-]	crochets fermés
] - [crochets ouverts

On peut mixer les options, par exemple, o->.

Ces options sont valides avec : courbeparam, Courbeparam(dont je ne parle pas) courbe, Courbe, ligne, line (dont je ne parle pas), Cercle et Ellipse (dont je ne parle pas).

Exemples :

[A B](->) ligne

1 2 {f}(o->) Courbe

-45 45 A 2 (o-) Cercle