METAPOST

Table des matières

1 Repère **2**

1.1 Axes . . . 2

1.2 Graduations . . . 2

1.3 Quadrillages . . . 3

2 Points, droites **4**

2.1 Points . . . 4

2.2 Projection de points . . . 4

2.3 Segment . . . 4

2.4 Droite . . . 5

3 Fonctions **6**

3.1 Courbes $y = f(x)$ . . . 6

3.2 Tangente . . . 6

4 Suites **8**

4.1 Suites $u_n = f(n)$ . . . 8

4.2 Suites $u_{n+1} = f(u_n)$ . . . 8

5 Intégration **9**

5.1 Aire sous une courbe . . . 9

5.2 Méthodes des rectangles . . . 10

5.3 Méthodes des trapèzes . . . 12

6 Géométrie **13**

6.1 Triangle . . . 13

6.2 Cube . . . 14

6.3 Diverses figures dans l'espace . . . 15

7 Arbre en proba **17**

8 Boîtes à moustaches **17**

9 Conversions **17**

9.1 Conversion en eps . . . 18

9.2 Conversion en pdf . . . 18

Tout d'abord, on commencera par copier le répertoire `newcourbes` dans le texmf personnel, par exemple dans /home/-moi/texmf/metapost avec dedans `newcourbes.mp, couleur.mp, geo.mp`

Chaque fichier mp commencera donc par le préambule

```
input newcourbes;
input geo;
input couleur;
```

La sortie pdf est de moins bonne qualité que la sortie ps, donc aucune hésitation pour le choix de l'extension : voyez, si vous êtes équipé(e), la différence en cliquant ICI

Les mots clés colorés en bleus sont ceux présents dans la configuration de MetaPOST par défaut. Les mots-clés rajoutés restent en noir.

1 Repère

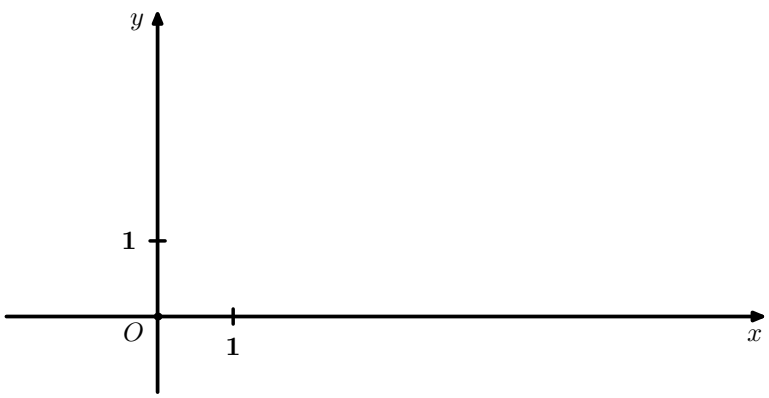
La première chose est de fixer le repère à l'aide de la macro `repere` :

```
repere(origine des x, origine des y,Xmin,Xmax,Ymin,Ymax,Unite x,Unite y)
```

1.1 Axes

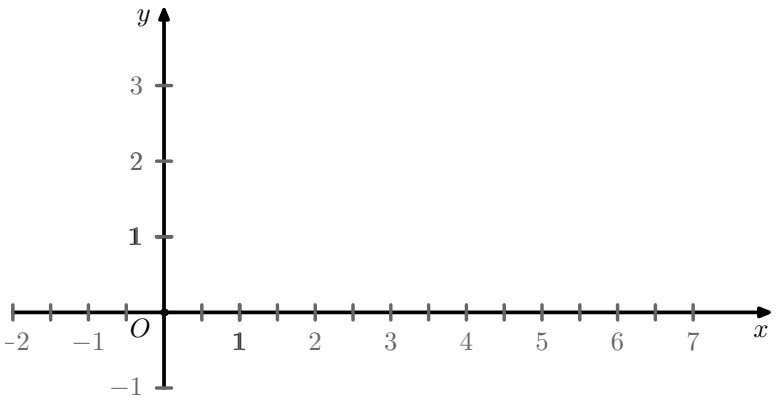
On tace les axes, l'origine, les unités, les labels des axes.

```
beginfig(1);
repere(0,0,-2,8,-1,4,1cm,1cm);
r_axes;
r_origine;
r_unites;
r_labelxy;
endfig;
```



1.2 Graduations

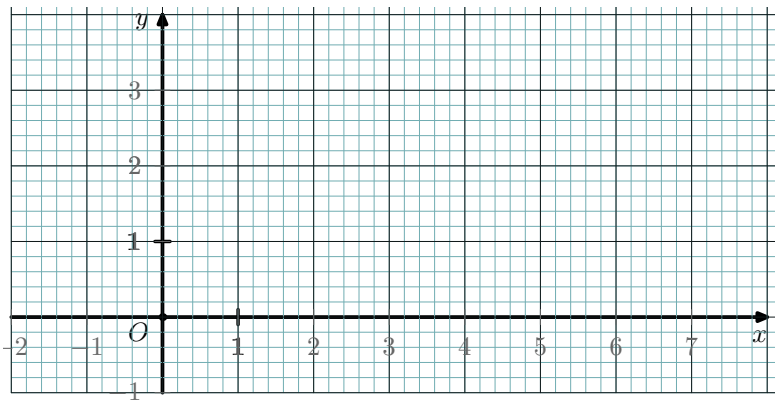
```
...
grad_x(0.5,1,0.4white);
grad_y(1,1,0.4white);
...
```



1.3 Quadrillages

On utilise `quad_xy(fraction de l'unité, couleur)` et `quadu_xy(couleur)` ou seulement `quad_x` et `quad_y` si on ne veut qu'une partie du quadrillage.

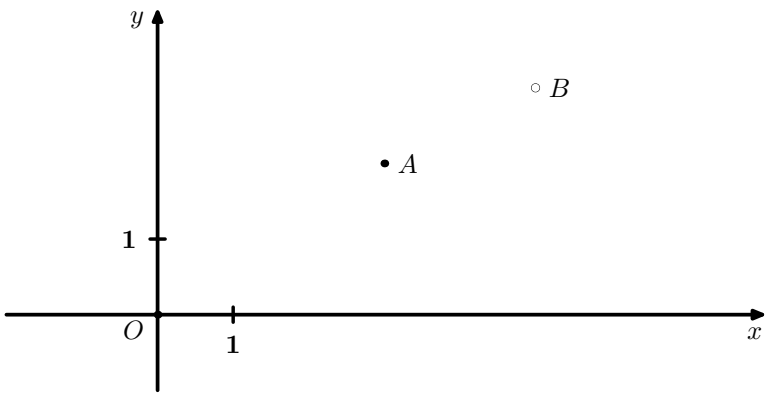
```
...
quad_xy(0.2,0.3*or);
quadu_xy(0.1*or);
...
```



2 Points, droites

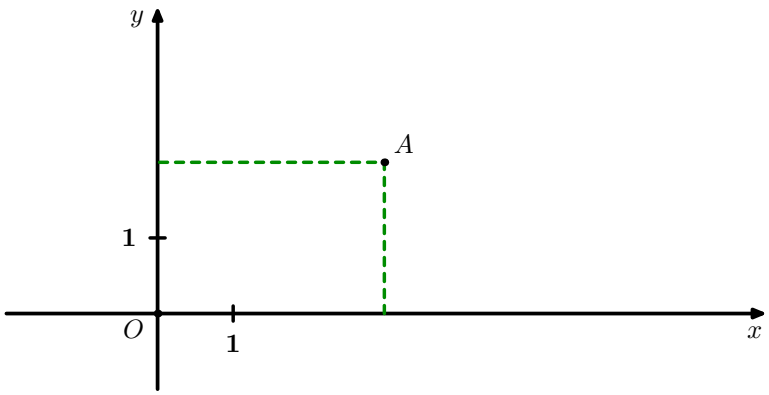
2.1 Points

```
...
r_pp(3,2);
r_cp(5,3);

label.rt(btex $A$ etex, r_p(3,2));
label.rt(btex $B$ etex, r_p(5,3));
...
```



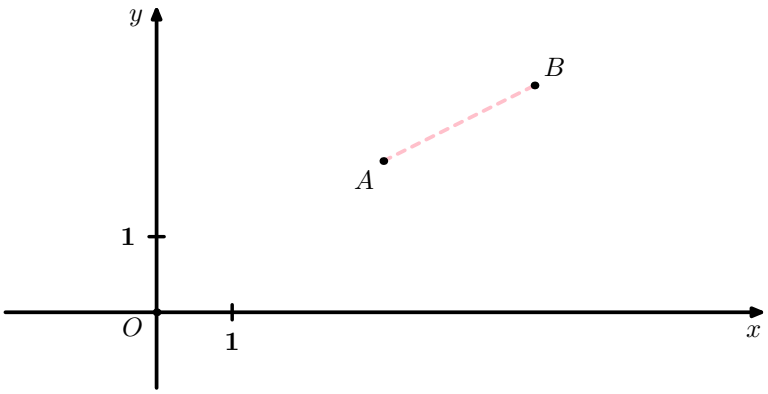
2.2 Projection de points

```
...
r_proj(3,2,vert_fonce);
...
```



2.3 Segment

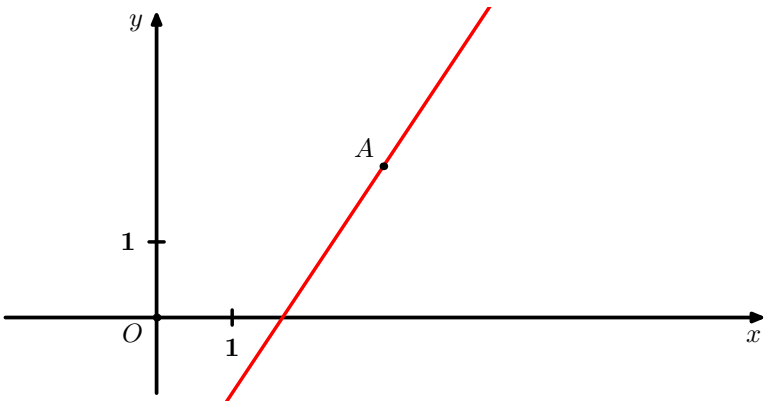
```
r_segment(3,2,5,3);
```



2.4 Droite

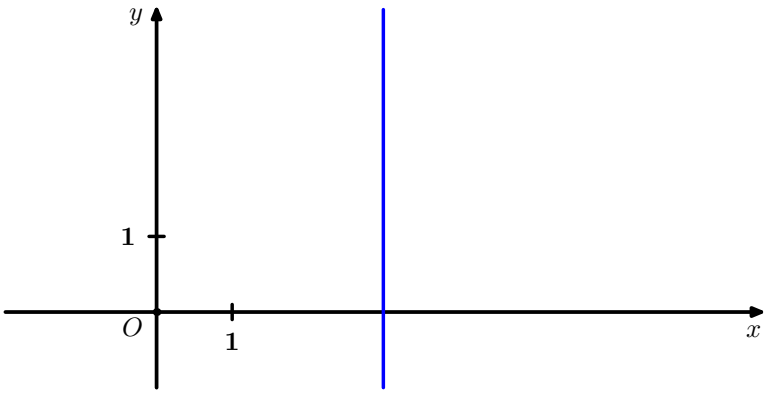
Droite passant par le point A(3 ; 2) et de coefficient directeur 1.5

```
...
draw r_droite(3,2,1.5) withcolor red;
...
```



Droite d'équation $x = 3$

```
...
draw rx_droite(3) withcolor blue;
...
```

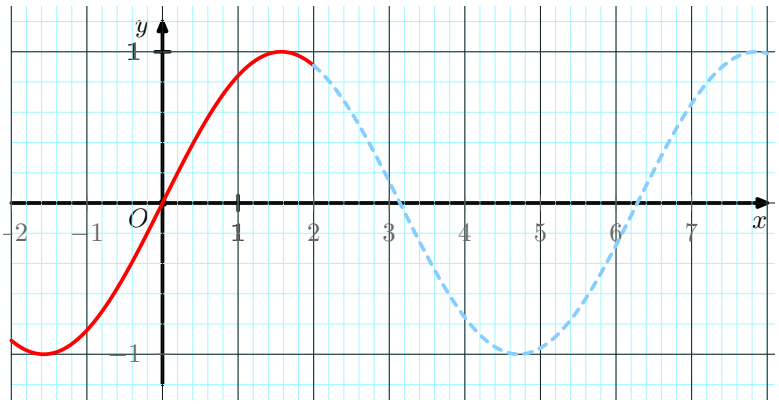


3 Fonctions

3.1 Courbes $y = f(x)$

On travaille en paramétrant les abscisses et les ordonnées pour plus de généralité, donc on définit `fx(t)` et `fy(t)` et on utilise `f_courbe(fx,fy,ti,tf,nb de points)`

```
beginfig(9)
%%%%%%%%%%%%%%%%%%%%%%%%
vardef fx(expr t)=
t
enddef;

vardef fy(expr t)=
                              sin(t) % c'est la seule ligne à changer
enddef;
%%%%%%%%%%%%%%%%%%%%%%%%
repere(0,0,-2,8,-1,1,1cm,2cm);
r_axes;
r_origine;
r_unites;
r_labelxy;
quad_xy(0.2,0.3*or);
quadu_xy(0.1*or);
grad_x(1,1,0.4white);
grad_y(1,1,0.4white);

draw f_courbe(fx,fy,-2,2,100)withpen pencircle scaled 1.5bp withcolor red;
draw f_courbe(fx,fy,2,8,100)withpen pencircle scaled 1.5bp withcolor bleu dashed evenly;

r_fin;
endfig;
```



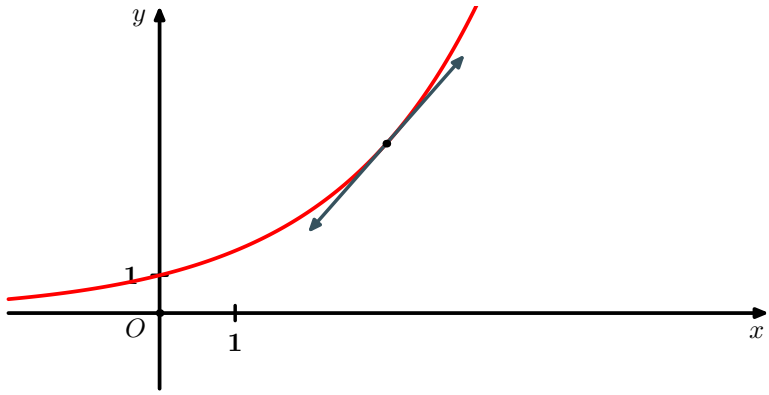
3.2 Tangente

On obtient une approximation du nombre dérivée en a à partir du taux de variation de pas h.

Si on veut une double flèche de « largeur » b au point d'abscisse a, on utilise `tracef_tangente(fx,fy,a,b,h,couleur)`.

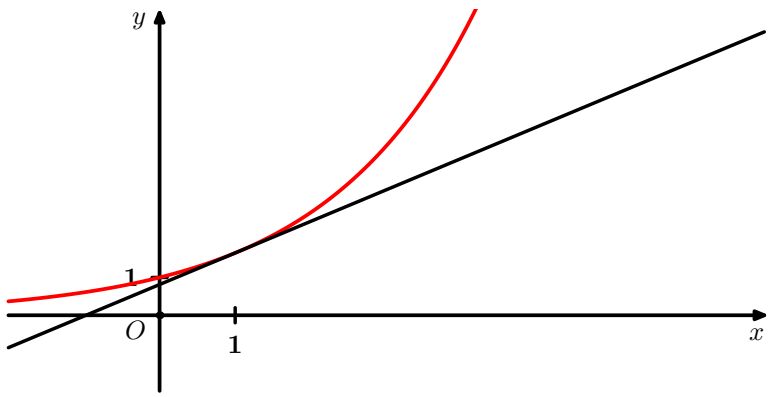
```
%%%%%%%%%%%%%%%%%%%%%%%%
vardef fx(expr t)=
t
enddef;
```

```
vardef fy(expr t)=
                              exp(0.5*t) % c'est la seule ligne à changer
enddef;
%%%%%%%%%%%%%%%%%%%%%%
repere(0,0,-2,8,-2,8,1cm,0.5cm);
r_axes;
r_origine;
r_unites;
r_labelxy;


draw f_courbe(fx,fy,-2,8,100)withpen pencircle scaled 1.5bp withcolor red;

tracef_tangente(fx,fy,3,1,0.05,bleu_f);

r_fin;
endfig;
```



Si on veut tracer la tangente, on préfèrera `f_tangente(fx,fy,a,h)`

```
...
draw f_tangente(fx,fy,1,0.05);
...
```

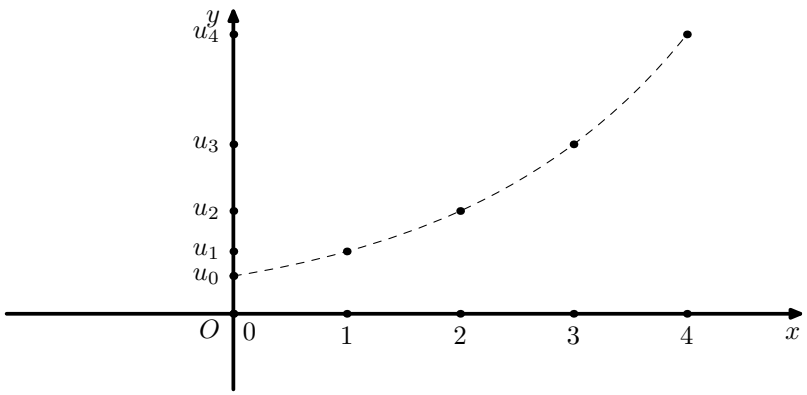


4 Suites

4.1 Suites $u_n = f(n)$

Il faut rentrer ux et uy et `u_courbe(ux,uy,ni,nf,t)` ; avec t prenant la valeur 1 si on veut les u_i et 0 sinon.

```
beginfig(12)
%%%%%%%%%%%%%%%%%%%%%%%%
vardef ux(expr t)=
t
enddef;
vardef uy(expr t)=
                                exp(0.5*t) % c'est la seule ligne à changer
enddef;
%%%%%%%%%%%%%%%%%%%%%%%%
repere(0,0,-2,5,-2,8,1.5cm,0.5cm);
r_axes;
r_origine;

 u_courbe(ux,uy,0,4,1);

r_fin;
endfig;
```

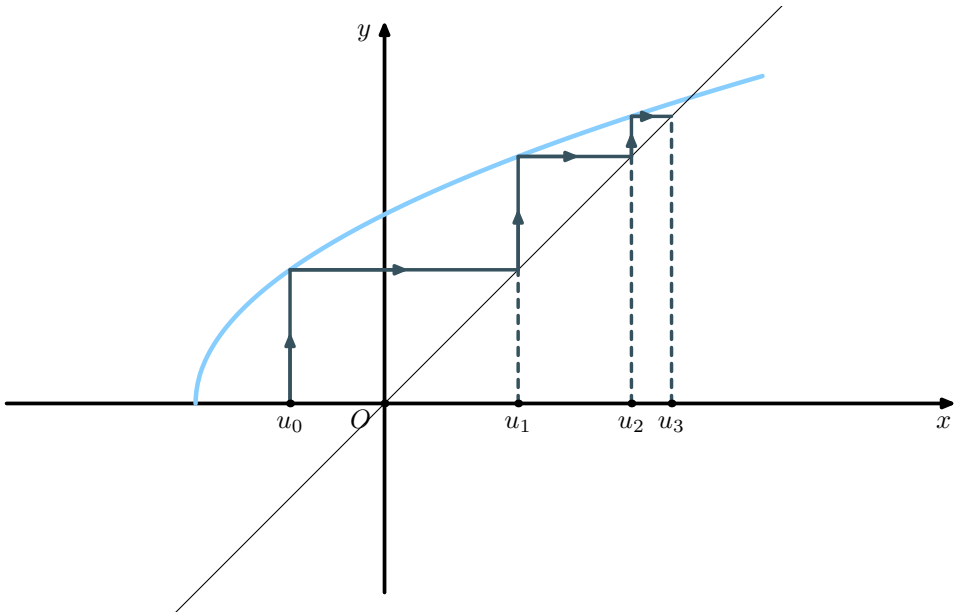


4.2 Suites $u_{n+1} = f(u_n)$

On utilise `u_reccourbe(fx,fy,u0,ni,nf,xi,xf,t`, t jouant le même rôle que plus haut.

```
beginfig(13)
%%%%%%%%%%%%%%%%%%%%%%%%
vardef fx(expr t)=
t
enddef;
vardef fy(expr t)=
                                sqrt(1+t) %
enddef;
%%%%%%%%%%%%%%%%%%%%%%%%
repere(0,0,-2,3,-1,2,2.5cm,2.5cm);
r_axes;
r_origine;

 u_reccourbe(fx,fy,-0.5,0,3,-1,2,1);
```

```
r_fin;
endfig;
```

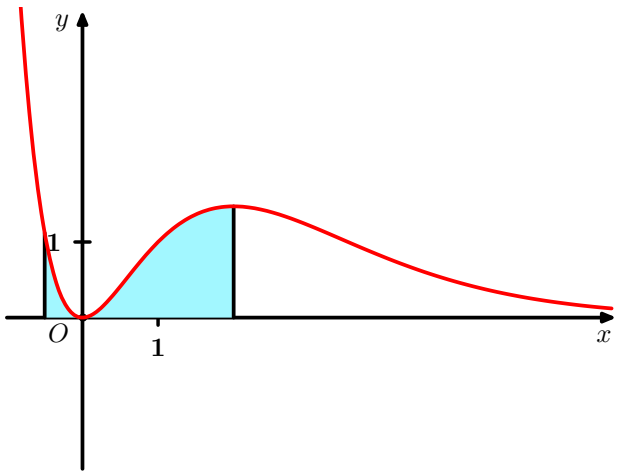


5 Intégration

5.1 Aire sous une courbe

On utilise `Aire(fx,fy,a,b,couleur)` pour représenter le domaine compris entre la courbe d'équation $y = f(x)$, l'axe des abscisses, les droites d'équation $x = a$ et $x = b$.

```
beginfig(32)

%%%%%%%%%%%%%%%%%%%%%%%%
vardef fx(expr t)=
t
enddef;

vardef fy(expr t)=
                              t*t*exp(1-t)
enddef;
%%%%%%%%%%%%%%%%%%%%%%%%
repere(0,0,-1,7,-2,4,1cm,1cm);

Aire(fx,fy,-.5,2,bleu_ciel);

r_axes;
r_origine;
r_unites;
r_labelxy

draw f_courbe(fx,fy,-1,7,50)withpen pencircle scaled 1.5bp withcolor red;

r_fin
endfig;
```

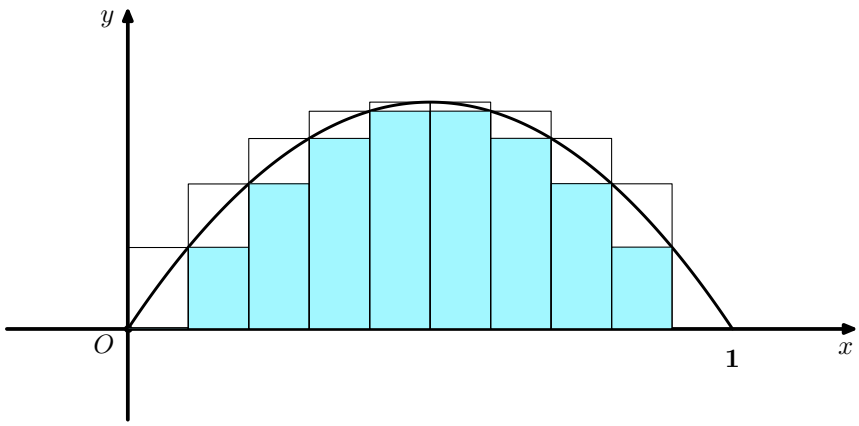


5.2 Méthodes des rectangles

On utilise `trace_rectangles_min_c(fx,fy,a,b,largeur en cm ,couleur)` et `trace_rectangles_max(fx,fy,a,b,largeur en cm)`

Il existe une version « transparente » de la première macro : `trace_rectangles_min_t(fx,fy,a,b,largeur en cm )`

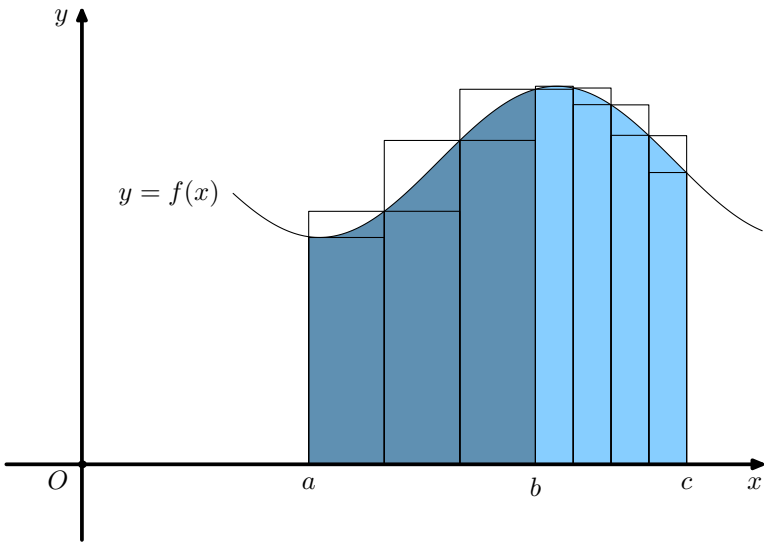
```
%=============================================================================
vardef fx(expr t) =
t
enddef;
vardef fy(expr t) =
                        (-2)*((t-0.5)**2)+0.5
enddef;
%=============================================================================

beginfig(14);
repere(0,0,-0.2,1.2,-0.2,0.7,8cm,6cm);
r_axes;
r_origine;
r_labelxy;

draw f_courbe(fx,fy,0,1,100)  withpen pencircle scaled 1.2bp;

 trace_rectangles_min_c(fx,fy,0,1,1/10,bleu_ciel);
trace_rectangles_max(fx,fy,0,1,1/10);

r_fin;
endfig;
```



Après, on peut s'amuser :

```
%=============================================================
vardef fx(expr t) =
t
enddef;
vardef fy(expr t) =
               cos(t)+4
enddef;
%=============================================================
beginfig(15);
path p[],q[],t[];
repere(0,0,-1,9,-1,6,1cm,1cm);

%definition de la surface
p1 = f_courbe(fx,fy,2,9,100);
q1 = rx_droite(3);
q2= rx_droite(6);
q3 =rx_droite(8);
q4 = r_droitedir(0,0,0);

%remplissage de la surface
t1 = buildcycle(q2,p1,q3,q4);
fill t1 withcolor bleu;
t2= buildcycle(q1,p1,q2,q4);
fill t2 withcolor bleu_m;

%trace de la surface
draw p1;
draw r_point(3,0)--f_point(fx,fy,3);
draw r_point(6,0)--f_point(fx,fy,6);
draw r_point(8,0)--f_point(fx,fy,8);
label.lft(btex $y=f(x)$ etex, f_point(fx,fy,2));
label.bot(btex $a$ etex,  r_point(3,0));
label.bot(btex $b$ etex,  r_point(6,0));
label.bot(btex $c$ etex,  r_point(8,0));


trace_rectangles_min_t(fx,fy,3,6,1);

trace_rectangles_max(fx,fy,3,6,1);

trace_rectangles_min_t(fx,fy,6,8,1/2) ;
```

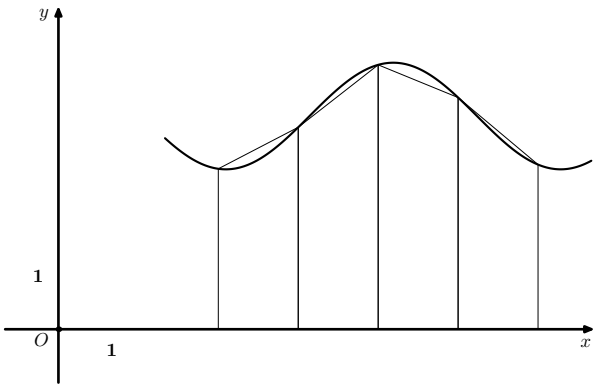
```
trace_rectangles_max(fx,fy,6,8,1/2);

r_axes;
r_origine;
r_fin;
endfig;
```



5.3 Méthodes des trapèzes

trace_trapezes(fx,fy,a,b,largeur)

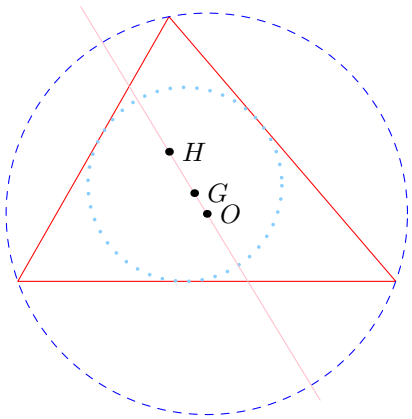
```
%=============================================================
vardef fx(expr t) =
t
enddef;
vardef fy(expr t) =
                    cos(t)+4
enddef;
%=============================================================
beginfig(16);
repere(0,0,-1,10,-1,6,1cm,1cm);
r_axes;
r_origine;
r_labelxy;

draw  f_courbe(fx,fy,2,10,100) withpen pencircle scaled 1.2bp;

 trace_trapezes(fx,fy,3,9,1.5);

r_fin;
endfig;
```



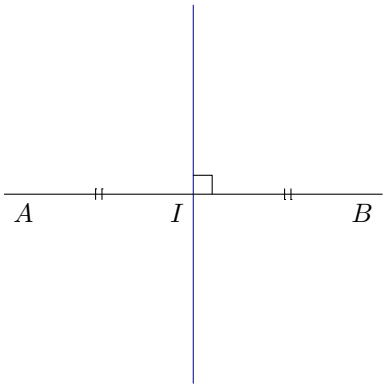
6 Géométrie

6.1 Triangle

```
beginfig(18);
         pair A,B,C,O,G,H;
        string s;
        u=1cm;
        A=origin;B=(5u,0);C=(2u,3.5u);
        draw A--B--C--cycle withcolor red;
        O=centrecerclecirconscrit(A,B,C);
        G=centredegravite(A,B,C);
        H=orthocentre(A,B,C);

            draw cerclecirconscrit(A,B,C) dashed evenly withcolor blue;
            draw cercleinscrit(A,B,C) dashed withdots withpen pencircle scaled
            1.5bp withcolor bleu;

          pickup pencircle scaled 2 bp;
        for t=O,G,H :
                draw t withpen pencircle scaled 3bp;
        endfor

        draw droite(O,G,10) withcolor rose;

        label.rt(btex $O$ etex, O);
        label.rt(btex $G$ etex, G);  label.rt(btex $H$ etex, H);
endfig;
```



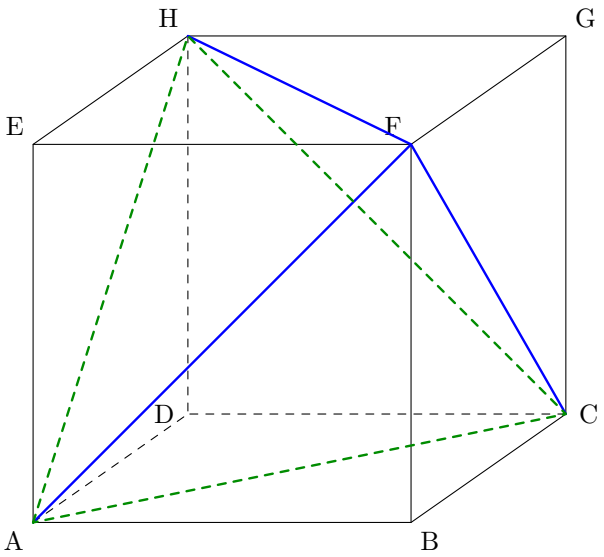
```
beginfig(19);
        pair A,B,M,I;
        A=origin;B=(5u,0);

draw mediatrice(A,B,0.5) withcolor blue;
I:=milieu(A,B);

M:=B rotatedaround(I,90);

draw symbole_ortho(B,I,M,0.25u);

draw A--B;
label.lrt(btex $A$ etex, A);
label.llft(btex $B$ etex, B);

label.llft(btex $I$ etex, I);

draw_marks(A--I,2);         draw_marks(B--I,2);

endfig;
```



6.2 Cube

La macro `cube(origine,largeur arête)` trace un cube à partir du point origine et `nommecube` nomme les sommets de manière usuelle.

```
beginfig(17);
        picture lecube;
        pair A,H,F,C;
        lecube=cube((0,0),5cm);
        draw lecube;
        nommecube;
        A=sommetCube0;
        H=sommetCube7;
        F=sommetCube5;
        C=sommetCube2;
        pickup pencircle scaled 1bp;
        draw A--F--C withcolor blue;
        draw H--F withcolor blue;
        draw A--H--C--cycle dashed evenly withcolor vert_fonce;
endfig;
```

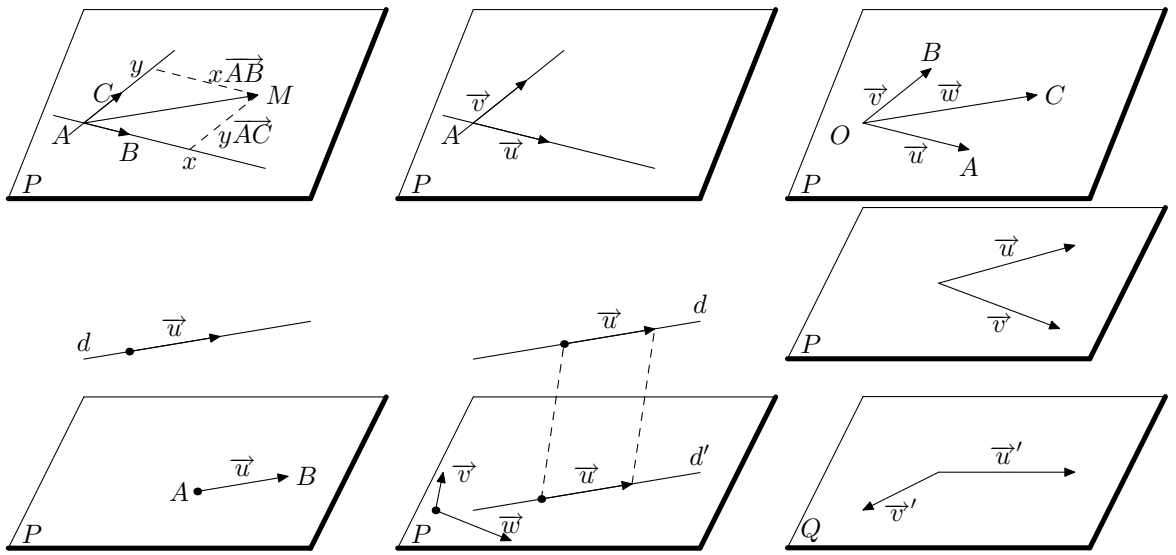


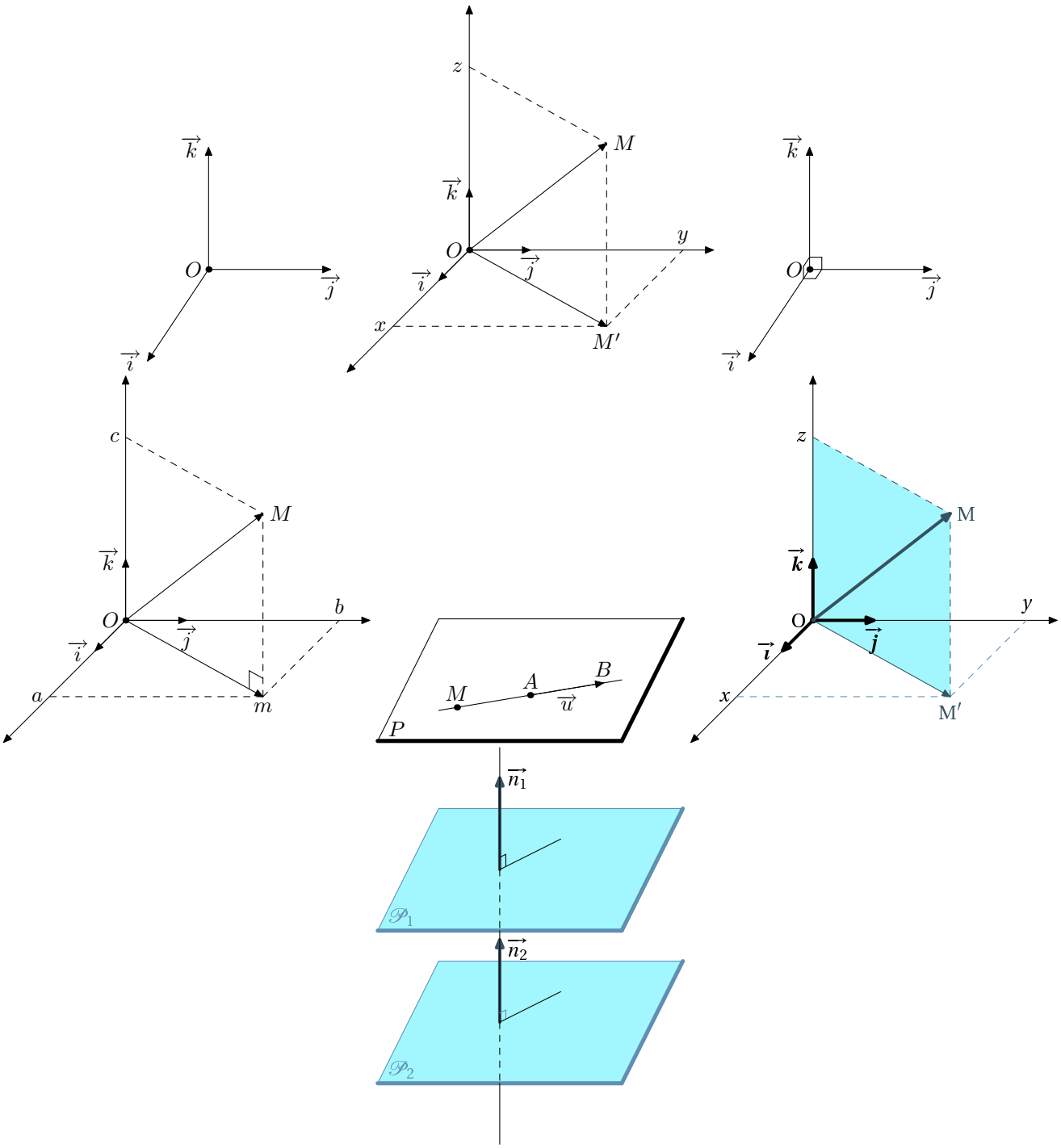
6.3 Diverses figures dans l'espace

Les sources sont à l'adresse

http://193.55.139.7/syracuse/metapost/cours/nivaud/fig1sc_geoespvec/fig1sc_geoespvec.mp

Le site contient encore bien d'autres magnifiques figures à copier-coller...

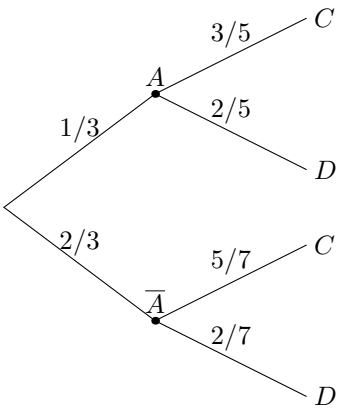



$\vec{k}$
O
$\vec{j}$
$\vec{i}$
z
M
$\vec{k}$
O
y
$\vec{j}$
$\vec{i}$
x
M'
$\vec{k}$
O
$\vec{j}$
$\vec{i}$
c
M
$\vec{k}$
O
b
$\vec{j}$
$\vec{i}$
a
m
M
A
B
$\vec{u}$
P
z
M
$\vec{k}$
O
y
$\vec{\imath}$
$\vec{j}$
x
M′
$\vec{n_1}$
$\mathscr{P}_1$
$\vec{n_2}$
$\mathscr{P}_2$

7 Arbre en proba

Pour produire rapidement un arbre 2 × 2

```
input TEX
beginfig(1)
arbre("$A$","$\overline{A}$","$C$","$D$","1/3","2/3","3/5","2/5","5/7","2/7");
endfig;
```



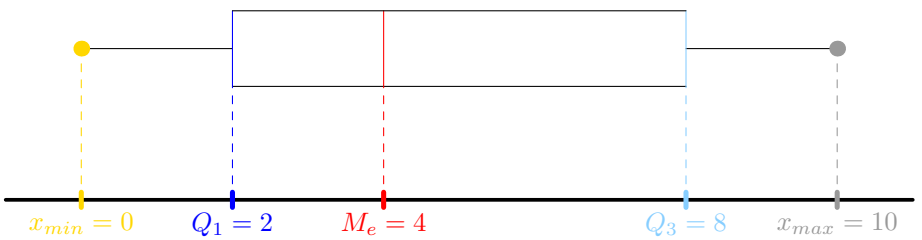
8 Boîtes à moustaches

Un moyen rapide de produire une boîte à moustache est d'utiliser la macro `moustache(min,Q1,Me,Q3,max,Unite,Origine_X,t)` où unité est l'unité de l'axe horizontal et t un paramètre qui vaut 2 si on veut les valeurs des quartiles avec leurs noms, 1 si on ne veut que les noms et 0 si on ne veut que les valeurs.

Par exemple

```
input newcourbes; input couleur; input TEX;
beginfig(1)
moustache(0,2,3,6,10,1cm,0,2);
endfig;
end
```

donne



9 Conversions

Voici quelques makefile pour convertir les fichiers produits par metapost en eps et en pdf. Il suffit d'enregistrer ces fichiers dans `/usr/bin` sous la forme par exemple de `mp2eps.sh`, d'éditer le fichier metapost sans préambule ni `beginfig` et de lancer `./mp2eps.sh fichier`. Dans un éditeur comme Kile ou emacs, on fabrique facilement un raccourci du type Alt+M,E pour cette conversion.

9.1 Conversion en eps

```
#!/bin/sh
FILE=${1%.*}
cat>mptemp.mp<<EOF
input /home/moi/Lycee/TS/figures/newcourbes/newcourbes;
input /home/moi/Lycee/TS/figures/couleur;
verbatimtex
%&latex
\documentclass{polymaths}
\begin{document}
etex
beginfig(1)
input $FILE
endfig;
end
EOF
mpost mptemp
cat>textemp.tex<<EOF
\documentclass{polymaths}
\thispagestyle{empty}
\begin{document}
\begin{TeXtoEPS}
\includegraphics{mptemp.1}
\end{TeXtoEPS}
\end{document}
EOF
latex textemp
dvips -o $FILE.eps -E 'textemp.dvi'
rm -f textemp.*
rm -f mptemp.*
```

9.2 Conversion en pdf

```
#!/bin/sh
FILE=${1%.*}
cat>mptemp.mp<<EOF
input /home/moi/Lycee/TS/figures/newcourbes/newcourbes;
input /home/moi/Lycee/TS/figures/couleur;
verbatimtex
%&latex
\documentclass{polymaths}
\begin{document}
etex
beginfig(1)
input $FILE
endfig;
end
EOF
mpost mptemp
cat>textemp.tex<<EOF
\documentclass{polymaths}
\thispagestyle{empty}
\begin{document}
\begin{TeXtoEPS}
\includegraphics{mptemp.1}
\end{TeXtoEPS}
\end{document}
EOF
```

```
latex textemp
dvips -o 'textemp.eps' -E 'textemp.dvi'
epstopdf textemp.eps --debug --outfile=$FILE.pdf
rm -f textemp.*
rm -f mptemp.*
```