```
  p3:=point([0,0,1],color=green,symbol=circle):
  p4:=point([.5,.5,0],color=green):
  p5:=point([0,.5,.5],color=blue):
  p6:=point([.5,0,.5],color=red):
  display([p1,p2,p3,p4,p5,p6]); # remarquez la différence entre un cercle
et un vrai point…
> l:=line([0,0],[3,4],color=gold,linestyle=3);
  display(l);
> l:=line([0,0,0],[1,1,1],color=maroon,linestyle=2);
  display(l);
```

Exercice :
+> Dessiner un carré, les arètes d'un cube, une étoile à 5 branches…

Nous reviendrons ultérieurement sur les graphismes (animations, etc…).

'Pratique' du logiciel

Manipulation des fichiers
Tout peut être fait par l'intermédiaire du manu File, ou à l'aide de boutons de la bare de commande pour les usages les plus courants.

Pour lire (charger dans Maple) un fichier, cliquez sur File / Open, ou sur le bouton  . Vous pouvez ensuite choisir le fichier maple (.mws) souhaité.

Pour ouvrir un nouveau document Maple (vierge), File / New ou  .

Pour enregistrer le document courant, File / Save ou  . Choisissez bien d'enregistrer sur la disquette, pour être sur de le retrouver. Pour enregistrer sous un autre nom que le nom courant, File / Save As.
Si plusieurs documents sont ouverts en même temps, vous pouvez passer de l'un à l'autre par Window.

Mise en page simple
Vous pouvez copier facilement des parties de texte , et les déplacer ou recopier.
Pour mettre en mémoire une zone de texte, sélectionnez-la à l'aide de la souris (positionnez-vous au début de la zone, bouton gauche, restez appuyés, allez à la fin de la zone, relacher le bouton gauche), puis copiez-la en mémoire à l'aide de Edit / Copy ou Ctrl+C ou  .
Pour recopier un élément mis auparavant en mémoire, positionnez-vous d'abord à l'endroit où vous voulez insérer celui-ci, puis Edit / Paste ou Ctrl+V ou  .
Pour copier une zone en mémoire et l'effacer du document Maple en même temps, sélectionnez la zone, puis Edit / Cut ou Ctrl+X ou  .

Vous pouvez toujours annuler la dernière opération effectuée avec Edit / Undo ou  , et refaire ce que vous venez d'annuler avec Edit / Redo ou  .
Plus simplement, pour déplacer une zone de texte, sélectionnez-la , puis placez la souris dessus, bouton gauche, restez appuyés, déplacez la souris là ou vous voulez déplacer la zone de texte, et relacher le bouton gauche

Pour insérer un commentaire dans le document Maple, cliquez sur  .

Pour insérer une nouvelle zone Maple [> après la zone en cours, Insert / Execution Group / After Cursor ou  .
Pour l'insérer avant la zone en cours, Insert / Execution Group / Before Cursor.

Divers

En cas de calcul sans fin (Maple 'bloqué'), cliquez sur  .

Aide
Pour lancer une recherche dans l'aide, vous pouvez cliquer sur Help / Topic search, ou taper le mot cherché puis ?, puis valider.

Autres options (rarement utilisées) :

- **ambientlight** = [r,g,b] fixe la coloration de la lumière d'ambiance rendue ; r, g et b (red/green/blue) doivent être des valeurs entre 0 et 1.
- **light** = [phi,theta,r,g,b] rajoute une lumière de la couleur [r,g,b], dans la direction [phi,theta].
- **lightmodel** = none, light1, light2, light3, ou light4, pour choisir le type d'éclairage utilisé.
- **projection** = r spécifie le type de projection utilisée, avec r entre 0 (proj. grand angle) et 1 (proj. orthogonale) (par défaut), ou FISHEYE, NORMAL, ou ORTHOGONAL (0, 0.5, ou 1).

Les styles possibles sont différents : POINT, HIDDEN, PATCH, WIREFRAME, CONTOUR, PATCHNOGRID, PATCHCONTOUR, ou LINE.

De même qu'avec plot, pour dessiner plusieurs graphes sur une même image, remplacer la fonction par un ensemble de fonctions.

Exemples :

```
> plot3d(sin(x+y),x=-1..1,y=-1..1);
  plot3d((1.3)^x*sin(y),x=-1..2*Pi,y=0..Pi,coords=spherical);
> plot3d([1,x,y],x=0..2*Pi,y=0..2*Pi,coords=toroidal(10),
scaling=constrained);
  plot3d(sin(x*y),x=-Pi..Pi,y=-Pi..Pi,style=contour);
> plot3d(sin(x*y),x=-Pi..Pi,y=-x..x,orientation=[45,0]);
  plot3d([x*sin(x)*cos(y),x*cos(x)*cos(y),x*sin(y)],x=0..2*Pi,y=0..Pi);
  plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,grid=[49,49],orientation=[-
90,30]);
> plot3d({sin(x*y),x+2*y},x=-Pi..Pi,y=-Pi..Pi);
  c1:= [cos(x)-2*cos(0.4*y),sin(x)-2*sin(0.4*y),y]:
  c2:= [cos(x)+2*cos(0.4*y),sin(x)+2*sin(0.4*y),y]:
  c3:= [cos(x)+2*sin(0.4*y),sin(x)-2*cos(0.4*y),y]:
  c4:= [cos(x)-2*sin(0.4*y),sin(x)+2*cos(0.4*y),y]:
  plot3d({c1,c2,c3,c4},x=0..2*Pi,y=0..10,grid=[25,15],style=patch,
color=sin(x));
```

Pour plus d'infos : **?plot3d**.

Lorsqu'un dessin est affiché, si vous cliquez sur celui-ci, la barre suivante apparaît :

Les deux premières zones vous indique l'angle de vue du dessin (theta et phi). Le premier groupe de boutons permet de modifier le style. Le second groupe permet de modifier les axes. Le dernier affiches les axes à la même échelle.
De plus, lorsqu dessin est affiché, il suffit de cliquer sur celui-ci et de bouger la souris sans relacher le bouton pour faire tourner le dessin en 3D.

En plus…
On peut aussi afficher, des objets (lignes, points, sphères, etc…). Il faut pour cela charger le pack plottools. Celui-ci contient les objets suivants : arc,.arrow, circle, cone, cuboid, curve, cutin, cutout, cylinder, disk, dodecahedron, ellipse, ellipticArc, hemisphere, hexahedron, hyperbola, icosahedron, line, octahedron, pieslice, point, polygon, rectangle, semitorus, sphere, tetrahedron, et torus (!!). Pour des détails sur ceux-ci, utilisez **?plottools**. Nous nous contenterons de line et point.

Un point est défini comme suit : **p:=point([*coordonnées*], *options*);**. Les coordonnées (2 ou 3) du point sont donc regroupées en une liste, et les options sont les mêmes que pour **plot** ou **plot3d**.
Une ligne est définie comme suit : **l:=line([coord. 1], [coord 2], options);**. Les coordonnées 1 et 2 correspondent aux extrémités de la ligne, et les options sont les mêmes que pour **plot** ou **plot3d**.
Une fois que les lignes et points sont définis, on les affiche avec **display()**.

Exemples :

```
> p:=point([0,0],color=green):
  display(p,axes=boxed);
> p1:=point([1,0,0],color=blue,symbol=circle):
  p2:=point([0,1,0],color=red,symbol=circle):
```

Graphiques

Pour utiliser la bibliothèque graphique, il faut la charger dans Maple à l'aide de
> with(plots);
On a déjà vu les fonctions **implicitplot()** et **implicitplot3d()**, dans le TD 4 , qui représentent graphiquement les solutions d'une équation.

2D
L'instruction de base est
> plot(*fonction*, *intervalle horizontale*, *intervalle verticale*, *options*);
L'intervalle verticale est facultative, de même que chacune des options.
Les intervalles se définissent sous la forme classique des intervalles Maple, c'est-à-dire **a..b**.
Vous pouvez mettre simplement **a..b**, ou **x=a..b**, ou autre chose si la variable n'est pas x. Si vous n'indiquez rien, l'intervalle horizontale est –10 10, et la verticale est ajustée au graphe.
Pour la fonction, vous pouvez indiquer soit le nom d'une fonction définie précédemment, soit la valeur (l'expression explicite) de la fonction dont vous voulez dessiner le graphe.
Les options possibles sont, entre autre :

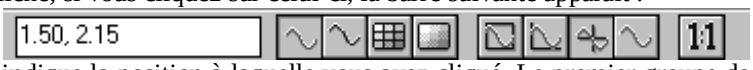
- **axes =** FRAME, BOXED, NORMAL, ou NONE, pour choisir le type d'axes à l'affichage.
- **color =** aquamarine, black, blue, navy, coral, cyan, brown, gold, greenn, gray, grey, khaki, magenta, maroon, orange, pink, plum, red, sienna, tan, turquoise, violet, wheat, white ou yellow (!!) pour choisir la couleur de dessin.
- **coords =** bipolar, cardioid, cassinian, elliptic, hyperbolic, invcassinian, invelliptic, logarithmic, logcosh, maxwell, parabolic, polar, rose, ou tangent (!!) pour définir le système de coordonnées utilisé (coordonnées cartésiennes par défaut).
- **labels =** [x,y] pour définir les noms des axes.
- **style =** LINE, POINT, PATCH ou PATCHNOGRID, pour le style de dessin.
- **title =** `blablabla` pour mettre un titre.
- **etc…**

Il faut les indiquer à la suite les unes des autres, séparées par des virgules, l'ordre étant quelconque.
Pour dessiner plusieurs graphes sur une même image, remplacer la fonction par un ensemble de fonctions.

Exemples :
```
> plot(cos(t)+sin(t), t=0..Pi);
  plot(tan(t),t=-Pi..Pi);
  plot([sin,cos,-Pi..Pi]);
  plot(sin);
> plot([sin(x),x-x^3/6],x=0..2,color=[red,blue],style=[point,line]);
> s:=t->100/(100+(t-Pi/2)^8): r:=t->s(t)*(2-sin(7*t)-cos(30*t)/2):
  plot([r(t),t,t=-Pi/2..3/2*Pi],numpoints=2000,coords=polar,axes=none);
> s:=seq([n,cos(n/10)],n=1..10);
  plot([s], x=0..15,style=point,symbol=circle);
```

Pour plus d'infos : **?plot**.

Lorsqu'un dessin est affiché, si vous cliquez sur celui-ci, la barre suivante apparaît :



La première zone vous indique la position à laquelle vous avez cliqué. Le premier groupe de boutons permet de modifier le style. Le second groupe permet de modifier les axes. Le dernier affiches les axes à la même échelle.

3D
L'instruction de base est
> plot3d(*fonction*, *intervalles*, *options*);
Ici aussi, les interervalles sont facultatives, de même que les options.
Pour la (les) fonction(s), vous pouvez indiquer soit le nom ou l'expression précise.
Les options possibles sont les mêmes que pour plot, avec, en plus, des options spécifiques au rendu 3D. Principalement :

- **orientation** = [theta,phi] indique l'angle de vue du dessin 3D, en coord. sphériques.

Maple TD 6

Infos diverses

Je vous rappelle qu'il n'y a pas de séance mardi prochain (21 novembre).
De plus, étant donné de lourdes incertitudes sur la disponibilité des locaux et le déroulement du déménagement, **l'examen partiel est reporté à une date ultérieure** (probablement le 12 décembre).

Rappels
Algèbre linéaire et espaces vectoriels (fin du TD 5)
Graphiques
'Pratique' du logiciel

Rappels

L'ensemble des fonctions d'algèbre linéaire doit être chargé dans Maple à l'aide de l'instruction

```
> with(linalg);
```

Un vecteur est défini par

```
> vector([x1,x2,x3,x4]);
> vector(4,0);
> f:=i->(-1)^i; vector(4,f);
```

La syntaxe est semblable pour les matrices :

```
> matrix(2,3,0);
> matrix(2,2,[x1,x2,x3,x4]);
> matrix([[1,2],[3,4]]);
> matrix(2,3,[1,2,3,4,5,6]);
> f:=(i,j)->(-1)^(i+j); matrix(3,3,f);.
```

Le produit de la matrice A par la matrice B est **`A&*B`**. La multiplication de la matrice M par le scalaire a est **`a*M`**. On évalue par **`evalm()`**.

Algèbre linéaire et espaces vectoriels (fin du TD 5)

Voyons maintenant quelques considérations classiques sur les espaces vectoriels : dimension, familles libres ou génératrices, bases, sous-espace engendré…

La principale fonction est **`basis()`**, qui donne une base de l'espace vectoriel engendré par une famille (*liste*) de vecteurs (de même dimension).

On rappele que

- la dimension **n** de l'espace initial est nécessairement la dimension commune des **k** vecteurs de la famille.
- la dimension **l** de l'espace engendré est exactement le nombre de vecteurs de la base de celui-ci.
- si une famille est liée, la dimension **l** de l'espace qu'elle engendre est < au nombre **k** de vecteurs qu'elle contient.
- si elle est libre, la dimension **l** de l'espace qu'elle engendre est = au nombre **k** de vecteurs qu'elle contient.
- si une famille est génératrice, la dimension **l** de l'espace qu'elle engendre est = à la dimension **n** de l'espace initial.
- Si elle ne l'est pas, la dimension **l** de l'espace qu'elle engendre est < à la dimension **n** de l'espace initial.

Exercices :
>En utilisant **`nops()`** et **`basis()`**, écrivez 3 procédures qui donnent le nombre de vecteurs d'une famille de vecteurs, la dimension de l'espace initial et la dimension de l'espace engendré.
>Après une synthèse des rappels ci-dessus, écrivez des procédures qui indiquent si une famille de vecteurs est libre, liée, génératrice ou non, et si c'est une base. Vérifiez dans les 4 cas de figure possibles.