Université Pierre et Marie Curie.
Année universitaire 2012/2013

Unité d'enseignement LP206.
Mathématiques pour physiciens I

Premiers pas avec MATHEMATICA

I. Introduction

MATHEMATICA est un logiciel de calcul formel qui permet de manipuler des expressions mathématiques symboliques. Contrairement à la plupart des langages de programmation (tels que C et FORTRAN), il peut manier non seulement des quantités numériques (nombres entiers, réels ou complexes), mais aussi des polynômes, des fonctions, des séries, etc.

MATHEMATICA permet d'effectuer des opérations telles que la dérivation, l'intégration, le calcul de limites, ainsi que la simplification d'expressions compliquées et la résolution d'équations. Grâce à l'interface graphique, il est également possible de visualiser les résultats en traçant des courbes ou même des surfaces dans l'espace tridimensionnel.

La version standard de MATHEMATICA contient des milliers de fonctions prédéfinies. Il est possible d'augmenter encore les possibilités en incluant des bibliothèques de fonctions adaptées à tel ou tel domaine de la physique, des mathématiques, de la chimie, de l'ingénierie, etc. De ce fait, MATHEMATICA est devenu un outil de travail quotidien pour bon nombre de chercheurs.

II. Prise en main

En LP206, nous allons utiliser la version graphique de MATHEMATICA, qui est la plus simple.

Pour bien profiter de cette séance d'initiation, il faut ouvrir une session et retaper toutes les commandes sur fond gris pour s'accoutumer au style de MATHEMATICA. *N'hésitez pas à faire vos propres expériences en essayant quelques variantes. Vérifiez à chaque étape que vous comprenez parfaitement le résultat obtenu.*

Pour lancer MATHEMATICA,

1. cliquez sur « `Bureau de L'UTĚS` » ;
2. entrez votre identifiant UPMC et votre mot de passe ;
3. allez dans « `Pédagogie / Mathématiques / Mathematica 8` » ;
4. cliquez sur « `Notebook` ».

On se retrouve avec une feuille de calcul vierge. On peut alors écrire un calcul qu'on exécutera en tapant la combinaison `Majuscule` + `Entrée` ※1.

Ne pas utiliser la touche `Alt Gr` *à L'UTĚS : elle risque d'introduire des caractères imperceptibles qui feront échouer l'exécution ; utiliser* `Ctrl` + `Alt` *à la place. Si vous avez utilisé* `Alt Gr`*, il est plus sage de retaper l'instruction…*

Voici quelques exemples simples :

`2␣3␣4` ;

`(3 + 4)^2 - 2 (3 + 1)/4` .

On notera que le circonflexe ※2 indique la puissance et que l'espace entre deux quantités (noté ␣ ※3 dans le premier calcul) sous-entend un signe de multiplication ※4 (on peut aussi utiliser un astérisque, plus explicite entre deux nombres). Les espaces dans d'autres positions sont simplement ignorés (on peut en mettre pour rendre l'expression plus lisible).

1. On peut aussi utiliser la touche `Entrée` du pavé numérique au lieu de cette combinaison.
2. Ne pas utiliser la touche `²` (2 en exposant) à gauche du `&` pour une élévation au carré.
3. Nous n'utiliserons ce symbole que lorsqu'il y aura un risque de confusion.
4. Attention, les multiplications, mêmes implicites, ont la même priorité que les divisions. À priorité égale, ces opérations sont effectuées de gauche à droite : ainsi, le résultat de **`1/2 a`** est **`a/2`**, pas **`1/(2 a)`**.

On vient de créer deux cellules d'entrée, **In[1]** et **In[2]**, et deux cellules de sortie, **Out[1]** et **Out[2]**. On aurait d'ailleurs pu mettre les deux instructions dans la même cellule d'entrée en passant à la ligne : on aurait quand même obtenu deux cellules distinctes en sortie.

Vous pouvez d'ores et déjà sauver la feuille de calcul (menu « `File / Save as` »); si vous vous appelez Jean Dupont et qu'il s'agit de la séance d'initiation (séance 0), appelez votre fichier « `DupontJean0` »[※5]. Pensez à sauver le fichier après chaque calcul par la suite (menu « `File / Save` » ou `Ctrl` + `S`) : ceci vous permettra de conserver votre travail si MATHEMATICA se bloque. Pour interrompre un calcul trop long, il suffit généralement de sélectionner « `Evaluation / Abort Evaluation` », mais il est parfois nécessaire de quitter le « noyau » (là où les calculs sont réalisés ; *kernel* en anglais) à l'aide du menu « `Evaluation / Quit Kernel` », voire l'application.

Sur une feuille de calcul compliquée, on a intérêt à ajouter des commentaires, soit pour se souvenir ultérieurement de la logique du calcul, soit pour le présenter à autrui :

```
(* On calcule le nombre de choix de 13 objets parmi 30. *)
30!/(13! 17!)
```
.

On peut structurer le document en convertissant certaines cellules en titre général, titre de section, texte, etc., à l'aide du menu « `Format / Style` », notamment pour numéroter les questions.

Les calculs en MATHEMATICA sont en général faits avec une précision arbitraire. Par exemple, les commandes

```
2^100
50!
```

donnent des entiers exacts et assez longs. Si l'on désire un résultat approximatif, on peut écrire

```
N[Pi]
N[2^100]
N[E, 50]
```
.

Par défaut, MATHEMATICA affiche les valeurs approchées avec 6 chiffres significatifs, quitte à passer en notation exponentielle si besoin est. Dans le dernier calcul, on a demandé une précision plus élevée en donnant le nombre de chiffres significatifs en second argument.

Il faut noter que les entiers sont considérés comme des expressions exactes, tandis qu'une expression avec un point décimal[※6] est une approximation numérique (par défaut correcte à 6 chiffres). Si l'on mélange plusieurs approximations numériques, MATHEMATICA ajuste la précision à celle du terme le moins précis, comme le montre cet exemple :

```
1. + N[Pi, 50]
```
.

III. Les fonctions prédéfinies

On vient d'utiliser la fonction **N**. En MATHEMATICA, toutes les fonctions (prédéfinies ou non) s'expriment en fonction d'une ou plusieurs variables entre *crochets* : les *parenthèses* sont réservées au regroupement de termes et les *accolades* aux listes. S'il y a plusieurs variables, celles-ci sont séparées par des virgules.

MATHEMATICA possède toute une collection de fonctions prédéfinies. Par convention, ces fonctions commencent par une majuscule. Ceci vaut aussi pour les constantes prédéfinies : outre **Pi** (π) et **E** (e) que nous avons déjà vues, nous utiliserons **I** (i) et **Infinity** (∞).

Voici quelques exemples de fonctions :

`Sqrt[36]` ;

`Exp[2.3]` ;

5. Ne pas utiliser d'espace ni d'accent, de cédille, d'apostrophe ou tout autre caractère non ASCII dans le nom du fichier.
6. MATHEMATICA utilise le point pour séparer les parties entière et décimale d'un nombre.

`Log[E^5]` ;

`Log[2, 1024]` ;

`Sin[3 Pi/2]` .

`Sqrt[x]` est la racine carrée de x. **`Log[x]`** est le logarithme népérien (fonction « ln ») ; **`Log[2, x]`**, le logarithme en base 2. D'autres fonctions indispensables sont **`Cos`**, **`Tan`**, **`ArcSin`**, **`Abs`** (valeur absolue), **`Floor`** (partie entière), **`Sign`** (signe), **`Re`** (partie réelle), **`Im`** (partie imaginaire), **`Cosh`**, **`ArcSinh`**, etc.

Attention, Mathematica ne retourne pas toujours le résultat attendu :

`(-1.)^(1/3)`

ne donne pas **`-1`** mais la valeur numérique de $(1 + \mathrm{i}\sqrt{3})/2$, une autre racine cubique de -1.

IV. L'aide en ligne

Étant donné le nombre de possibilités, l'aide en ligne devient vite indispensable. Il suffit de mettre le signe **`?`** devant le nom de la fonction pour accéder à l'aide. La variante **`??`** donne plus d'informations :

`?Cos` ;

`??FactorInteger` .

Cliquer sur ≫ pour avoir encore plus de détails, notamment connaître les options : Mathematica ouvre alors une fenêtre d'aide. En bas de celle-ci, des fonctions apparentées et des tutoriels sont souvent proposés[※7].

De la même manière, la commande **`?x`** rappelle les différentes affectations d'une variable x définie par l'utilisateur.

Si on ne se souvient pas de la commande exacte, on peut remplacer les lettres inconnues par **`*`** :

`?Si*` .

V. Calculs plus avancés

Il n'est pas recommandé de faire de copier-coller de résultats obtenus précédemment, ni de les retaper. Il est en revanche possible de les rappeler et de les combiner pour faire des calculs plus complexes. On peut utiliser **`%`** pour appeler le résultat précédent, **`%%`** pour celui d'avant, et ainsi de suite. Exemple :

`7^2` ;

`% + 1` ;

`3 % + %^2 + %%` .

L'expression **`%`*n***, où n est une valeur numérique entière, renvoie au résultat **`Out[`*n*`]`**.

Attention si vous revenez en arrière et insérez d'autres calculs : ce qui compte pour Mathematica, c'est l'ordre d'exécution des calculs, pas leur disposition dans la feuille de calcul. Pour bien comprendre ceci, tapez les expressions ci-dessous en gris :

In[*n*] = `2;` ,

7. Pour des informations encore plus complètes, on peut utiliser un moteur de recherche en ajoutant le mot « `Mathematica` » à la recherche. Une introduction assez complète à Mathematica est également disponible en anglais à l'adresse `http://www.math.armstrong.edu/faculty/hollis/classes/3900/MathematicaIntro.pdf`.

`In[n+1] =` `%^2`

et

`In[n+2] =` `3;` .

Vous obtenez **`Out[n+1] = 4`**. Réévaluez maintenant la cellule n° $n + 1$ (c.-à-d. sélectionnez-la et tapez `Majuscule` + `Entrée`) : celle-ci devient **`In[n+3]`** et **`Out[n+3] = 9`**. Attention également si vous réexécutez l'ensemble de la feuille※8 : les **`%n`** ne renverront plus aux mêmes instructions.

Les variables constituent un outil plus puissant :

`x = 5` ;

`2 x - 3 x^2` ;

`coulomb = 1.602␣10^-19` .

On peut choisir n'importe quelle suite de lettres ou de chiffres, à condition que le premier symbole soit une lettre. On peut aussi utiliser des lettres grecques : pour entrer le caractère ν par exemple, taper `Échap`, `n`, `Échap`. Il est recommandé de toujours commencer par une lettre minuscule pour éviter toute confusion avec les fonctions et variables prédéfinies : celles-ci commencent en effet par une majuscule. Éviter également le symbole **`_`**.

Si l'on veut réutiliser le même nom de variable **`x`** dans un autre contexte plus tard, on peut le libérer avec **`Clear[x]`** (il est bien entendu interdit de libérer les variables prédéfinies). Il est même recommandé d'effacer régulièrement toutes les variables que l'on a créées, notamment au début d'une nouvelle partie. Pour cela, taper

```
ClearAll["Global`*"]
```

(on obtient l'accent grave en tapant `Ctrl` + `Alt` + `è`, puis la barre d'espacement). Il vaut mieux mettre aussi cette instruction en début de fichier, au cas où l'on réexécuterait l'ensemble d'une feuille de calcul (menu « `Evaluation / Evaluate Notebook` ») sans avoir préalablement quitté le noyau (menu « `Evaluation / Quit Kernel` »).

Avec certaines fonctions, on peut imposer des hypothèses sur le domaine d'une variable. Essayez

```
Simplify[x - Floor[x]]
Simplify[Abs[y]]
```
,

```
$Assumptions = {Element[x, Integers], y >= 0}
```

et à nouveau

```
Simplify[x - Floor[x]]
Simplify[Abs[y]]
```
.

Pour supprimer les hypothèses, tapez

`$Assumptions = true` .

Pour une hypothèse utilisée ponctuellement, la fonction **`Assuming`** ou l'option **`Assumptions`** sont plus appropriées. Exemple :

`Simplify[x - Floor[x], Assumptions -> Element[x, Integers]]` .

Remarquez que le contenu d'une option est indiqué à l'aide de **`->`**.

Certains calculs nécessitent l'utilisation d'une variable d'itération. Par exemple, $\sum_{i=1}^{10} i^2$ peut se calculer ainsi :

`Sum[i^2, {i, 1, 10}]` .

8. C'est ce que fera votre enseignant après avoir supprimé toutes les sorties avec « `Cell / Delete All Output` ».

Les produits se calculent avec la commande **Product**.

On peut également utiliser des listes, c'est-à-dire des objets regroupés par des accolades :

`liste = {2, 3, 5, 7, 11}` ;

`Sum[i^2, {i, liste}] (* "i" prend ses valeurs dans l'ensemble "liste". *)` .

Il existe de nombreuses possibilités pour manipuler des listes (surtout lorsqu'elles sont imbriquées). Pour extraire un élément d'une liste, on utilise des crochets doubles :

`liste[[4]]` .

On peut exécuter plusieurs commandes sur une même ligne en les séparant par des points-virgules. La sortie d'une commande terminée par **;** n'est pas affichée. Exemple :

`x = 4; y = 6; z = y + 6` .

VI. Calculs symboliques

Une des forces de Mathematica est de pouvoir manipuler des expressions symboliques, même compliquées, de manière efficace :

`Clear[x, y]; (x - 1)^10 (y - 2)^5` ;

`Expand[%]` ;

`Factor[%]` .

Together réduit au même dénominateur une expression contenant des fractions ; **Apart** fait l'inverse. **Cancel** simplifie si possible une fraction.

À l'aide d'une « règle » (*rule* en anglais) telle que ***expr* /. *var* -> *val***, on peut attribuer une valeur à une variable utilisée dans une expression et évaluer cette dernière :

`Sum[Sin[k␣x], {k, 1, 3}]` ;

`% /. x -> Pi/4` .

Bien noter la différence avec l'attribution d'une valeur définitive à la variable, comme dans **x = Pi/4**.

On peut simplifier des expressions compliquées ou bien les stocker dans des variables :

`expr = ((1 + Sqrt[5])/2)^10` ;

`Simplify[expr]` ;

`Sin[x]^2 + Cos[x]^2` ;

`Simplify[%]` .

Il est possible de comparer une expression à une autre :

`Sin[x]^2 + Cos[x]^2 == 1` ;

`Simplify[%]` .

Bien noter la différence entre les symboles **=** et **==** : le premier correspond à une attribution, le second à une comparaison. D'autres opérateurs de comparaison sont **!=** ($\neq$), **>**, **<=** ($\leqslant$), etc.

Mathematica peut résoudre des équations. Il n'est pas nécessaire de spécifier pour quelle variable on veut les résoudre, sauf s'il y en a plusieurs :

`Solve[21 x^2 - 68 x + 55 == 0, x]` ;

`Solve[x^3 - 2 x^2 + x - 1 == 0]` ;

`Simplify[%]` ;

`FullSimplify[%]` .

Parfois, il n'est pas possible de résoudre une équation en termes de fonctions connues, ou bien la résolution est possible mais donne une expression très compliquée :

`Solve[x^5 + x + 1 == 0, x]` .

On peut alors chercher des solutions numériques :

`NSolve[x^5 + x + 1 == 0, x]` .

Cette stratégie échoue si l'équation est transcendantale※[9]. On peut néanmoins trouver numériquement une racine près d'une valeur initiale spécifiée :

`eqn = 3 Cos[x] == Log[x]` ;

`NSolve[eqn, x]` ;

`FindRoot[eqn, {x, 1}]` ;

`FindRoot[eqn, {x, 10}]` .

Remarquer que les solutions sont données sous forme d'une liste de *règles*, même s'il n'y en a qu'une : ici, **`{x -> 13.1064}`**. Si vous tapez

`3 Cos[%] - Log[%]` ,

ça ne marche pas. Il faut écrire

`3 Cos[x] - Log[x] /. %%` .

Si l'on ne veut pas utiliser les **`%`**, on peut aussi taper

```
sol = FindRoot[eqn, {x, 10}]
eqn /. sol
```
.

VII. Nombres complexes

Il est possible de faire des calculs avec des nombres complexes :

`z = (4 + 3 I)/(2 - I)` ;

`Re[z]` ;

`Im[z]` ;

`N[Exp[2 + 9 I]]` .

Si $z = a + \mathrm{i}\, b$, alors le conjugué $\overline{z} = a - \mathrm{i}\, b$ est obtenu par **`Conjugate[z]`**. Dans l'écriture $z = |z|\, \mathrm{e}^{\mathrm{i}\, \phi}$, le module $|z|$ est donné par **`Abs[z]`** et l'argument ϕ par **`Arg[z]`**.

Dans les manipulations courantes, Mathematica suppose que tous les nombres peuvent être complexes. Il est cependant possible de développer une expression en imposant aux variables des valeurs réelles. Comparer

`Sin[x + I␣y]` ,

9. Une équation algébrique est du type $P(x) = 0$, où P est un polynôme à coefficients entiers. Une équation transcendantale est une équation qu'on ne peut pas ramener à une expression de ce genre.

`Expand[%]`

et

`ComplexExpand[%]` .

Mathematica connait bien sûr la formule de Moivre, mais développe le résultat avec beaucoup de circonspection :

`n = 5 ; moivre = (Cos[x] + I␣Sin[x])^n` ;

`FullSimplify[moivre]` ;

`ComplexExpand[%]` .

Les fonctions **`ExpToTrig`** et **`TrigToExp`** permettent de convertir des expressions contenant des exponentielles de nombres complexes en sinus et cosinus, et inversement.

VIII. Différentiation et intégration

Mathematica peut dériver une fonction par rapport à une de ses variables (dérivée partielle) :

`D[Sqrt[Tanh[x]], x]` ;

`D[ArcTan[x␣y], x]` .

On peut aussi utiliser **`f'[x]`**, **`f''[x]`**, etc., pour une fonction d'une seule variable.
Les différentielles totales se calculent ainsi :

`Dt[x^n]` ,

où la différentielle dx est notée **`Dt[x]`**. Remarquez que **`n`** est a priori considéré comme une variable (l'option **`Constants`** permet de préciser que **`n`** est une constante).
Les intégrales indéfinies (= primitives) se calculent avec la fonction **`Integrate`** :

`Integrate[ArcTan[x], x]` .

Il en est de même pour les intégrales définies (propres ou impropres) :

`Integrate[Tan[x], {x, 0, Pi/4}]` ;

`Integrate[Sin[x]/x, {x, 0, Infinity}]` .

Idem pour les intégrales multiples :

`Integrate[x^2 + y^2, {x, 0, 1}, {y, 0, x}]` .

(L'intégration se fait d'abord sur la variable y, puis sur la variable x.)
Parfois le résultat nous permet de faire connaissance avec des fonctions exotiques :

`Integrate[Log[1 + Tanh[x]], {x, 0, Pi}]` ;

`?PolyLog` .

Si **`Integrate`** ne donne pas de résultat ou produit une expression trop compliquée, on peut essayer une intégration numérique avec **`NIntegrate`** :

`NIntegrate[Log[1 + Tanh[x]], {x, 0, Pi}]` .

IX. Sommes et produits

Les sommes finies et les séries se calculent aisément :

`Sum[x^i/i, {i, 1, 7}]` ;

`Sum[1/k^2, {k, 1, Infinity}]` ;

`Sum[x^n/n, {n, 1, Infinity}]` .

Que peut bien vouloir dire l'expression

`Sum[1/x^4, {x, 1, Infinity, 2}]` ?

(Utiliser l'aide en ligne.)

Les produits se calculent de la même manière :

`Product[(x - a), {a, 1, 10}]` ;

`prod = Product[(3 j - 1)^2/((3 j) (3 j - 2)), {j, 1, Infinity}]` .

Il est (peut-être) rassurant de voir que Mathematica ne connaît pas *toutes* les règles de simplification algébrique :

`FullSimplify[prod == 3 Gamma[1/3]^3/(4 Pi^2)]` ;

`N[%, 100]` .

X. Définition de fonctions ; programmes

Nous pouvons définir nos propres fonctions :

```
a = 1
f[x_] := 3 - 2 x + a x^2
g[x_] = 3 - 2 x + a x^2
h[x_, y_] := x Sin[y^2/x]
```
.

Le tiret bas (_) est nécessaire dans le membre de gauche et interdit dans celui de droite : il identifie les variables dont la fonction dépend.

Le symbole `:=` désigne une affectation différée tandis que `=` est utilisé pour une affectation immédiate. Dans le premier cas, la fonction est évaluée quand elle est appelée ; dans le second cas, dès qu'elle est définie. Pour le voir, redéfinissons a et évaluons f et g en $x = 1$:

```
a = 2
f[1]
g[1]
```
.

On constate que $f(1)$ a été calculée avec $a = 2$, et $g(1)$ avec $a = 1$.

L'affectation différée permet de définir des suites par récurrence :

```
u[1] = 1 ;
u[2] = 1 ;
u[n_] := u[n-1] + u[n-2] ;
u[5]
```
.

Mathematica fournit de nombreuses fonctionnalités pour éviter un travail répétitif. En particulier, on peut faire des boucles,

`Do[Print[n!], {n, 1, 10}]` ,

ainsi que des structures de contrôle,

`Do[If[Mod[n, 7] != 0 && Mod[n, 5] != 1, Print[n]], {n, 1, 50}]` ※10.

Cette instruction énumère les nombres entre 1 et 50 qui ne finissent pas par un « 1 » ou un « 6 » et qui ne sont pas divisibles par 7.

La fonction **`If`** ci-dessus permet de définir des fonctions par morceaux. On peut aussi utiliser **`Which`**, **`Switch`** ou **`Piecewise`**.

XI. Séries de Taylor

La série de Taylor d'une fonction $f(x)$ autour de x_0 à l'ordre n est donnée par **`Series[f[x], {x, x0, n}]`**. Quelques exemples :

`dev1 = Series[Exp[x], {x, 0, 5}]` ;

`dev2 = Series[(1 + x)^n, {x, 0, 2}]` .

Noter que Mathematica garde la précision du développement au moyen de la notation **`O[x]^n`** (= $O(x^n)$). De même que pour les approximations numériques, la combinaison d'objets de précisions différentes est faite de manière cohérente :

`dev1 + dev2` .

La fonction **`Normal[...]`** élimine le terme $O(x^n)$ et permet de traiter le résultat comme une expression ordinaire par la suite :

`Normal[dev1 + dev2]` ;

`Solve[% == 0, x]` .

XII. Limites

Voici d'abord deux tentatives infructueuses pour déterminer $\lim_{x\to 0}(\sin x/x)$, suivies par la solution qui marche :

`t = Sin[x]/x` ;

`t /. x -> 0` ;

`t /. x -> 0.01` ;

`Limit[t, x -> 0]` .

Pour les limites à gauche et à droite, consultez l'aide de **`Limit`**.

10. **`&&`** représente le « et » logique ; les symboles **`!`** et **`||`** sont utilisés respectivement pour les connecteurs logiques « non » et « ou » (inclusif). **`Mod[n, 5]`** est le reste de la division euclidienne de n par 5.

XIII. Équations différentielles

Définissons d'abord une équation différentielle et sa condition initiale :

```
equadiff = y'[x] == a␣y[x]
condinit = y[0] == 1
```
.

La solution générale et celle qui satisfait la condition initiale s'obtiennent alors par

```
DSolve[equadiff, y[x], x]
```

(dans ce résultat, `C[1]` représente une constante n^o 1) et

```
DSolve[{equadiff, condinit}, y[x], x]
```
.

Noter que le deuxième argument est la fonction pour laquelle on veut résoudre l'équation différentielle, tandis que le troisième argument est la variable indépendante. Pour une résolution numérique, utiliser **NDSolve**.

Il est également possible de résoudre des équations différentielles couplées :

```
Clear[t]
DSolve[{x'[t] == y[t], y'[t] == x[t]}, {x[t], y[t]}, t]
```
.

XIV. Graphes

Tracer le graphe d'une fonction ou d'une collection de fonctions est aisé :

```
Plot[Sin[x], {x, 0, 2 Pi}]
```
;

```
Plot[{Sin[2 x], Cos[3 x]}, {x, 0, 2 Pi}]
```
.

De nombreuses options permettent de personnaliser le graphe. Exemple :

```
Plot[Sin[x^2], {x, 0, 3}, Frame -> True, GridLines -> Automatic]
```
.

Vous pouvez consulter l'aide en ligne pour plus de détails : voir en particulier les options **PlotRange** et **PlotStyle**.

L'affectation différée d'une fonction peut poser problème à **Plot** quand elle est trop tardive :

```
f[x_] := a␣x^2
g[x_] := D[f[x], x]
a = 1
Plot[g[x], {x, -1, 1}]
```
.

Il faut alors utiliser la fonction **Evaluate** :

```
Plot[Evaluate[g[x]], {x, -1, 1}]
```
.

Pour une fonction de plusieurs variables, il peut être utile de visualiser ses courbes de niveau :

```
ContourPlot[Sin[x] Sin[y], {x, -2, 2}, {y, -2, 2}]
```
.

On peut également faire des graphes paramétriques en trois dimensions. Par exemple, une hélice et un tore sont produits respectivement par

```
ParametricPlot3D[{u␣Sin[t], u␣Cos[t], t/3}, {t, 0, 15}, {u, -1, 1}]
```

et

```
ParametricPlot3D[{Cos[t] (3 + Cos[u]), Sin[t] (3 + Cos[u]), Sin[u]},
    {t, 0, 2 Pi}, {u, 0, 2 Pi}]
```
.

Pour connaître tous les types de graphiques possibles, tapez

```
?*Plot*
```
.

XV. À la fin de la séance et entre deux séances

Quitter puis relancer MATHEMATICA (ou sélectionner « `Evalutation / Quit Kernel` ») et tout réexécuter en bloc à l'aide de « `Evaluation / Evaluate Notebook` ». Corriger les éventuelles erreurs, puis supprimer toutes les sorties avec « `Cell / Delete All Output` ».

Enregistrer ou copier (et non déplacer, de manière à pouvoir travailler dessus entre deux séances !) le fichier dans l'espace partagé indiqué par votre enseignant (quelque chose du genre « `231 sur 'serveur de gestion du parc [...]` » si vous êtes en salle 231). Sinon, envoyer le fichier à l'adresse électronique que celui-ci vous a fournie (cliquer sur « `Internet` » sur le bureau pour accéder à la messagerie).

Vous pouvez travailler sur le fichier à L'UTĚS ou chez vous (demandez une licence MATHEMATICA en allant dans l'onglet « `Mes outils` » de `mon.upmc.fr`).