

---

# M 1 : Introduction à *Mathematica*

---

## Préambule

---

### I. Introduction (15 min par l'enseignant, et + par l'étudiant...)

#### ■ A. Description générale de *Mathematica*

##### ■ 1. Description générale de *Mathematica*

*Mathematica* est un système pour faire des mathématiques avec un ordinateur. Il est développé par Wolfram Research. C'est à la fois :

un logiciel de calcul formel :

manipulation d'expressions mathématiques sous forme symbolique,  
comme le calcul de dérivées, de primitives ou d'intégrales exactes, la simplification d'expressions, etc...

un logiciel de calcul numérique :

évaluation d'expressions mathématiques sous forme numérique (précision machine ou arbitraire),  
comme le calcul des premières décimales du nombre  $\pi$ , évaluation approchée d'intégrales, etc...

un langage de programmation cohérent qui autorise différents paradigmes de programmations : fonctionnelle, par règles, mais aussi procédurale, graphique, etc ...

L'objectif des séances M1 et M2 est précisément de découvrir ce que tout ceci veut dire.

*Mathematica* est très utilisé dans de très nombreux domaines principalement dans la recherche scientifique, les mathématiques financières, l'enseignement, et dans l'industrie, mais aussi pour la conception de bâtiments, de jeux, ...

##### ■ 2. Où trouver *Mathematica* ?

L'UPMC a une licence globale permettant d'installer *Mathematica* sur tous les ordinateurs d'enseignement de l'université, mais aussi sur l'ordinateur personnel des étudiants ou des enseignants. En particulier, il est disponible sur les ordinateurs du L'UTÈS (Bâtiment Atrium), y compris ceux en libre service.

#### ■ B. Utiliser *Mathematica* chez soi

##### ■ 1. Sans installation: Bureau de L'UTÈS

Si vous êtes sous Windows, vous pouvez utiliser *Mathematica* via une connexion Internet sans rien installer chez vous. Il suffit pour cela d'aller sur le **bureau de L'UTÈS** : <http://lutes.upmc.fr/ts/> et suivre les instructions (aide via?) pour ouvrir votre bureau de L'UTÈS à distance.

Une fois sur le bureau de L'UTÈS (Étagère = Compagnon de L'UTÈS), il faut choisir :

PEDAGOGIE (en haut à droite),

puis MATHEMATIQUES,

puis plus bas dans la liste dans la rubrique LOGICIELS DE CALCUL on trouve MATHEMATICA.

La connexion à distance est aussi un moyen pour récupérer / travailler sur vos fichiers créés lors des TP dans les salles de L'UTÈS.

En 2017, la version 11.1.1 de *Mathematica* est installée à L'UTÈS pour les connexions, locales ou à distance, avec authentification, tandis que la version 10 est disponible sans authentification.

##### ■ 2. Avec installation chez vous sur votre ordinateur

Sur le portail Web de l'UPMC ( <https://mon.upmc.fr> ) se trouve dans la rubrique "Mes Outils" l'accès au logiciel *Mathematica* pour une installation sur n'importe quel ordinateur. L'activation par un mot de passe est liée à l'appartenance à l'UPMC : cette appartenance est vérifiée par votre e-mail (...@etu.upmc.fr) que vous devez

donnez lors de votre première inscription sur le site de Wolfram. La version actuelle en téléchargement est la version 11.3.0 qui a quelques fonctionnalités en plus des versions 10.4.1 disponibles à L'UTÈS.

### ■ C. Utiliser *Mathematica* à L'UTÈS

Pour lancer Mathematica directement dans les salles de L'UTÈS il faut faire comme ci-dessus.

Bannière d'authentification :

n° d'étudiant  
mot de passe

Une fois sur le bureau de L'UTÈS (Étagère = Compagnon de L'UTÈS), il faut choisir :

PEDAGOGIE (en haut à droite),

puis MATHEMATIQUES,

puis plus bas dans la liste dans la rubrique LOGICIELS DE CALCUL on trouve MATHEMATICA.

Sous Unix, taper dans une fenêtre "Console" : `mathematica &` et valider.

### ■ D. Organisation

cf. fichier ci-joint *PlanTP4M062Txt.pdf*

### ■ E. Modes d'évaluation des TP

Fichiers M\*Etu.nb rendus sur Moodle

## II. Débuter avec *Mathematica* (15 min, 0h30)

### ■ A. Interface utilisateur et Noyau

*Mathematica* est composé de deux parties ou programmes distincts :

l'interface avec l'utilisateur ou interface graphique ("*Front End*").

le noyau ("*Kernel*").

Les deux sont reliés par un troisième programme "*MathLink*" qui gère aussi les liaisons avec les autres langages de programmation.

#### ■ 1. L'interface avec l'utilisateur ("*Front End*")

**L'interface** gère la mise en page et les entrées/sorties. Cette interface de *Mathematica* est tout à fait remarquable. Elle permet de préparer des entrées de données ou ordres ("*inputs*") et d'afficher les sorties ("*outputs*") comme expliqués dans la section suivante. Les fichiers créés par *Mathematica* appelés notebook(s) ou cahiers permettent de résoudre des problèmes interactivement, de saisir les fonctions à exécuter, de faire faire des calculs, et de visualiser les résultats de ces calculs, de générer des graphiques, des sons et des animations, de rédiger.

C'est donc aussi un traitement de texte, qui permet d'inclure du texte parmi les calculs effectués ; ce document a d'ailleurs été rédigé avec *Mathematica*.

L'interface dépend de la machine sur laquelle on travaille, mais on retrouve les mêmes grands principes.

#### ■ 2. Le noyau ("*Kernel*")

**Le noyau** constitue le cœur du logiciel; il interprète les instructions d'entrée (écrites en langage *Mathematica*), puis calcule et retourne le résultat. Il est le même pour tous les ordinateurs PC, Mac, stations UNIX ou LINUX, etc... Il est constitué de milliers de fonctions opérantes en solo ou composées. Ce sont les *primitives*.

### ■ B. Menu Aide (Help)

*Mathematica* inclut une documentation exhaustive sous 3 formes principales :

Recherche par nom : c'est le mode dictionnaire

Recherche par rubrique : c'est le mode encyclopédique avec un classement hiérarchique par thèmes, puis rubriques, sous-rubriques, ...

Recherche par guide ou "tutorial" : ce sont environ vingt livres guides rédigés, soit environ 250 Mo.

#### ■ 1. Recherche par nom (mode dictionnaire)

→ Sélectionner un mot proche de celui cherché (en anglais), par exemple **plot**

Puis barre de menu **Help> Find Selected Function** (ou touche "F1" sous PC, ou "⌘ F" sous Mac)

Ici la première proposition est la bonne **Plot**

en double cliquant sur **Plot**, on obtient toute la documentation sur la fonction, sa définition, avec des exemples, etc ...

On peut aussi obtenir de l'aide directement dans le notebook en tapant dans une cellule d'Input : **? Plot**.

Mais pour cela il faut solliciter le noyau, ce que nous ferons dans la section III. ci-dessous.

On peut consulter l'aide de l'interface pendant qu'un calcul s'effectue parce que ce sont des programmes distincts.

## ■ 2. Recherche directe (en mode encyclopédique, *i.e.* par notions)

Il suffit de rechercher quelque chose (par exemple des fonctions pour faire des tracés de fonctions)

soit dans le **Help > Documentation Center**

soit dans le **Help > Virtual Book**

(Les deux modes ne diffèrent que par leur présentation d'affichage).

On peut alors faire une exploration hiérarchique des fonctions ou recherche par rubrique en double cliquant sur les rubriques d'intérêt.

Solutions :

## ■ 3. Et aussi : utilisation (éventuelle) des icônes

La fenêtre du DOCUMENTATION CENTER comporte trois icônes en haut à gauche

La "petite maison" est HOME le point de départ dans le documentation center (là où on se trouve) : recherche encyclopédique avec une présentation avec deux niveaux de hiérarchie.

Le F[...] le FUNCTION NAVIGATOR : recherche encyclopédique.

Similaire au Documentation Center avec plus de deux niveaux de hiérarchie. Il y a quelques différences avec le précédent dans la présentation, et aussi dans le contenu : on y trouve par exemple les add-ons.

Le "livre ouvert" est VIRTUAL BOOK : les livres guides ou "tutorials"

Attention : le système des liens permet de naviguer dans tous les sens !

## ■ C. Notebook = fichier *Mathematica* = NomDuFichier.nb

Les fichiers *Mathematica*, sont appelés "notebooks", et ont pour extension ".nb".

Les notebooks sont totalement portables (ce sont des fichiers texte ou ASCII). Par exemple, un notebook créé sur un Mac peut être envoyé par courrier électronique, réceptionné, et lu par quelqu'un qui a un PC ou sous Unix, et inversement.

Quelle que soit la plate-forme (PC, Mac, Unix), l'utilisation du fichier est la même à quelques nuances près.

Ainsi, par exemple sous Mac la touche ⌘ tient lieu (sauf exception) de la touche **CTRL**. Sur un Mac on fera le copier/coller en marquant la zone utile puis ⌘ c pour copier et ⌘ v pour coller. Avec un PC, on fera **CTRL** c puis **CTRL** v et sous Unix, on utilisera (par exemple) le bouton gauche, puis celui du milieu.

Les notebooks *Mathematica* peuvent contenir des éléments dynamiques comme des animations ou encore des Manipulate. Un notebook peut être exporté en format CDF (Computational Document Format) de Wolfram, un format qui peut être lu par tous grâce à l'installation d'un lecteur gratuit de CDF. On peut donc distribuer ses rapports écrits interactifs *Mathematica* à des personnes qui n'ont pas de licence *Mathematica*.

# III. Travailler avec *Mathematica* (1h30)

Faire cette section en même temps que les exercices de façon progressive.

## ■ A. Dans l'interface seule (45 min avec les exercices)

### ■ 1. La barre de fenêtre

C'est un outil pratique pour générer des cellules d'un type donné, mais surtout un outil indispensable pour connaître en permanence le type de la cellule dans laquelle on se trouve.

Pour faire apparaître la barre de la fenêtre, il suffit de faire dans la barre de menu :

**Window > Toolbar > Formatting**

### ■ 2. Types utilisés de cellule

L'interface permet de structurer la présentation à l'écran (et le fichier) en cellules ("cells"). Une cellule est constituée d'une ou de plusieurs lignes et est délimitée par un crochet à droite de la fenêtre.

Les cellules sont de différents types. Les types de cellules, les “contenants” que nous utiliserons sont :

Title  
 Subtitle (ne pas confondre avec Chapter)  
 Subsubtitle (ne pas confondre avec Subchapter)  
 Section  
 Subsection  
 Subsubsection

Ces cellules servent à subdiviser le contenu de la fenêtre, et surtout à le hiérarchiser.

Elles peuvent contenir des cellules “contenues” que nous utiliserons, et notamment :

Text  
 Input  
 Output

### ■ 3. Génération d’une cellule

Pour créer une nouvelle cellule : on observe que le curseur de la souris est vertical quand il passe au dessus d’une cellule, et qu’il est horizontal ailleurs. Dans cette dernière situation, il suffit alors de cliquer dans la fenêtre, vide de cellule, puis de taper sur une touche (une lettre ou la touche ↵ (“retour à la ligne” ou “retour chariot” ou “carriage return”)) pour faire apparaître une nouvelle cellule de type “Input”.

On peut changer le type d’une cellule en sélectionnant le crochet à droite de la cellule, puis en choisissant le type de la cellule dans le menu de la barre de la fenêtre, ou dans le menu : **Format ▶ Style ▶ Text**

On peut aussi créer plus directement une nouvelle cellule d’un type donné à l’aide des raccourcis clavier :

Mac	PC	Type de cellule générée
⌘ 1	Alt 1	Title
⌘ 2	Alt 2	Chapter ou Subtitle (selon les versions et les styles)
⌘ 3	Alt 3	Subchapter ou Subsubtitle (idem)
⌘ 4	Alt 4	Section
⌘ 5	Alt 5	Subsection
⌘ 6	Alt 6	Subsubsection (souvent utilisée)
⌘ 7	Alt 7	Text (la plus souvent utilisée)
⌘ 9	Alt 9	Input (peu utilisée, car c’est la cellule par défaut)

On rappelle que les cellules “Input” sont créées par défaut.

### ■ 4. Édition de texte ou d’une cellule

Pour supprimer une cellule, il suffit de la sélectionner (en cliquant sur le crochet qui la délimite), et de la supprimer avec la touche d’effacement (sur Mac) ou la touche suppr (sur PC).

Pour éditer du texte dans une cellule, les opérations habituelles fonctionnent très bien :

soit en faisant **Edit ▶ Copy** soit en faisant ⌘ c (Mac) ou CTRL c (PC).

Mac	PC	Action
⌘ x	CTRL x	couper
⌘ c	CTRL c	copier
⌘ v	CTRL v	coller
⌘ z	CTRL z	revenir en arrière (undo) : une fois seulement dans le texte uniquement

### ■ 5. Règle d’or d’écriture

Chaque cellule a son style propre, c’est-à-dire une police, une taille de caractère, etc ... Il ne faut jamais (à moins d’avoir de très, très bonnes raisons !) modifier le contenu de la cellule en modifiant ces caractéristiques.

Une raison simple est que l’on peut faire passer le contenu d’une cellule “Text” pour une cellule “Input”, et vice-versa. Bien évidemment on ne sait plus du tout ce que l’on fait.

Mais il y a bien d’autres raisons encore, notamment sur le traitement informatique du notebook par *Mathematica*. C’est même une source de plantage potentiel pour le système *Mathematica* qui finit par ne plus savoir que faire !

Au total, il est impératif de se contenter de modifier le type de la cellule !

### ■ 6. Feuille de style

La feuille de style qui est installé par défaut :

**Format ▶ Stylesheet ▶ Default**

Il est toujours possible d'adopter un autre style (de cellules) en changeant la feuille de style.

Le style **Standard Report** présente l'avantage de mieux distinguer les cellules de type Input/Output et de texte à l'écran. Mais il a l'énorme désavantage d'être coloré : à déconseiller donc.

À moins d'un usage particulier, le style **Default** est le plus pratique.

Toutefois, pour faire une présentation de type PowerPoint avec des cellules interactives, utiliser par exemple :

**Format » Stylesheet » Slideshow » Earth**

Pour créer des Démonstrations avec *Mathematica* (cf. Aide) :

**Format » Stylesheet » Wolfram » Demonstration**

## ■ 7. Saisie et édition d'expressions mathématiques : Input ou Text

Plusieurs palettes d'outils sont disponibles dans la barre de menu pour la saisie et l'édition d'expressions mathématiques.

Il est bon de se rappeler que l'on utilise :

(menu **Palettes » Basic Math Assistant**)

pour saisir et calculer des formules dans une cellule **Input**

(menu **Palettes » Writing Assistant**)

pour écrire du texte dans une cellule **Text**

Et qu'il est préférable de ne pas intervertir : on finit par planter *Mathematica*.

## ■ 8. Raccourcis clavier

Pour aller un peu plus vite, il est utile de connaître les raccourcis clavier très utiles (Mac ou PC) dans la suite pour écrire :

les lettres grecques

- **[ESC]a[ESC]** :  $\alpha$
- **[ESC]b[ESC]** :  $\beta$  et ainsi de suite pour toutes les lettres grecques minuscules  
 $\gamma \delta \epsilon \zeta \eta \theta \iota \kappa \lambda \mu \nu \xi \omicron \omega \pi \rho \sigma \tau \upsilon \phi \chi \psi$   
 ou majuscules  $\Delta, \Phi, \Gamma, \Lambda, \Omega, \Pi, \Theta, \Sigma, \Psi, \Omega$ , et notamment :
- **[ESC]p[ESC]** : le nombre  $\pi$  (équivalent à Pi)

les opérations très courantes :

- **[CTRL]6** : puissance (exemple :  $x^2$ )
- **[CTRL]2** : racine (exemple :  $\sqrt{x}$ )
- **[CTRL]/** : fraction (exemple :  $\frac{1}{2}$ )

les symboles mathématiques usuels :

- **[ESC]ee[ESC]** : le nombre d'Euler  $e$  (équivalent à E)
- **[ESC]ii[ESC]** : le nombre imaginaire  $i$  (équivalent à I)
- **[ESC]inf[ESC]** : l'infini  $\infty$  (équivalent à Infinity)
- **[ESC]dsN[ESC]** : l'ensemble  $\mathbb{N}$  (équivalent à Integers)
- **[ESC]dsR[ESC]** : l'ensemble  $\mathbb{R}$  (équivalent à Reals)
- **[ESC]->[ESC]** :  $\rightarrow$  (équivalent à une flèche graphique, mais pas à une règle  $\rightarrow$ )
- **[ESC]int[ESC]** :  $\int$  (équivalent à Integrate[])
- **[ESC]sum[ESC]** :  $\sum$  (équivalent à Sum[])
- **[ESC]el[ESC]** :  $\in$  (équivalent à Element[])

etc... pour en connaître d'autres très rapidement il suffit de passer la souris sur le symbole d'intérêt dans la palette Basic Math Assistant ou Writing Assistant.

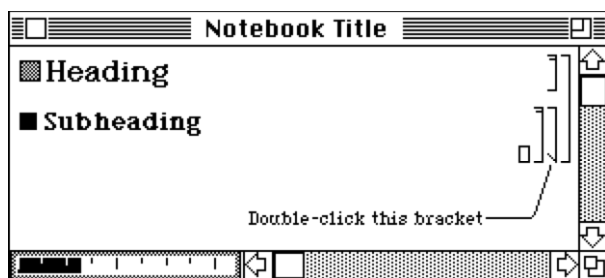
## ■ 9. Règle d'écriture

Un paragraphe par cellule.

## ■ 10. Emboîtement et masquage des cellules

On observe que ces cellules sont hiérarchisées et contiennent par défaut une ou plusieurs cellules en dessous dans la liste.

Ces cellules peuvent être fermées (masquées) ou ouvertes (démasquées) pour ne faire apparaître que les titres ou en-têtes. Pour les ouvrir, double-cliquez dans le crochet (bracket) ou dans le rectangle de la cellule, comme indiqué :



C'est très pratique et très utile pour faciliter la lecture du fichier.

## ■ 11. Sauvegarde du notebook

Pensez à sauvegarder souvent votre fichier *Mathematica* (menu **File** ► **Save as**, puis "Save" pour les sauvegardes suivantes), les mauvaises manipulations étant fréquentes... Le plus rapide étant ensuite d'utiliser le raccourci clavier :  $\text{⌘}$  s sur le Mac ou **CTRL** S sur le PC Windows.

On peut exporter un notebook en entier dans d'autres formats comme PDF, CDF ou TeX via le menu **File** ► **Save As**. (Ctrl Maj S sous PC ou  $\text{⌘}$  Maj S sous Mac) en choisissant un autre type de fichier dans la liste Type (PC) ou Format (Mac).

On peut par exemple exporter le notebook en PDF en choisissant "PDF Document (\*.pdf)".

Si on veut garder la possibilité de pouvoir interagir avec le document ou garder les éléments dynamiques (animations, Manipulate, ...), il faut choisir "Computable Document (\*.cdf)".

## ■ 12. Exercices

Vous êtes maintenant équipé pour faire les exercices afférents proposés dans le chapitre "Exercices" ci-dessous.

## ■ B. Dans l'interface avec le noyau (45 min avec les exercices)

### ■ 1. Faire exécuter un calcul

La touche  $\leftarrow$  permet de générer un saut de ligne dans une cellule, mais pas de faire des calculs comme on s'y attend généralement.

Pour faire un calcul, il faut d'abord

générer une cellule de type "Input",

écrire  $1+1$

valider par **SHIFT**  $\leftarrow$  ou bien la touche **Enter** (la touche la plus en bas à droite du pavé numérique)

**In[1]:= 1 + 1**

**Out[1]= 2**

Pour évaluer **SHIFT**  $\leftarrow$  ou la touche **Enter**. Ne pas confondre avec : aller à la ligne:  $\leftarrow$

### ■ 2. Indexation des calculs et des résultats

On note que cellule de type "Input" utilisent par défaut des caractères **gras** de type machine à écrire "courier". C'est pratiquement le seul type de cellule qui est actif, voire interactif, et qui permet d'effectuer des calculs, tracer des courbes, etc...

Le **In[ 1 ] :=** et le **Out[ 1 ] =** sont les noms des cellules. Cette numérotation est le moyen que *Mathematica* utilise pour ranger et cataloguer ses "inputs" et "outputs". *Mathematica* conserve la mémoire de tout ce que vous validez par les cellules Input. (Très important pour comprendre le fonctionnement de *Mathematica* !)

À l'inverse, le noyau de *Mathematica* ne connaît prend en compte que les cellules qui ont été directement validées, ou les cellules d'Input contenues dans des Sous-sous-sections, Sous-sections, Sections, ... qui ont été globalement validées.

### ■ 3. L'ordre d'exécution est imposé par l'utilisateur

L'ordre d'exécution est imposé par l'utilisateur (et non par l'ordre du texte)

Il est possible de revenir en arrière dans une cellule, de la modifier, et de la réévaluer. Ce qui compte c'est l'ordre d'évaluation des cellules, pas leur place dans la page, ni l'ordre d'ouverture des notebooks dans lesquels on se trouve.

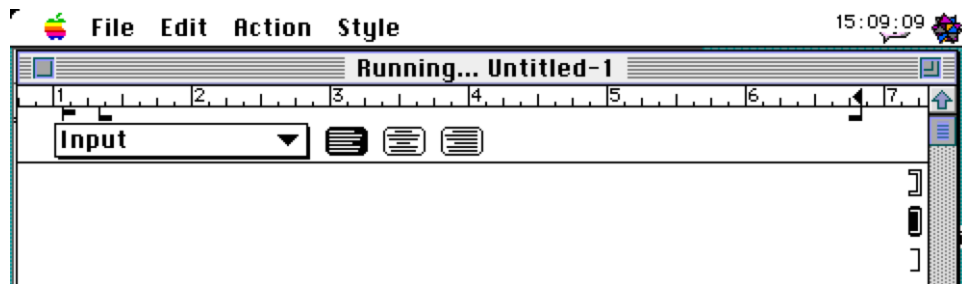
Une *session* est la suite des échanges *demande-réponse* ou *Input-Output* entre l'utilisateur et le noyau. Demande-

réponse c'est le point de vue utilisateur ; Input-Output c'est le point de vue du noyau de *Mathematica*. Ce qui rentre chez lui et ce qui en ressort. Entre les deux le noyau *évalue*. *Mathematica* garde en mémoire toutes les valeurs définies par l'utilisateur pendant une session. Elles s'évanouissent lorsqu'on quitte.

Il est possible de travailler directement avec le noyau sans l'interface. On doit alors se contenter d'un système de calcul avec un éditeur ligne à ligne sommaire.

#### ■ 4. "Running"

Lors du premier calcul, *Mathematica* initialise la mémoire du noyau, et établit la communication entre le noyau et l'interface. Si l'on observe attentivement, on verra la mention "Running" (sur la barre de titre de la fenêtre), et le doublement de l'épaisseur du crochet de la cellule : ils indiquent que le programme est en cours de calcul.



On peut donc toujours savoir globalement et localement ce que fait le noyau.

#### ■ 5. "Aborting"

La conception de *Mathematica* sous la forme de plusieurs programmes, essentiellement l'interface "*Front End*", et le noyau "*Kernel*" est très pratique car elle permet notamment d'arrêter un calcul, soit en interrompant le calcul dans le noyau ("Abort" sans quitter le noyau), soit en arrêtant (quittant ou tuant) le noyau (depuis l'interface).

Ainsi si un calcul prend trop de temps et que l'on décide d'y renoncer en cours d'évaluation, il suffit pour cela d'utiliser le menu "Evaluation", puis "Abort Evaluation". Si de plus vos instructions sont trop embrouillées, le plus simple est de repartir avec un nouveau noyau en faisant "Evaluation", puis "Quit Kernel".

Au cours des premières séances de découvertes avec *Mathematica*, il semble que les débutants aient tendance à saisir des CTRL, ALT, ALT GREC, ESC, ... intempestifs. On dit qu'ils ont les doigts gras (vocabulaire non homologué), car au lieu de saisir une bonne touche simple, ils saisissent par inadvertance des touches composées qui comportent ces touches de contrôles qui ne sont ni imprimables, ni visibles. Par définition, c'est très difficile à mettre en évidence. Par contre *Mathematica* ne sait pas interpréter ces codes qui ont l'air tout à fait normaux pour nous, mais qui n'ont aucun sens informatique. Rassurez-vous, il semble aussi que ces problèmes disparaissent assez vite au bout d'une séance environ, avec le gain d'un peu d'assurance : il est donc très important de s'entraîner un peu sérieusement à une saisie propre et soignée. Avec les caractères de contrôles, vous ne disposez pas d'un clavier, mais d'au moins 5 ou 6 claviers, ce qui requiert une certaine maîtrise.

Pour les raisons évoquées ci-dessus, il peut être (très) utile de quitter, puis de redémarrer le noyau, afin d'effacer de la mémoire tous les résultats des calculs précédents. Pour cela, on utilise le menu "Evaluation" puis "Quit Kernel". Le noyau sera alors redémarré lors de la validation du prochain calcul.

#### ■ 6. Fonctions *Mathematica*

*Mathematica* rassemble un grand nombre de fonctions testées et bien documentées. Toutes les fonctions *Mathematica* ont un nom généralement explicite **NomDeFonction**[], qui commence par une majuscule avec des arguments mis entre crochets.

Pour taper des crochets [ ] sous Mac, maintenir les touches "option" et "shift" enfoncées, et taper "(" ou ")". Sous PC, il faut (souvent) utiliser "alt grec" et taper "(" ou ")".

#### ■ 7. L'essentiel de la syntaxe *Mathematica*

Les trois éléments essentiels de la syntaxe des expressions :

- |    |           |  |
|----|-----------|--|
| () | (1+2)     | les () servent à regrouper des expressions   |
| [] | Plus[1,2] | Les fonctions <i>Mathematica</i> commencent par une majuscule et l'argument est entre crochets |
| {} | {1,2,3}   | les {} servent à définir les listes (List)   |

Autres éléments importants de la syntaxe des expressions :

- |       |              |  |
|-------|--------------|--|
| [[ ]] | {a,b,c}[[2]] | Les [[ ]] servent à adresser un élément de la liste, ici le deuxième |
| " "   | "A String"   | Les " " servent à définir les chaînes de caractères                  |
| ;     | 1+1;         | supprime les sorties de calcul                                       |

## ■ 8. Saisie d'expressions

On rappelle que la saisie d'expressions mathématiques peut être effectuée avec :

(menu **Palettes** » **Basic Math Assistant**)

pour saisir et calculer des formules dans une cellule **Input**

Des raccourcis clavier peuvent être indiqués dans les cas les plus utilisés.

## ■ 9. Recherche par nom (mode dictionnaire) dans le noyau

On peut aussi obtenir une aide succincte dans le notebook en tapant dans une cellule d'Input, et en validant avec la touche **Enter** du pavé numérique, ou bien valider par la combinaison de touches **SHIFT** **↵** du clavier.

**? Plot**

`Plot[f, {x, xmin, xmax}]` generates a plot of  $f$  as a function of  $x$  from  $x_{min}$  to  $x_{max}$ .

`Plot[{f1, f2, ...}, {x, xmin, xmax}]` plots several functions  $f_i$ . >>

On observe que `Plot[]` requiert au moins deux arguments pour fonctionner :

une fonction (ou une liste de fonctions) de  $x$  à tracer

un domaine de calcul de la fonction, qui sert aussi de domaine de tracer par défaut.

`PlotRange` est probablement l'une des options les plus importantes de `Plot`, car elle focalise le tracé sur une région clairement spécifiée du tracé.

La liste de plus de 100 Options de `Plot` est obtenue avec `Options[Plot]`, ou aussi en faisant :

**?? Plot**

`Plot[f, {x, xmin, xmax}]` generates a plot of  $f$  as a function of  $x$  from  $x_{min}$  to  $x_{max}$ .

`Plot[{f1, f2, ...}, {x, xmin, xmax}]` plots several functions  $f_i$ .

`Plot[{..., w[fi], ...}, ...]` plots  $f_i$  with features defined by the symbolic wrapper  $w$ .

`Plot[..., {x} ∈ reg]` takes the variable  $x$  to be in the geometric region  $reg$ . >>

`Attributes[Plot] = {HoldAll, Protected, ReadProtected}`

`Options[Plot] = {AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , Axes → True, AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {}, Background → None, BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic, ContentSelectable → Automatic, CoordinatesToolOptions → Automatic, DisplayFunction →  $\$DisplayFunction$ , Epilog → {}, Evaluated → Automatic, EvaluationMonitor → None, Exclusions → Automatic, ExclusionsStyle → None, Filling → None, FillingStyle → Automatic, FormatType → TraditionalForm, Frame → False, FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {}, ImageMargins → 0., ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic, LabelStyle → {}, MaxRecursion → Automatic, Mesh → None, MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic, Method → Automatic, PerformanceGoal →  $\$PerformanceGoal$ , PerformanceGoal →  $\$PerformanceGoal$ , PlotLabel → None, PlotLabels → None, PlotLegends → None, PlotPoints → Automatic, PlotRange → {Full, Automatic}, PlotRangeClipping → True, PlotRangePadding → Automatic, PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme →  $\$PlotTheme$ , PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &), RotateLabel → True, ScalingFunctions → None, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}, WorkingPrecision → MachinePrecision}`

`Plot[]` est donc une fonction qui demande une certaine attention.



**■ 10. Règle d'or d'écriture**

Une demande par cellule.

**■ 11. Sauvegarde du notebook**

Pensez à sauvegarder souvent votre fichier *Mathematica*.

*Certes, on l'a déjà dit, mais on ne le dit jamais assez.*

**■ 12. Exercices**

Vous êtes maintenant équipé pour faire les exercices afférents proposés dans le chapitre “Exercices” ci-dessous.

# Exercices

À faire avec le fichier joint *M1\_4M062IntroductionEtu.nb*

## I. Interface

### ■ A. Mise en forme

#### ■ → 1. Les cellules et leur hiérarchie

Reproduire avec l'Interface de *Mathematica* la présentation donnée dans le fichier : *M1MiseEnFormeIEtu.pdf*

### ■ B. Saisie d'expressions

#### ■ → 1. Avec la palette Texte (Writing Assistant)

Dans une cellule "Texte" saisir les formules :

$$\sum_{i=1}^{10} i, \quad \sum_{i=1}^n i, \quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

(NB : utiliser Tab pour passer d'une case à l'autre)

#### ■ → 2. Avec la palette Input (Basic Math Assistant)

Dans une cellule "Input" saisir les formules :

$$\sum_{i=1}^{10} i$$

$$\sum_{i=1}^n i$$

#### ■ → 3. Avec la palette et les raccourcis clavier

→ a. Dans une cellule "Texte" saisir les formules :

$$\sum_{i=1}^{10} i^2, \quad \sum_{i=1}^n i^2, \quad \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

NB : pour les matrices, on peut aussi utiliser **Insert** ► **Table/Matrix** ► **New** ► ... pour contrôler tous les formats de tables ou de matrices.

→ b. Dans une cellule "Input" saisir la formule :

$$(x^5 + 4x^3 + 2) \exp[-(y^2)/2] / \sqrt{\pi i}$$

sous une forme plus lisible

$$\frac{(x^5 + 4x^3 + 2) e^{-\frac{y^2}{2}}}{\sqrt{\pi}}$$

## II. Un tour introductif

Voici quelques calculs que nous soumettons à votre sagacité.

Pour chacune des cellules, nous vous proposons de faire ces démarches successivement :

1. Bien observer, et Anticiper le résultat attendu : ce qui pourrait / devrait se passer,
2. Effectuer le calcul dans une fenêtre séparée (un autre notebook vierge)  
en faisant un copier-coller de la cellule intérieure
3. Expliquer le résultat du calcul obtenu (LE PLUS IMPORTANT)
4. Comparer avec la solution cachée et le commentaire en double-cliquant

Cf. la suite du notebook *M1IntroductionMMaTxt.nb* qui doit être rendu pour que nous puissions suivre votre progression (ce que vous avez compris ou non suivant vos explications).

## ■ A. Opérations de base (symboliques avec précision arbitraire)

Bonjour !

### ■ 1. Élémentaire

$1 + 1$

$4a + 5a$

$\text{Cos}[\text{Pi}]$

$\text{Cos}[x] /. x \rightarrow \pi$

$3x + 2x$

### ■ 2. Les trois opérations

$31 \times 5$

$315$

$14$

$3^{100}$

Solutions à retenir

### ■ 3. Valeurs numériques approchées $N[]$

Il faut donc retenir qu'*a priori*, *Mathematica* calcule toujours de façon symbolique et utilise les règles de l'algèbre pour simplifier et obtenir un résultat exact. *Mathematica* est d'abord un système symbolique, et c'est aussi principalement un langage de programmation fonctionnel.

Pour obtenir une valeur numérique, on peut utiliser la fonction  $N[]$  qui convertit le calcul sous forme Numérique ordinaire. Cette fonction  $N[]$ , se présente, comme toutes les fonctions *Mathematica*, sous la forme d'un nom commençant par une majuscule avec des arguments mis entre crochets. Elle est très (très) utilisée, d'où son nom (exceptionnellement) très court.

Pour corriger une erreur ou modifier un calcul dans une cellule, cliquer au point d'intervention et modifier en conséquence, puis validez.

$N\left[\frac{315}{14}\right]$

$N[3^{100}]$

$N[3^{100}, 20]$

$3^{100} // \text{Timing}$

$3^{100}; // \text{Timing}$

### ■ 4. Quelques tests utiles à partir de calcul de puissances pour comprendre

Ces types de tests sont très importants pour utiliser *Mathematica* à bon escient :

$3^{100}; // \text{AbsoluteTiming}$

$3^{1000}; // \text{AbsoluteTiming}$

$3^{10000}; // \text{AbsoluteTiming}$

$3^{100000}; // \text{AbsoluteTiming}$

$3^{1000000}; // \text{AbsoluteTiming}$

$3^{10000000}; // \text{AbsoluteTiming}$

$3^{100000000}; // \text{AbsoluteTiming}$

$3^{1000000000}; // \text{AbsoluteTiming}$

$3.^{1000000000} // \text{AbsoluteTiming}$

$\{0.001013, 2.9646009515484 \times 10^{47.712125}\}$

Il est possible de forcer le calcul numérique, en fixant la précision numérique d'un nombre flottant, ici 3..

Le calcul n'est plus une multiplication en nombres entiers de 100000000 de termes. Il devient un simple exercice de conversion en passant par le logarithme d'un nombre réel.

```
N[3100 000 000]; // AbsoluteTiming
```

```
N[3.100 000 000] // AbsoluteTiming
```

Conclusions intermédiaires :

*Mathematica* est un système symbolique qui calcule *a priori* en nombres entiers avec une précision arbitraire.

On peut contrôler cette précision en utilisant des nombres flottants, ou décimaux (et non N[]).

On peut contrôler l'affichage avec N[] qui garantit aussi la précision du calcul effectué en n'affichant jamais plus que la précision du calcul.

## ■ 5. Exemples récapitulatifs

```
N[ $\pi$ ]
```

```
 $\pi$  // N
```

```
N[ $\pi$ , 30]
```

```
N[ $\pi$ , 2000]
```

## ■ B. Opérations de base (numériques)

Les calculs numériques avec *Mathematica* sont posés et effectués de façon habituelle. Voici un exemple avec une soustraction (-), une multiplication (\*) et une division (/) sur des nombres réels (flottants) affectés d'une précision bien définie :

```
(10.2 - 3.1) * 6.2 / 2.9
```

On peut aussi faire des calculs avec des nombres complexes, le nombre imaginaire *i* étant écrit "I" ici :

```
(2 + 3 I) * (4 - 5 I)
```

On remarque que "\*" sert, comme très souvent en informatique, de signe "multiplier".

On peut utiliser aussi le caractère "espace" (ou le signe "x" à ne pas confondre avec x, ou le produit vectoriel "x").

## ■ C. Nombres et Fonctions élémentaires

### ■ 1. Quelques constantes mathématiques

*Mathematica* dispose de quelques constantes prédéfinies que l'on peut toujours écrire avec un **Nom** commençant par une majuscule et des lettres, ou bien un symbole comme expliqué plus haut.

`==` est une notation de type Infix pour la fonction test d'égalité `Equal[]` entre deux arguments. Tous les tests d'égalité ci-dessous renvoient `True` quand ils sont validés.

- le nombre  $\pi$

```
Pi ==  $\pi$ 
```

- le nombre *e*

```
E == e
```

- le nombre imaginaire *i*

```
I == i
```

- l'infini  $\infty$  :

```
Infinity ==  $\infty$ 
```

### ■ 2. Quelques fonctions mathématiques

*Mathematica* dispose de nombreuses fonctions prédéfinies dont le nom suit la règle **Nom[]** :

- racine carrée :  $\sqrt{x}$

```
Sqrt[x] ==  $\sqrt{x}$ 
```

- exponentielle,  $e^x$ , qui peut être notée dans *Mathematica* de 5 façons toutes équivalentes :

```
Exp[x] == E ^ x == Ex == e ^ x == ex
```

- logarithme népérien :  $\ln(x)$

**Log[x]**

- logarithme en base  $b$  :  $\log_b(x)$  (le plus souvent,  $b = 10$ )

**Log[b, x]**

- fonctions trigonométriques :

**Sin[x]**

**Cos[x]**

**Tan[x]**

- fonctions trigonométriques inverses :

**ArcSin[x]**

**ArcCos[x]**

**ArcTan[x]**

- valeur absolue :  $|x|$

**Abs[x]**

- factorielle :  $n!$

**n! == Factorial[n]**

Sélectionnez la fonction **Cos[]** et tapez sur F1 (sous PC) ou ⌘ F (sous Mac) pour afficher la page d'aide de cette fonction.

Explorez d'autres fonctions avec la rubrique SEE ALSO "ref/ArcCos" (en bas de la page)

Revenez à la page d'aide précédente, et explorez TUTORIALS "tutorial/SomeMathematicalFunctions" qui donne une petite liste des fonctions élémentaires.

Revenez à la page d'aide précédente, et explorez MORE ABOUT puis Elementary Functions, "guide/SomeMathematicalFunctions" qui donne la liste des fonctions élémentaires.

## ■ D. Fonctions

### ■ 1. Programmation procédurale et fonctionnelle

Le même mot "fonction" est utilisé en programmation procédurale, et en programmation fonctionnelle. Or ce mot n'a pas la même signification, car il est utilisé dans des contextes très différents.

La programmation procédurale ou impérative repose sur le modèle des machines à états comme le modèle de la machine de Turing (1936). On s'intéresse à l'état de la mémoire (centrale et différents registres éventuellement) et aux instructions qui modifient son état par des affectations successives. Les mots clés ici sont "état de la mémoire", "instructions", et "affectations successives". Les procédures, encore appelées routines, ou sous-routines, ou fonctions servent donc à modifier l'état de la mémoire au travers d'étapes à réaliser.

La programmation fonctionnelle repose sur l'évaluation de fonctions mathématiques dans un système formel comme dans le lambda-calcul de Church (1930). Les langages fonctionnels les plus anciens sont Lisp (1958), Scheme (1975), et Common Lisp (1984). Par principe, on ne s'intéresse pas à la mémoire, ni à la notion d'état, et aux affectations. Un programme est donc simplement une application, au sens mathématique.

Chaque mode de programmation a un langage particulier, et des méthodes de travail différentes :

Dans le premier cas (procédural), le programmeur doit déclarer tous les objets dans la mémoire pour les manipuler par l'intermédiaire du compilateur.

Dans le second cas (fonctionnel), tous les objets seront déclarés et indexés automatiquement par le programme en temps utile. Le programmeur doit donc se concentrer principalement sur les opérations de réécriture des expressions. Ce travail sur les symboles est souvent plus intuitif et plus naturel car plus proche d'un langage mathématique.

Remarque (un peu spécialisée) sur le typage des variables et des fonctions :

Bien souvent, en tous cas, au niveau débutant, le programmeur ne déclare pas de variables dans *Mathematica*, et il pourrait finir par penser qu'il n'y a pas de typage des variables. En réalité, tout est typé, car tout est fonction, et donc tout est encapsulé dans quelque chose. Au niveau le plus bas, *i.e.* au niveau le plus interne on retrouve les

feuilles habituelles : les **Integer**, **Real**, **Character**, **String**, ...

Dans les programmes plus professionnels, tout est normalement typé pour éviter tout abus de langage.

Beaucoup de langages plus récents sont multiparadigmes : procéduraux, fonctionnels, et orientés objet comme Caml (1987), Python (1990), OCaml (1996).

*Mathematica* est multiparadigme et l'on peut programmer de façon procédurale ou fonctionnelle, ... selon les besoins. Il n'en demeure pas moins que *Mathematica* est d'abord un système symbolique et fonctionnel.

## ■ 2. Mode d'utilisation d'une fonction

En dehors de la notation fonctionnelle standard **Nom[argument1, argument2, ...]**, il y a trois façons d'utiliser ou de noter les fonctions :

**Prefix**

**Infix**

**Postfix**

Exemple : **Sqrt[2]** est la notation fonctionnelle standard de l'extraction d'une racine carrée.

Le nom de la fonction "**Sqrt**" est devant l'argument : **Sqrt@2** est la notation dite **Prefix**

$$\text{Sqrt}[2] == \text{Sqrt}@2 == \text{Sqrt}@ (2) == \sqrt{2}$$

Exemple : **Plus[2, 3]** est la notation fonctionnelle standard de l'addition. Le nom de l'opération ou de la fonction "**Plus**" est devant les arguments 2, et 3, mis en séquence.

**Plus[2, 3]**

Traditionnellement, on utilise la notation de type **Infix** pour l'addition "+":

$$\text{Plus}[2, 3] == 2 \sim \text{Plus} \sim 3 == (2 + 3)$$

Il est aussi possible de mettre les arguments devant le nom de la fonction en utilisant la notation **Postfix** avec **//** :

$$N[\sqrt{2}] == \sqrt{2} // N$$

$$2 // \text{Sqrt} == \sqrt{2}$$

En résumé :

Notation fonctionnelle	fonction f	Sqrt	Plus	Traditionnel
Standard	f[x]	Sqrt[2]	Plus[2, 3]	
Prefix	f @ x	Sqrt @ 2		$\sqrt{2}$
Infix	x ~ f ~ y		2 ~ Plus ~ 3	2 + 3
PostFix	x // f	2 // Sqrt		

## ■ 3. Composition de fonctions, et écriture avec indentation automatique

C'est le mode le plus simple de programmation fonctionnelle. Exemple :

**Exp[Cos[x]]**

Saisir, et constater que la hiérarchie d'indentation est automatiquement respectée à chaque retour à la ligne :

```
Plot[
  Cos[x],
  {x, -2 π, 2 π}
]
```

Écrire des fonctions composées en se servant de l'indentation automatique confère une meilleure écriture et une meilleure lecture.

Mais on peut écrire comme on le souhaite, l'exemple ci-dessous ne nécessite pas d'indentation :

```
Plot[Cos[x], {x, -2 π, 2 π}]
```

On peut faire autant de compositions que souhaitées :

```
Plot[
  Exp[Cos[ArcSin[ $\sqrt{x}$ ]]],
  {x, -2  $\pi$ , 1}
]
```

**Manipulate[]** est utilisé pour faire une animation interactive d'un graphique :

```
Manipulate[
  Plot[
    Cos[a x + b],
    {x, -2  $\pi$ , 2  $\pi$ }
  ],
  {a, 1, 10},
  {b, 0, 2  $\pi$ }
]
```

D'une façon générale, on gagne beaucoup de temps à écrire ses fonctions en respectant la hiérarchie de la composition de fonctions. Cela permet généralement de comprendre d'emblée ce qui est fait, et de repérer très rapidement les problèmes d'écriture.

#### ■ 4. Arguments et Options des Fonctions : tracer de courbes

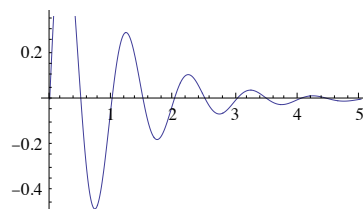
Les fonctions ont généralement des arguments obligatoires, et un certain nombre d'Options que l'on peut ou non rajouter pour préciser la fonction. Étudions la fonction très utile **Plot[]** au travers des exemples suivants :

Plot comporte deux arguments obligatoires :

la fonction à tracer,  $e^{-x} \sin[2 \pi x]$

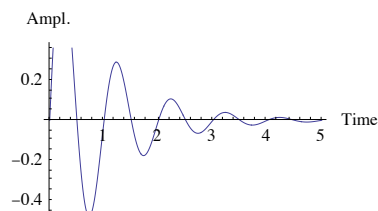
et la spécification de l'intervalle sur lequel on cherche à tracer la fonction,  $\{x, 0, 5\}$

```
Plot[ $e^{-x} \sin[2 \pi x]$ , {x, 0, 5}]
```



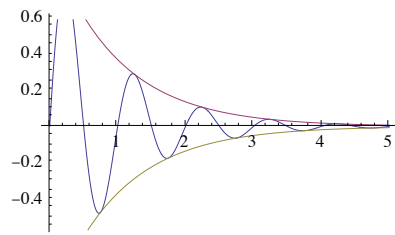
On peut rajouter en option `AxesLabel`  $\rightarrow \{\text{"Time"}, \text{"Ampl."}\}$ , pour spécifier les axes. Elle consiste à remplir avec les contenus donnés dans la liste  $\{\text{"Time"}, \text{"Ampl."}\}$  la variable `AxesLabel` :

```
Plot[
   $e^{-x} \sin[2 \pi x]$ ,
  {x, 0, 5},
  AxesLabel  $\rightarrow \{\text{"Time"}, \text{"Ampl."}\}$ 
]
```



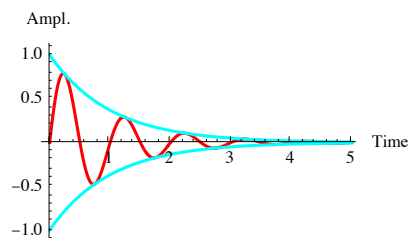
Tracé de plusieurs fonctions sur le même graphe au moyen d'une liste de fonctions à tracer ...

```
Plot[{e^-x Sin[2 π x], Exp[-x], -Exp[-x]}, {x, 0, 5}]
```



... de l'utilité des couleurs, et et des labels, et du cadrage de toute la fonction, ...

```
Plot[{
  e^-x Sin[2 π x],
  Exp[-x],
  -Exp[-x]
},
{x, 0, 5},
PlotStyle -> {RGBColor[1, 0, 0], Hue[0.5`], Hue[0.5`]},
AxesLabel -> {"Time", "Ampl."},
PlotRange -> All
]
```



## ■ 5. Définition de nouvelle fonction par l'utilisateur

Pour définir la fonction  $f(x) = \frac{ax}{b+x}$ , il faut écrire la définition ci-dessous, et la valider pour qu'elle soit connue du noyau :

```
f[x_] :=  $\frac{a x}{b + x}$ 
```

On vérifie que  $f$  est correctement définie, en interrogeant le noyau :

```
? f
```

```
Global`f
```

```
f[x_] :=  $\frac{a x}{b + x}$ 
```

Cette fonction est utilisable avec n'importe quel argument, valeur ou expression :

(notez que **titi** et **grosminet** sont de couleur bleue car ces variables ne sont pas définies et sont inconnues du système)

```
f[titi]
```

```
a titi
```

```
b + titi
```

```
f[grosminet]
```

```
a grosminet
```

```
b + grosminet
```



$$f[\sqrt{5}]$$

$$\frac{\sqrt{5} a}{\sqrt{5} + b}$$

$$f[y + \sqrt{2} z + 0.3]$$

$$\frac{a (0.3 + y + \sqrt{2} z)}{0.3 + b + y + \sqrt{2} z}$$

Nous avons utilisé “:=” qui est le symbole Infix pour une “affectation retardée”. Cela signifie que l’expression toute entière “ $f[x_] := \frac{a x}{b+x}$ ” est mise en mémoire dans le noyau, et chaque fois que l’utilisateur invoquera cette fonction avec un argument quelconque, la fonction sera évaluée comme on l’a vu plus haut.

On peut donc (très correctement) voir cette expression toute entière comme une règle de remplacement dans un traitement de texte : chaque fois que le noyau rencontrera  $f[\text{quelqueChose}]$ , il remplacera  $f[\text{quelqueChose}]$  par la partie droite de l’expression  $\frac{a x}{b+x}$ .

Il est possible d’utiliser “=” en lieu et place de “:=”. Dans ce cas  $f[x]$  prend la valeur de la partie droite de l’expression. On remarque que si  $x$  est affectée d’une valeur ( $x = 4$ ),  $f[x]$  prend la valeur  $\frac{4a}{b+4}$  une fois pour toutes. Sauf si l’on recherche spécifiquement ce type de comportement, et en cas de doute, il vaut généralement mieux utiliser “:=”.

Il est possible d’omettre “\_” dans “ $x_$ ”. Dans ce cas  $f[x]$  n’est définie que pour  $x$ , ce qui n’est généralement pas souhaitable, ni utile.

Une fonction peut avoir plus variables :

$$g[x_, y_] := \frac{a x}{b+x} + \frac{a y^4}{b+y^4}$$

? g

Global`g

$$g[x_, y_] := \frac{a x}{b+x} + \frac{a y^4}{b+y^4}$$

Pour que la définition soit générale, *i.e.* qu’elle s’applique pour n’importe quel nom d’argument, les arguments  $x$ ,  $y$ , ... doivent être des variables muettes, qui ne servent qu’à définir le membre de droite dans l’expression. C’est du tiret bas “\_” (souligné ou “underscore” en anglais).

On peut effacer tout ce qui se rattache à la définition de  $g$  :

ClearAll[g]

? g

Global`g

## ■ 6. Éléments d’étude de fonction

a/ Calcul de la dérivée première de  $f$  en utilisant la notation traditionnelle de type Infix :

$$f'[s]$$

La notation fonctionnelle de la dérivation de  $f$  par rapport à  $x$  est :  $D[f[x], x]$

On peut facilement calculer la dérivée cinquième ... (cf. documentation dictionnaire)

$$D[f[x], \{x, 5\}] = f'''''[x]$$

$$D[f[x], \{x, 5\}]$$

True

$$-\frac{120 a x}{(b+x)^6} + \frac{120 a}{(b+x)^5}$$

**b./** Calcul des asymptotes :

`Limit[f[x], x → ∞]`

`Limit[f[x], x → -b]`

**c./** Calcul de valeurs particulières :

`Limit[f'[x], x → 0]`

**d./** Calcul de valeurs particulières par résolution d'équations :

`Solve[f[s] == a / 2, s]`

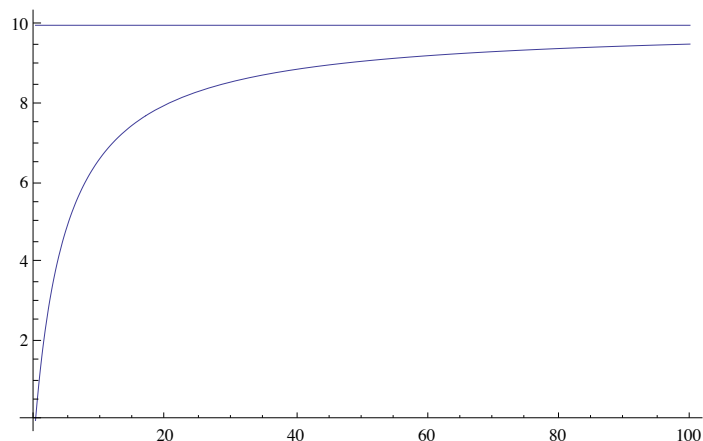
**e./** Tracé de courbes :

```
Plot[
  {f[x], 10} /. {a → 10.0, b → 5.0},
  {x, -60, 60},
  PlotRange -> {-60, 60}
]
```

Que remarque-t-on ?

En pratique, cette fonction n'est intéressante que pour des valeurs réelles. On résume les résultats dans le graphique :

```
Plot[{f[x], 10} /. {a → 10.0, b → 5.0}, {x, 0, 100}, PlotRange -> All]
```



*f* conserve sa définition symbolique (pas d'effets de bord) :

`f[s]`

## ■ E. Graphiques 2D, 3D, sons et animations

### ■ 1. Représentation graphique

*Mathematica* permet de représenter facilement n'importe quelle expression sous forme graphique 2D, 3D, 3D animé, et même sous forme sonore. L'un des aspects très pratiques de ces tracés est la possibilité de pouvoir modifier tous les paramètres du graphique très facilement ainsi que l'échelle d'observation. La visualisation permet dans certains cas de comprendre des notions difficiles d'abord. Par exemple avec le tracé suivant, on observe immédiatement que la fonction `Sinus[10000/t]` présente une curiosité mathématique au voisinage de 0. Pouvez-vous formuler laquelle?

```
Plot[Sin[ $\frac{10000}{t}$ ], {t, 0, 4000}]
```

Mais on arrive toujours à mettre le système en difficulté (Expliquer pourquoi ?):

```
Plot[Sin[ $\frac{10000}{t}$ ], {t, 0, 40}]
```

## ■ 2. Représentation sonore

On peut d'ailleurs l'écouter, pour se convaincre de la pathologie de la fonction en ce point. On remarquera que les singularités du type  $1/0$  ne sont pas un obstacle, ce qui est très appréciable.

```
Play[Sin[10 000 / t], {t, -2, -0.001}]
```

## ■ 3. Tracé de fonction en 3 dimensions - Animation 3D

Tracé en 3D d'une fonction simple. On trouve ce type de courbe pour décrire la surface d'un cristal.

```
Plot3D[Sin[x] Sin[y], {x, -5, 5}, {y, -5, 5}, PlotPoints -> 30]
```

```
Manipulate[
```

```
Plot3D[Sin[a x + b] Sin[y], {x, -5, 5}, {y, -5, 5}, PlotPoints -> 30],
```

```
{a, 1, 5},
```

```
{b, 0,  $\pi$ }
```

```
]
```

# I. Suppléments

## ■ A. Langage naturel

*Mathematica* et Wolfram Alpha ([www.wolframalpha.com](http://www.wolframalpha.com)) peuvent aussi être utilisés en langage naturel, c'est-à-dire à partir de mots anglais. Si on cherche une fonction dont on ne connaît pas très bien le nom, et que l'on ne veut pas recourir à l'aide, on peut :

### ■ 1. Complétion automatique avec $\mathbb{K}$

Commencez à saisir le mot, et cherchez tous les mots commençant ainsi :

```
pl (* puis  $\mathbb{K}$  *)
```

```
Pl (* puis  $\mathbb{K}$  *)
```

permet de retrouver le nom de la fonction **Plot**.

### ■ 2. = Wolfram Alpha

Saisissez (en anglais) ce que vous cherchez après un signe = placé au tout début de la ligne (Input).

Exemple :

```
= Plot sinus from 0 to 10
```

Avec un peu de chance Wolfram Alpha comprend votre demande et donne non seulement la solution, mais aussi la traduction de votre demande en syntaxe *Mathematica*.



**Plot sinus from 0 to 10**

## ■ B. Step-by-step

Wolfram Alpha Pro (la version payante de Wolfram Alpha) a introduit une nouvelle fonctionnalité, le *step-by-step* :

<http://www.wolframalpha.com/pro/step-by-step-math-solver.html>

Le *step-by-step* donne les étapes intermédiaires de la résolution mathématique pour de nombreux problèmes, comme par exemple :

◁ Résolution d'un système d'équations

◁ Calcul d'une dérivée ou d'une primitive

◁ Transformations algébriques et trigonométriques (moins évident, à tester)

Ces étapes intermédiaires sont fournies par Wolfram Alpha qui est appelé depuis *Mathematica* dans un notebook, sans abonnement à la version Pro de Wolfram Alpha. Pour cela on peut faire appel à WolframAlpha avec une petite fonction **ShowSteps** qui filtre les résultats *step-by-step* :

```
ShowSteps[exp_] :=
  WolframAlpha[ToString@HoldForm@InputForm@exp, TimeConstraint -> Infinity,
    IncludePods -> {"Input", "Indefinite Integral", "Root", "Result"},
    PodStates -> {"Step-by-step solution", "Show all steps"}]
```

```
SetAttributes[ShowSteps, HoldAll]
```

Ceci est une version améliorée de la fonction ShowSteps obtenue sur le site :

<http://mathematica.stackexchange.com/questions/148/get-a-step-by-step-evaluation-in-mathematica>

Il suffit de :

soit copier-coller cette fonction dans son notebook, puis de l'évaluer,  
soit l'importer avec :

```
<< showsteps.m
```

Le fichier **showsteps.m** est généré automatiquement en validant la totalité du fichier *M1showsteps.nb* qui est donné sur SAKAI après l'avoir recopié dans le dossier où se trouve ce notebook.

On peut vérifier que la fonction **ShowSteps** est bien lu dans le noyau avec la commande :

```
? ShowSteps
```

Puis en la testant avec par exemple :

```
D[Sin[x^2], x] // ShowSteps
```

---

## I. Annexes

### ■ A. Wolfram et autour

Voir sur le site dans le répertoire "Livres" :

EIWLStephenWolfram-ElementaryIntroductionWolframLanguage(2015).pdf  
un-langage-pour-ingenieurThierryVerdel2102.pdf

### ■ 1. Les Add-ons

Des bibliothèques supplémentaires ou *Add-ons* sont installées dans la distribution de *Mathematica* et accroissent les potentialités du noyau. Elles ne sont pas normalement chargées dans la mémoire, ni accessibles directement car elles sont habituellement réservées à un usage spécialisé. Pour charger cette bibliothèque de fonctions ou "Package" il faut charger le fichier correspondant comme ceci :

```
<< StatisticalPlots`
```

Dans le cas présent il sert à utiliser des fonctions de tracés statistiques plus sophistiqués... (hors programme !!!)

### ■ 2. Wolfram Demonstrations

Wolfram Demo regroupe plus de 10000 démos interactives Mathematica pour expérimenter avec Manipulate une large collection de sujets, et pour mieux comprendre un sujet de mathématique, physique, chimie, biologie, etc...

Elles mettent en valeur les fonctionnalités de Mathematica, même si la création d'une telle démo est hors programme de cette UE.

### ■ 3. Wolfram Alpha

Wolfram Alpha ([www.wolframalpha.com](http://www.wolframalpha.com)) est un outil en ligne gratuit capable de calculer une réponse à une question (qui doit être posée en anglais). Il utilise une grande base de donnée et *Mathematica*.

Il permet d'accéder avec un simple navigateur internet à beaucoup des fonctionnalités de *Mathematica*.

Par exemple, la demande "derivative of sin(x)" (dérivée de sin(x)) calcule la réponse "cos(x)" et donne au passage d'autres informations utiles (propriétés de la fonction, courbes, développements limités, etc...).

À essayer !.

Wolfram Alpha peut être interrogé directement depuis *Mathematica* avec une connexion Internet.

Dans un nouveau notebook (Ctrl + N) (voir en bas), taper :

```
== GDP France
```

Si un message d'erreur apparaît c'est que la connexion Internet n'est pas bien configurée.

Sur les postes de L'UTÈS ou le bureau à distance de L'UTÈS il faut configurer *Mathematica* pour qu'il puisse accéder à Internet. Dans *Mathematica* aller dans le menu (sous PC) **Edit > Preferences** ou (sous Mac) **Mathematica > Preferences** choisir l'onglet **Internet Connectivity** choisir l'onglet **Proxy Setting** puis cocher la case **Proxy Settings** et entrer **proxyweb.upmc.fr** dans la case **HTTP Proxy** et 3128 dans la case **Port**. Tester la connexion Internet avec le bouton **Test Internet Connectivity**.

#### ■ 4. Wolfram Education Portal

Sur le Wolfram Education Portal : <https://education.wolfram.com/> qui est assez récent, on trouve des cours de math et physique niveau lycée et supérieur. Ces cours sont enrichis avec des applications ("Widgets") ou démos avec *Mathematica*. À essayer et surveiller !.

Wolfram a fondé cette initiative pour promouvoir l'apprentissage des maths avec l'ordinateur et *Mathematica* dès l'école : <http://www.computerbasedmath.org/>

#### ■ 5. Livres numériques disponibles à l'UPMC

Le point d'entrée pour chercher dans les bibliothèques papier ou numérique est Jubil : "<http://www.jubil.upmc.fr>" En haut à droite sur la page d'accueil on peut directement chercher un livre papier ou numérique "Rechercher dans le catalogue". Alternativement si on cherche uniquement des livres ou revues numériques on peut choisir à gauche "Ressources en ligne>Les livres numériques", puis la recherche "2. dans la liste A-Z des e-books" est plus adaptée si on connaît déjà le titre.

Hazrat : "*Mathematica : A Problem-Centered Approach*". Une introduction à *Mathematica* avec un point de vue de mathématique autour des nombres.

Stan Wagon : "*Mathematica in action*". Pour explorer les capacités de *Mathematica*.

#### ■ 6. Livres imprimés disponibles à l'UPMC

Patrick T. Tam : "*A Physicists guide to Mathematica*". Une bonne référence très complète pour les physiciens.

Marvin L. De Jong : "*Mathematica for calculus-based physics*". Une introduction courte à *Mathematica* très accessible est donné dans ce livre, qui présente des calculs numériques de trajectoire de systèmes simples, et plus complexes jusqu'au chaos.

Abell et Braselton : "*Mathematica by example*". Un livre un peu ancien, mais avec beaucoup d'exemples et d'exercices.

Jean-Pierre Xémard : "*Mathematica théorie et pratique*" en 4 tomes (Applications en Arithmétique, Algèbre Analyse, et Géométrie) Une série de livres en français, qui utilise *Mathematica* de manière approfondie pour faire des maths assez poussées (niveau prépa).

#### ■ B. Liens vers d'autres tutoriels/cours à l'UPMC

En Chimie (LC209) :

<http://www.lct.jussieu.fr/pagesperso/toulouse/enseignement/> <http://www.lct.jussieu.fr/pagesperso/reinh/enseignement/lc209/enonces.html>

En Physique (LP206 et LP207) : <http://www.phys.ens.fr/~jacobsen/LP206/> [http://www.lpthe.jussieu.fr/~zuber/Z\\_Notes.html](http://www.lpthe.jussieu.fr/~zuber/Z_Notes.html)

En Biologie (LV204) :

Disponible à L'UTÈS uniquement : Une fois sur le bureau de L'UTÈS, il faut choisir PEDAGOGIE (en haut à droite), puis BIOLOGIE, puis dans la liste dans la rubrique ENTRAÎNEMENT ET APPROFONDISSEMENT on trouve LV204. Un double clic sur LV204 produit un dossier LV204 sur votre bureau de L'UTÈS.

Ce dossier contient les solutions détaillées des TD de Maths de LV204 en format notebook Mathematica ou PDF.

#### ■ C. Quelques autres outils et logiciels

Un autre logiciel de calcul formel et numérique très utilisé est *Maple*. Il a des capacités formelles comparables à celles de *Mathematica*. Il est également installé sur les ordinateurs du L'UTÈS.

*Matlab* est un logiciel de calcul numérique très utilisé, mais ne permet pas de faire du calcul formel. C'est la raison pour laquelle Maple et Matlab sont souvent associés. *Scilab* est un logiciel libre alternatif à *Matlab*.

*Maxima* est un des logiciels libres et gratuits qui constituent une alternative aux logiciels commerciaux.

# M1 : Table / Contents

<b>Préambule</b>	1
I. Introduction (15 min)	1
A. Description générale de <i>Mathematica</i>	1
B. Utiliser <i>Mathematica</i> chez soi	1
C. Utiliser <i>Mathematica</i> à L'UTÈS	2
D. Plan de l'UE	2
E. Modes d'évaluation de l'UE	2
II. Débuter avec <i>Mathematica</i> (15 min, 0h30)	2
A. Interface utilisateur et Noyau	2
B. Menu Aide (Help)	2
C. Notebook = fichier <i>Mathematica</i> = NomDuFichier.nb	3
III. Travailler avec <i>Mathematica</i> (1h30)	3
A. Dans l'interface seule (45 min avec les exercices)	3
B. Dans l'interface avec le noyau (45 min avec les exercices)	6
<b>Exercices</b>	10
I. Interface	10
A. Mise en forme	10
B. Saisie d'expressions	10
II. Un tour introductif	10
A. Opérations de base (symboliques avec précision arbitraire)	11
B. Opérations de base (numériques)	12
C. Nombres et Fonctions élémentaires	12
D. Fonctions	13
E. Graphiques 2D, 3D, sons et animations	18
I. Suppléments	18
A. Langage naturel	18
B. Step-by-step	19
I. Annexes	19
A. Wolfram et autour	19
B. Liens vers d'autres tutoriels/cours à l'UPMC	21
C. Quelques autres outils et logiciels	21