TD Algorithmique n°1

PTSI B Lycée Eiffel

18 septembre 2012

Usage de l'informatique en mathématiques

Pour ces quelques TD d'algorithmique, qui ne s'étendront que sur la première période de la première année, nous allons essayer d'aborder l'outil informatique d'un point de vue un peu théorique : à quoi sert-il ? Quels sont les différents types de logiciels qui peuvent être utiles en mathématiques, et comment s'en sert-on ? Nous présenterons en particulier les différentes fonctionnalités du logiciel Maple, que nous manipulerons plus précisément lors des TP d'informatique. Ce logiciel présente l'intérêt d'être à la fois un outil de calcul formel, et de disposer d'un langage de programmation intégré, permettant donc de programmer des algorithmes (pas trop compliqués en ce qui nous concerne). Mais je commence déjà à vous parler de choses que je n'ai pas bien définies !

L'ordinateur

Vous avez déjà l'habitude de manipuler un outil informatique pour faire des mathématiques, mais qui n'est pas aussi évolué qu'un ordinateur, et qui présente l'intérêt d'être entièrement dédié à l'usage mathématique (et possède de ce fait, par exemple, un clavier adapté à la saisie rapide d'expressions mathématiques) : la calculatrice. Elle fonctionne sur le même principe que l'ordinateur : un outil électronique qui est capable de traiter de l'information numérique (des suites de 0 et de 1) extrêmement rapidement. Au départ, une calculatrice comme un ordinateur ne sait absolument rien faire, si ce n'est justement lire des 0 et des 1. Il faut donc, pour en faire un usage pratique, commencer par lui expliquer ce qu'est un nombre, comment on effectue les opérations élémentaires sur les nombres etc. Nous ne remonteront jamais jusqu'à une programmation aussi élémentaire, mais connaitre les principes de fonctionnement d'un algorithme est tout de même important.

Algorithmes et langages de programmation

Un algorithme est la description d'une suite de calculs permettant de résoudre un problème mathématique donné. Vous ne vous êtes sûrement jamais demandé comment votre calculatrice faisait pour afficher des valeurs approchées très précises de nombres comme $\ln(5)$ ou $\sqrt{38}$. Elle n'a évidemment pas en mémoire les valeurs possibles de tous les ln ou racines carrées de tous les nombres possibles, elle effectue donc sûrement une suite de calcul, en appliquant un algorithme qui a été préprogrammé dans sa mémoire. Par exemple, pour calculer une valeur approchée de $\sqrt{2}$, on peut appliquer l'algorithme suivant :

prendre $x = 2$
effectuer n fois de suite l'opération suivante : remplacer x par $x/2 + 1/x$

Plus n est grand, plus la valeur approchée sera bonne (cet embryon de programme est très imprécis). Un langage de programmation, c'est un langage, donc une syntaxe, des mots, des expressions, qui est implémenté dans la machine, et qui permet de lui décrire des algorithmes à l'aide de termes qu'elle comprend (cela suppose qu'il y ait déjà eu une étape de programmation préalable, pour que la machine puisse interpréter des instructions du type « effectuer n fois de suite »). Toute calculatrice est dotée d'un langage de programmation rudimentaire. Sur ordinateur, il existe des programmes qui sont de

purs langages de programmation, ne sachant faire que des opérations très élémentaires (additions, multiplications), mais disposant d'une syntaxe précise permettant de programmer à peu près tout et n'importe quoi. Des exemples de tels langages : Java, Python, C (et son collègue C++) et bien d'autres. Ils sont omniprésents dans l'informatique actuelle, à peu près tous les outils informatiques que vous manipulez quotidiennement ont été programmés avec de tels langages. Notre logiciel Maple dispose lui aussi d'un langage de programmation, assez similaire à celui des calculatrices.

Calcul formel

Un logiciel de calcul formel a une portée toute autre : il a déjà été intensivement programmé, de façon à posséder directement de nombreuses fonctionnalités de calcul avancé. Ainsi, on ne s'embêtera sûrement pas dans ce genre de logiciel à écrire un programme pour calculer des racines carrées (si ce n'est dans un but pédagogique), ça a déjà été fait pour nous ! Le logiciel de calcul formel est vraiment une gigantesque calculatrice permettant de faire à peu près tout ce dont vous pouvez rêver comme calculs mathématiques. Surtout, il dipose de la capacité d'effectuer du calcul formel et pas uniquement du calcul numérique approché. Si vous cherchez par exemple à résoudre une équation du second degré, une calculatrice basique ne vous donnera rien de mieux que des valeurs approchées des solutions. Un logiciel de calcul formel vous donnera des solutions exactes (sous la forme que vous ne le feriez après votre calcul de discriminant). De même pour un calcul de dérivée : Maple sera tout à fait capable de nous dire que la dérivée de $x \mapsto \tan(x)$ est $x \mapsto 1 + \tan^2(x)$. Il est aussi capable de manipuler des objets formels de façon très efficace, par exemple faire du développement ou de la factorisation de polynômes. Nous essaierons de combiner les deux aspects de Maple cette année, pour faire du calcul bien entendu mais aussi pour s'entrainer à la programmation élémentaire, y compris sur des choses que Maple sait déjà très bien faire tout seul.

Quelques exemples de problèmes d'algorithmique

Calcul de puissances

Pour illustrer cet aspect, considérons un problème simple : le calcul de la puissance d'un nombre réel x donné. Maple sait évidemment calculer des puissances très efficacement, mais essayons de nous demander comment il s'y prend. Imaginons donc que nous disposons d'une machine qui ne sait faire que des multiplications. On vous donne un réel x et on vous demande de calculer x^{16}. Quelle est la méthode élémentaire pour y parvenir ? Combien d'opérations effectue-t-elle ? En étant un peu plus malin, combien d'opérations au minimum faudrait-il ? On cherche désormais une méthode permettant de calculer x^n pour un entier $n > 0$ quelconque. Réfléchir à ce problème et tenter de décrire un algorithme y répondant le mieux possible.

Résolution d'équations du second degré

Décrire un algorithme permettant d'effectuer la résolution d'équations du second degré. On supposera qu'on dispose comme données des valeurs des coefficients a, b et c de l'équation.

Présentation d'un outil de calcul formel : le logiciel Maple

Fonctionnement général

Lorsqu'il est lancé, Maple affiche une fenêtre principale usuellement appelée feuille de calcul. Sur la première ligne se trouve le symbole [> qui est une invite de commande, c'est-à-dire qu'elle invite l'utilisateur à taper une commande. Celle-ci ne sera interprétée et exécutée que lorsque l'utilisateur aura appuyé sur la touche Entrée (si on veut passer à la ligne sans déclencher l'évaluation, on tape simultanément Shift et Entrée). Une telle commande doit impérativement se terminer par un ; (cas le plus fréquent, Maple affiche alors le résultat) ou par un : (Maple n'affiche rien). Dans le cas contraire, un message d'erreur sera renvoyé. Par exemple, la ligne [> 1 + 1; renvoie la valeur 2, et la ligne [> 1 + 1 : ne renvoie rien (et n'a donc aucun intérêt, mais par exemple [> $x := 1 + 1$: ne

renvoie rien mais effectue une action, en l'occurence l'affectation de la valeur 2 à la variable x). Le résultat de la dernière commande est réutilisable via le symbole ", et les deux précédents via "" et """. Les commandes peuvent être très variées, et composées de plusieurs instructions, éventuellement organisées en programmes. Commençons par le plus simple.

Calculs en Maple

Les calculs élémentaires utilisent les opérations usuelles ($+$,$-$,$*$,$/$), les puissances ($\wedge$), factorielles (!), racines carrées ($sqrt$) et bien d'autres encore. L'affichage des résultats peut par contre suprendre au premier abord puisque Maple affiche dès qu'il le peut des résultats exacts. Ainsi, l'instruction [> 2/3 + 4/5; renvoie 22/15 et l'instruction [> 47!; renvoie la valeur complète de l'entier 47! (une soixantaine de chiffres). Si l'on veut une valeur numérique approchée du résultat, il faut ajouter l'instruction $evalf$ (évaluation flottante), par exemple [> $evalf(2/3 + 4/5)$; renverra la valeur 1.4666666667. Par défaut, Maple donne 10 décimales, mais la précision est paramétrable via la variable $Digits$ (l'instruction [> $Digits := 25$; fait passer le nombre de décimales à 25). On peut également utiliser l'instruction $evalf(x, n)$ pour évaluer le nombre x avec n décimales. Certaines constantes sont prédéfinies en Maple, parmi lesquelles π (Pi, avec une majuscule à ne pas oublier), et un certain nombre de fonctions existent également, nous allons maintenant dresser une liste de celles qui sont le plus utilisées.

Variables et types

Comme tout langage de programmation, Maple manipule les objets sous forme de variables auxquelles sont associées un type d'objets mathématiques, avec des restrictions de calcul liées à ce type. C'est en fait comme en mathématiques : si vous manipulez un objet noté x qui est censé être un nombre réel, vous ne pourrez pas faire n'importe quoi avec, par exemple tenter de le dériver n'a aucun sens. On reviendra plus en détail sur cet aspect lors du prochain TD, lorsque nous décrirons plus en détail les principes de base de la programmation, où il est nécessaire de préciser le type des variables qui interviendront dans un programme. Pour l'instant, contentons-nous de constater que certaines commandes Maple ne peuvent s'appliquer qu'à certains types d'objets.

Fonctions utiles

Nous allons finir ce premier TD avec une première liste de commandes Maple que vous aurez l'occasion d'utiliser lors de vos séances sur machine. Il ne s'agit pas d'être exhaustif, les possibilités de Maple étant très très larges. En fait, les logiciels de calcul formel fonctionnent sous forme de bibliothèques de commandes dédiées à un certain type de calcul ou d'objets. Ainsi, une bibliothèque consacrée au calcul polynomial contiendra des commandes permettant de factoriser un polynome, de le dériver, de déterminer ses racines etc. Le logiciel est par ailleurs accompagné d'une aide détaillée (il faudra apprendre à s'en servir) , accessible par les menus ou en tapant simplement la commande ? Si vous tentez de taper un ? suivi d'un nom de commande, Maple vous ouvrira la section d'aide consacrée à cette commande, avec son champ d'application, sa syntaxe et quelques exemples.

Sur les entiers

Les fonctions suivantes doivent (ou du moins peuvent) prendre un ou plusieurs arguments entiers : abs (valeur absolue), $sign$ (signe), $factorial$ (même résultat que le !), $irem$ (reste de la division euclidienne, deux arguments, $iquo$ (quotient de cette même division euclidienne), $isqrt$ (racine carrée entière), $isprime$ (test de primalité, renvoie $true$ si l'argument est premier, $false$ sinon), $ifactor$ (décomposition en facteurs premiers), $igcd$ et $ilcm$ (renvoient respectivement le PGCD et le PPCM, deux arguments), et encore bien d'autres.

Sur les rationnels

Rien de particulier, sachez seulement que Maple fait tout seul les simplifications de fraction, et affiche donc les résultats sous forme irréductible.

Sur les réels

Maple continue de ne pas afficher de valeurs approchées pour les nombres alébriques (puissances de rationnels) et refusera même de développer les résultats si on ne l'y force pas, ce qu'on peut faire via l'instruction *expand*.

Il existe trop de fonctions prédéfinies sur les réels pour les lister, mais par exemple sin, exp, ln existent, mais Maple ne vous renverra pas de valeur approchée si « ça ne tombe pas juste », il laissera par exemple sous la forme $sin(3)$. Toutefois, si l'argument est donné sous forme flottante, le résultat le sera aussi (donc $sin(3.0)$ donne une valeur approchée).

Sur les complexes

Les fonctions évidentes Re, Im (avec des majuscules), $conjugate$, abs (pour le module), $argument$ sont prédéfinies. Par ailleurs, le nombre I est également prédéfini (encore une fois, avec une majuscule).

Manipulations d'objets plus complexes

Nous n'allons pas trop entrer dans le détail pour l'instant, mais Maple sait notamment faire du calcul sur des polynomes et plus généralement sur des fonctions. Quelques instructions utiles : *sort* permet d'afficher un polynome par puissances décroissantes (la notation habituelle, donc), $factor$ permet de factoriser une expression, *expand* de la développer, $diff$ permet de dériver une fonction, mais il faut préciser la variable et éventuellement le nombre de dérivations (par exemple [> $diff(x^4, x\$2)$; renverra $12x^2$ (dérivation seconde par rapport à la variable x). On peut même faire de la dérivation symbolique avec l'instruction D (par exemple [> $D(cos)$; renvoie $-sin$. Notons au passage que le symbole pour désigner la composition de fonctions en Maple est le @. Pour les intégrales, la commande est int, qui si vous ne lui donnez comme arguments qu'une fonction et une variable va en fait calculer une primitive de la fonction. Pour calculer véritablement une intégrale, il faut préciser les bornes de la façon suivante : [> $int(f(x), x = min..max)$;, par exemple [> $int(x^2, x = 0..3)$; renverra 9. Les bornes en question peuvent être infinies ($infinity$).

Maple peut également résoudre des équations par les commandes *solve* (résolution exacte, par exemple [> $solve(x^2 - 4x + 7, x)$; vous renverra des belles solutions avec radicaux), $fsolve$ pour une solution numérique approchée, *dsolve* pour des équations différentielles [> $dsolve(diff(y(x), x) = y(x), y(x))$; vous répondra $Cexp(x)$, C désignant une constante réelle) et $rsolve$ pour calculer des suites récurrentes.

Module graphique

Maple permet également de tracer des graphiques divers et variés. Dans un premier temps, contentons-nous de signaler qu'il peut tracer des courbes représentatives de fonctions via l'instruction *plot* et même des courbes en trois dimensions grâce à *plot3d*, par exemple [> $plot(cos(x) * exp(3x^2), x = -3..12)$; ou [> $plot3d(sin(x) * exp(y), x = 0..2, y = 0..3)$;.