DIALOGUER ET CALCULER AVEC MAPLE

Bernard Dupont

Bernard.Dupont@univ-lille1.fr

Les précédents exemples de worksheets montrent que Maple peut satisfaire de nombreux besoins dans le domaine de la modélisation économique. Ses compétences ne sont clairement pas réduites au rôle de super-calculatrice, mais savoir l'utiliser comme simple calculatrice permet déjà de se faire une idée de quelques règles spécifiques au maniement du logiciel.
Ce chapitre a trois objectifs : apprendre les bases du "dialogue" avec Maple; maîtriser les opérations et les fonctions élémentaires; connaître les types de nombres acceptés par le logiciel.

Dialoguer avec Maple

A l'ouverture d'une nouvelle feuille de travail en mode Worksheet paramétré comme indiqué dans le chapitre 2 (Tools→Options→Display→Input display : Maple Notation et Output display : 2-D Math Notation), Maple ouvre une page blanche sur la première ligne de laquelle est positionné un prompteur rouge (symbole "supérieur stritement à", soit **>**) qui attend une commande valide de l'utilisateur.

Pour connaître la somme de 1 plus 1, il est logique de taper juste derrière le prompteur 1+1, puis de valider l'instruction par la touche retour du pavé alphabétique ou la touche d'entrée du pavé numérique :

```
> 1+1
Warning, inserted missing semicolon at end of statement
                                2
```

Maple affiche un message d'avertissement/erreur signalant que la commande aurait dû être validée par un "point virgule" (semicolon) placé juste après l'énoncé (statement).

```
> 1+1;
                                2
```

D'où la première leçon : **toute requête adressée à Maple doit être clôturée par un point virgule**. Alors, en appuyant sur la touche retour du pavé numérique ou du pavé alphabétique, le résultat ou un message d'erreur s'affiche sous la forme d'un écho bleu au milieu de la ligne qui suit l'instruction écrite en rouge.

Si, pour une raison ou une autre, on ne veut pas afficher le résultat, il faut terminer l'instruction par le symbole "deux points", **:** , qui signale au logiciel que la commande doit être évaluée mais qu'il est inutile d'envoyer la réponse sur l'écran. Par exemple, demandons l'évaluation de factorielle 100 sans que le résultat ne soit affiché :

```
> 100!:
```

L'évaluation a bien été faite, ce qu'on vérifie ici en tapant sur la ligne suivante le symbole "pour cent" **%**, encore appelé dito (en anglais : ditto), qui est toujours égal au dernier résultat **chronologique** produit par Maple :

```
> %;
933262154439441526816992388562667004907159682643816214685929638952175999\
    932299156089414639761565182862536979208272237582511852109168640000000\
    000000000000000000
```

Le dito est souvent utilisé en pratique puisqu'il évite de recopier des résultats intermédiaires. Il existe aussi un double dito,**%%**, qui vaut l'avant dernier résultat chronomogique, et le triple

dito,%%%, qui vaut l'avant avant dernier résultat chronologique. Indubitablement utiles, ces symboles doivent être maniés avec précaution car l'ordre chronomogique peut différer de l'ordre d'affichage des résultats.

Poursuivons. Un prompteur peut être suivi d'une ou de plusieurs commandes. Dans le cas d'une succession de commandes, il ne faut pas oublier de clôturer chaque commande individuelle par un point virgule ou deux points. Les outputs sont renvoyés dans l'ordre des commandes terminées par un point virgule ou un "deux points" (donc, si toutes les commandes se terminent par deux points, aucun écho n'est renvoyé).

```
> 4+5;45+66;87+95+87+39;#toutes les réponses s'affichent
```

9
111
308

```
> 4+5;45+66:87+95+87+39;#seules la première et la troisième
  requêtes ont un écho
```

9
308

```
> 4+5:45+66:87+95+87+39:#aucune réponse ne s'affiche
```

En informatique, la prudence conseille de séparer nettement les instructions en les plaçant les unes sous les autres, ce qui se fait en tapant la combinaison des deux touches "Majuscule"+ "Enter" après chaque commande terminée par un deux points ou un point virgule.

```
> 4+5;#taper "Majuscule" puis "Enter" pour passer à la ligne
  45+66;#taper "Majuscule" puis "Enter" pour passer à la
  ligne
  87+95+87+39;
```

9
111
308

On insère des commentaires après les instructions en faisant précéder leur texte par un dièse #. Ce qui permet de documenter les lignes sans aucun effet sur l'évaluation des commandes.

Dernière remarque, qui peut être utile pour économiser de la place - par exemple pour une sortie papier de la feuille de travail - et qui va à l'encontre du principe de clarté de l'exposition des lignes de commande, on peut séparer les instructions d'une ligne par des virgules, ce qui a pour effet de renvoyer les résultats en ligne. Bien entendu, la dernière instruction doit être suivie d'un point-virgule.

```
> 4+5,45+66,87+95+87+39;
```

9, 111, 308

Opérations élémentaires

Les quatre opérations élémentaires sont celles du pavé numérique. En conséquence, la multiplication est obligatoirement signalée par une étoile * (touche de la multiplication sur le pavé numérique) et ne peut en aucun cas être remplacée par un espace comme on le ferait à la main.

```
> 5+6;#addition
  78-56;#soustraction
  8-177;#soustraction
```

11
22

$$-169$$

```
> 2 7;#oublier de signaler la multiplication renvoie un
  message d'erreur
Error, unexpected number
> 2*7;#multiplication licite
```

$$14$$

```
> 48/6;69/12;#division
```

$$8$$

$$\frac{23}{4}$$

La factorielle de l'entier naturel N s'écrit comme on le fait "à la main" en ajoutant un point d'exclamation, soit **N!** :

```
> 25!;
```

$$15511210043330985984000000$$

La puissance d'un nombre a une syntaxe particulière. L'expression "*x* à la puissance *n*" s'écrit en tapant d'abord le nombre **x**, puis en tapant la combinaison de touches "accent circonflexe"+ "espace" qui donne **^**, enfin le nombre **n**.

```
> 2.065^9;
```

$$682.7795483$$

La racine carrée d'un réel positif *x* se fait par la fonction pré-programmée **sqrt(x)** ou par **x^(1/2)** :

```
> sqrt(22.654);
  22.654^(1/2);
```

$$4.759621834$$

$$4.759621834$$

La racine *n*-ième du réel *x* s'exprime à l'aide de l'opération puissance, soit **x^(1/n)** :

```
> 2.065^(1/9); #racine 9-ième de 2.065
```

$$1.083904745$$

Il existe un code de priorité sur les opérations. Maple évalue d'abord les puissances, puis les produits et quotients, et enfin les additions et les soustractions. Il n'est pas toujours facile d'anticiper l'ordre d'exécution des calculs, surtout quand les formules sont longues. Pour éviter des désagréments, on a intérêt à utiliser des parenthèses,qui traduisent la propriété de commutativité de la multiplication par rapport à l'addition comme dans les calculs "à la main".

```
> 1+((2+58)*(478/7))/((5!-2!)/5),1+((2+58)*(478/7))/(5!-2!)
  /5,1+(2+58)*(478/7)/5!-2!/5;
```

$$\frac{72113}{413}, \frac{3281}{413}, \frac{1216}{35}$$

Nombres

Mathématiquement, les ensembles de nombres sont repérables par leurs propriétés structurelles algébriques (groupe, anneaux, corps) et sont hiérarchisés : $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$. D'un point de vue informatique, le regroupement des nombres se fait par type et c'est logique car même l'ordinateur le plus puissant n'atteindra jamais la quantité de chiffres nécessaires pour exprimer $\sqrt{2}$ avec exactitude. On peut connaître le "type Maple" du nombre **N** par la commande **whattype(N)**. Maple reconnaît les types de nombres suivants : **entiers** (integer),

fractionnaires (fraction) et **décimaux de la forme $a\ 10^b$** (float). Ces types sont réunis en classes. Les **rationnels** (rational) réunissent les entiers et les fractions, les **nombres** (numeric) réunissent les rationnels et les nombres décimaux.

Entiers relatifs

En théorie, Maple accepte n'importe quel nombre entier relatif, qu'il traite par défaut en base 10. On l'écrit comme on le ferait à la main. En particulier, il est précédé du signe **-** s'il est négatif.

```
> 100!;
  whattype(%);
  -45;
  whattype(%);
```

```
933262154439441526816992388562667004907159682643816214685929638952175\
    999932299156089414639761565182862536979208272237582511852109168640\
    000000000000000000000
                                integer
                                  -45
                                integer
```

En pratique, les nombres astronomiques sont refusés. La commande **kernelopts(maxdigits)** renvoie le nombre maximum de chiffres accepté par le logiciel pour traiter un nombre donné. Ce nombre maximum dépend de l'ordinateur utilisé. Dans notre cas, Maple peut travailler sur des entiers comportant jusqu'à 268.435.448 de chiffres!!

```
> 10^(100!);#nombre astronomique
  kernelopts(maxdigits);
```

```
Error, numeric exception: overflow
                              268435448
```

Les calculs sur les entiers acceptés sont toujours exacts. Le **principe d'exactitude du résultat** est central dans les manipulations algébriques effectuées par Maple.

```
> 100!*101-101!;
```

```
                                  0
```

Les nombres entiers sont un objet d'études mathématiques en soi. De nombreuses fonctions Maple leur sont consacrées : **isprime** (test de primalité), **igcd** (détermination du PGCD), **ilcm** (détermination du PPCM), **ifactor** (factorisation), etc.. Les économistes n'y ont quasiment jamais recours.

Rationnels

Les rationnels comprennent les entiers et les nombres fractionnaires que Maple simplifie automatiquement sous forme d'un quotient de deux entiers premiers entre eux, avec un dénominateur strictement positif.

```
> 15/25;whattype(%);
```

```
                                 3/5
                              fraction
```

Lorsque les calculs n'impliquent que des rationnels, Maple renvoie toujours un résultat exact, ce qui signifie que les divisions ne sont pas effectuées sauf si le résultat est un entier. Autrement dit, la fraction $\frac{1}{3}$ n'est pas approximée par 0, 333 ... car le principe d'exactitude du

résultat s'y oppose.

```
> 1/3+2/7-111/58;
```

$$-\frac{1577}{1218}$$

Si on a absolument besoin de travailler sur un nombre rationnel avec parties entière et décimale, il faut forcer Maple à approcher ce nombre par conversion dans le type *float*. Pour le nombre **N** et le nombre de chiffres significatifs **s**, on invoque la commande **evalf** (**eval**uation **f**loating) dont la syntaxe est **evalf(N,s)** ou, de manière équivalente, **evalf[s](N)**. Par défaut, le nombre de chiffres significatifs est 10. Si on s'en contente, il n'est pas nécessaire d'utiliser le second argument. Le nombre maximum de chiffres significatifs est celui donné par la commande **kernelopts(maxdigits)**.

```
> evalf(%);
  evalf(%%,250);
```

$$-1.294745484$$

$$-1.2947454844006568144499178981937602627257799671592775041050903119\backslash$$
$$86371100164203612479474548440065681444991789819376026272577996715\backslash$$
$$92775041050903119868637110016420361247947454844006568144499178981\backslash$$
$$93760262725779967159277504105090311986863711001642036$$

(3.2.1)

Réels

Les réels regroupent les rationnels et les éléments de type *float* au sein du type *numeric*, à quoi s'ajoutent des nombres particuliers qui seront présentés plus bas et les irrationnels regroupés dans le type *realcons*.
Les nombres du type *float* ont une partie entière - positive, négative ou nulle - et une partie décimale - éventuellement nulle ou vide, séparées à l'anglo-saxonne par un point.

```
> whattype(1.56);whattype(0.56);whattype(1.);
```

float

float

float

Les calculs élémentaires sur des nombres de type *float* renvoient un résultat avec 10 chiffres significatifs par défaut. La réponse est toujours un nombre de type *float*. Si les calculs mélangent des nombres entiers, rationnels et au moins un nombre de type *float*, **le principe d'exactitude du résultat est remis en cause** et la réponse est toujours un nombre de type *float*.

```
> restart;22/7+5;whattype(%);22./7+5;whattype(%);22/7.+5;
  whattype(%);22./7.+5;whattype(%);
```

$$\frac{57}{7}$$

fraction

8.142857143

float

8.142857143

float

8.142857143

float

On peut augmenter à volonté le nombre de chiffres significatifs en déclarant préalablement le degré de précision avec l'instruction **Digits:=n**, n étant le nombre de chiffres significatifs

désiré.

```
> Digits:=25;22./7;Digits:=75;22./7;
```

$$Digits := 25$$

$$3.1428571428571428571429$$

$$Digits := 75$$

$$3.142857142857142857142857142857142857142857142857142857142857\backslash$$
$$142857142857142857142857142857142857142857142857142857142857\backslash$$
$$1428571428571428571428571428571428571428571428571428571428571\backslash$$
$$4285714$$

Les irrationnels tels que $\sqrt{2}$, e^2, $ln2$, π ne sont pas du type *numeric* mais du type *realcons* (**Real cons**tants) . **En vertu du principe d'exactitude du résultat, ils sont manipulés comme expressions littérales dans les calculs et seules quelques rares simplifications élémentaires sont effectuées**. Leur conversion en nombre approché du type *float* se fait à l'aide de la commande **evalf** décrite plus haut.

```
> sqrt(2);evalf(%);sqrt(2)^2;evalf(%);sqrt(2)*8.4;evalf(%);
  #cas des nombres irrationnels
```

$$\sqrt{2}$$

$$1.414213562$$

$$2$$

$$2.$$

$$8.4\sqrt{2}$$

$$11.87939392$$

```
> log(2);evalf(%);(log(2))^(1/2);evalf(%);log(2)*8.4;evalf
  (%);#cas des nombres logarithmiques
```

$$\ln(2)$$

$$0.6931471806$$

$$\sqrt{\ln(2)}$$

$$0.8325546112$$

$$8.4\ln(2)$$

$$5.822436317$$

```
> exp(2);evalf(%);exp(2)^(1/2);evalf(%);exp(2)*8.4;evalf(%)
  ;#cas des nombres exponentiels
```

$$e^2$$

$$7.389056099$$

$$\sqrt{e^2}$$

$$2.718281828$$

$$8.4\,e^2$$

$$62.06807123$$

```
> Pi;evalf(Pi);evalf(Pi,55);Pi*2^2;evalf(%);#cas du nombre
  π
```

$$\pi$$

$$3.141592654$$

$$3.1415926535897932384626433832795028841971693993751058210$$

$$4\pi$$

$$12.56637062$$

Il se peut que Maple renvoie un résultat du type *float* dans la notation scientifique $a\ 10^b$ où a, la mantisse, est un nombre positif ou négatif et b, la puissance, est un nombre entier positif ou négatif.

```
> (-9.87)^151;
```

$$-1.386404799\ 10^{150}$$

Les nombres complexes

L'écriture et les calculs sur des complexes se font sous forme cartésienne ou sous forme trigonométrique.

Forme cartésienne d'un nombre complexe

Le nombre complexe $a + b\ i$ s'écrit en Maple **a+b*I**. L'imaginaire pur i tel que $i^2 = -1$ s'écrit directement **I**. Pour Maple, le type d'un nombre complexe écrit sous forme cartésienne ne comportant pas de variable libre est *complex(extended_numeric)*.

```
> 2+3*I;whattype(%);
```

$$2 + 3\ I$$

$$complex(extended\_numeric)$$

Soit le complexe $a+b\ i$ où a et b sont des entiers relatifs. Le résultat d'un calcul n'impliquant que ce nombre est un complexe de la forme $\alpha + \beta\ i$, où α et β sont des entiers. Par extension, des calculs qui n'impliquent que des nombres de ce type donnent un résultat du même type.

```
> 3-5*I;(3-5*I)^4;(3-5*I)*(9+14*I);
```

$$3 - 5\ I$$

$$-644 + 960\ I$$

$$97 - 3\ I$$

De même, si a et b sont des rationnels du type *fraction*, le résultat de calculs sur $a+b\ i$ est un complexe de la forme $\alpha + \beta\ i$, où α et β sont des rationnels du type *fraction*. Et des calculs qui n'impliquent que des nombres de ce type donnent un résultat du même type et par conséquent le principe d'exactitude du résultat est toujours vérifié.

```
> 3/7+5/6*I;(3/7+5/6*I)^4;(3/7+5/6*I)*(9/11-14/3*I);
```

$$\frac{3}{7} + \frac{5}{6}\ I$$

$$-\frac{775799}{3111696} - \frac{4505}{6174}\ I$$

$$\frac{2938}{693} - \frac{29}{22}\ I$$

Si l'un au moins des nombres a et b est un réel du type *float*, le résultat des calculs sur $a+b\ i$ est un complexe de la forme $\alpha + \beta\ i$, où α et β sont des réels du type *float*.

```
> 3.12+0.655*I;(3.12+0.655*I)^4;(4.566-1.458*I)*(
  (3.1+9.8*I)+(-7.+5.4117*I));
```

$$3.12 + 0.655\ I$$

$$69.88480005 + 76.06585740\ I$$

$$4.3712586 + 75.1428222\ I$$

Enfin, si l'un au moins des nombres a et b est un réel du type *realcons*, le principe d'exactitude guide les manipulations algébriques et le résultat des calculs n'est pas nécessairement un complexe de la forme $\alpha + \beta\ i$, où α et β sont des réels du type *float*.

Pour forcer le passage à la forme cartésienne, on utilise la commande **evalc** (**eval**uation **c**omplex).

```
> (sqrt(2)+I)/(1+sqrt(2)*I);evalc(%);
```

$$\frac{\sqrt{2}+\mathrm{I}}{1+\mathrm{I}\sqrt{2}}$$

$$\frac{2}{3}\sqrt{2}-\frac{1}{3}\,\mathrm{I}$$

Pour connaître une valeur réelle approchée des parties réelle et imaginaire du nombre $\frac{\sqrt{2}+I}{1+I\sqrt{2}} = \frac{2}{3}\sqrt{2}-\frac{1}{3}\,\mathrm{I}$, on peut utiliser les commandes **Re** et **Im** combinées avec **evalf**.

```
> evalf(Re(2/3*sqrt(2)-1/3*I));evalf(Im(2/3*sqrt(2)-1/3*
  I));%%+%*I;
```

$$0.9428090414$$

$$-0.3333333333$$

$$0.9428090414 - 0.3333333333\,\mathrm{I}$$

```
> evalf(Re((sqrt(2)+I)/(1+sqrt(2)*I)));evalf(Im((sqrt(2)+
  I)/(1+sqrt(2)*I)));%%+%*I;
```

$$0.9428090414$$

$$-0.3333333333$$

$$0.9428090414 - 0.3333333333\,\mathrm{I}$$

mais il est plus simple d'utiliser la commande **evalf** sur le complexe considéré

```
> evalf((sqrt(2)+I)/(1+sqrt(2)*I));
```

$$0.9428090417 - 0.3333333331\,\mathrm{I}$$

Mentionnons la commande **conjugate(z)** qui, comme son nom l'indique, retourne le conjugué du nombre complexe *z*.

```
> conjugate(2+5*I);
```

$$2-5\,\mathrm{I}$$

Forme trigonométrique des nombres complexes

Soit $z = a + b\,I$ un nombre complexe sous sa forme cartésienne. Sa forme trigonométrique est donnée par le couple (θ, ρ) où θ est le module et ρ l'argument (réel unique compris entre $-\pi$ et π, défini à $2\,k\,\pi$ près, vérifiant $\tan(\rho) = \frac{b}{a}$). La commande **abs** renvoie le module et la commande **argument** ... l'argument de *z*.

```
> 2+3*I;
  abs(%);
  argument(%%);
```

$$2+3\,\mathrm{I}$$

$$\sqrt{13}$$

$$\arctan\left(\frac{3}{2}\right) \qquad \textbf{(3.4.2.1)}$$

Ces deux commandes obéissent au principe d'exactitude du résultat. Pour obtenir des nombres réels approchés, on utilise **evalf**.

```
> evalf(abs(1/7-5/8*I));
```

```
evalf(argument(1/7-5/8*I));
```

0.6411186812

-1.346085158 **(3.4.2.2)**

Nombres particuliers

Maple attribue à des nombres particuliers des identificateurs. Il s'agit du nombre π (3.14...) identifié par **Pi**, de la constante d'Euler dite "gamma" identifiée par **gamma**, l'imaginaire pur *i* identifié par **I** , de la constante de Catalan identifiée par **Catalan** et de l'infini identifié par **infinity**. Ils sont manipulés dans les calculs suivant la règle d'exactitude du résultat. Leur valeur approchée est renvoyée par **evalf**.

```
> Pi;evalf(Pi);
  gamma;evalf(gamma);
  I;evalf(I);
  Catalan;evalf(%);
  infinity;evalf(infinity);
```

π

3.141592654

γ

0.5772156649

I

1. I

Catalan

0.9159655942

∞

Float(∞)

Fonctions prédéfinies basiques

Les fonctions usuelles de l'économiste sont disponibles dans la bibliothèque principale. Bien que Maple soit en mesure de les traiter dans des situations plus générales, on supposera ici que ce sont toutes des fonctions numériques d'une variable réelle. Autrement dit, l'ensemble de départ est un sous-ensemble de $\mathbb{R}$ et l'ensemble d'arrivée un sous-ensemble de $\mathbb{R}$.

Fonction exponentielle de base e

L'exponentielle du nombre **N** s'écrit **exp(N)**. Le résultat d'une évaluation de **exp** obéit au principe d'exactitude. Pour obtenir une approximation, on doit, suivant les cas, introduire des réels du type *float* ou utiliser **evalf.**

```
> exp(0);exp(Pi);evalf(%,25);
```

1

e^{π}

23.140692632779269005729086

```
> exp(24/79);exp(24./79);evalf(exp(1)+56);
```

$e^{\frac{24}{79}}$

1.354994599

58.71828183

Fonction exponentielle de base *a*

La fonction exponentielle de base *a* n'est autre qu'une fonction puissance **a^n**, **n** étant un nombre.

```
> 4^7;10^(2/3);10^(2./3);
```

16384

$10^{2/3}$

4.641588834

Fonction logarithme népérien

La fonction logarithme népérien s'écrit indifféremment à l'anglo-saxonne **log()** ou à la française **ln().**

```
> log(5);ln(5);log(5.);ln(5.);
```

$\ln(5)$

$\ln(5)$

1.609437912

1.609437912

Elle est la réciproque de la fonction exponentielle.

```
> ln(exp(4));
```

4

Fonction logarithme de base a

Quand on veut un logarithme décimal, on utilise la commande **log10()**. Sinon, on utilise **log[a]().**

```
> log10(10);log10(Pi);evalf(log10(Pi));log[2](2);log[2](10)
  ;evalf(log[2](10));
```

1

$\frac{\ln(\pi)}{\ln(10)}$

0.4971498728

1

$\frac{\ln(10)}{\ln(2)}$

3.321928095

Fonction cosinus

On calcule le cosinus du nombre **N** en utilisant la fonction prédéfinie **cos**. On écrit donc **cos(N)**. Maple distingue la lettre grecque "pi" et le nombre irrationnel "Pi", qui est, rappelons-le, une constante particulière. Dans l'exemple qui suit, **pi** est une variable muette alors que **Pi** est bien un nombre fixe.

```
> cos(pi);evalf(cos(pi));cos(Pi);
```

$\cos(\pi)$

$$\cos(\pi)$$

$$-1$$

```
> cos(0);cos(Pi/2);cos(Pi);cos(3*Pi/2);#quelques cosinus
  remarquables
```

$$1$$

$$0$$

$$-1$$

$$0$$

Fonction sinus

On calcule le cosinus d'un nombre en utilisant la fonction prédéfinie **sin**.

```
> sin(0);sin(Pi/2);sin(Pi);sin(3*Pi/2);sin(2*Pi);#quelques
  sinus remarquables
```

$$0$$

$$1$$

$$0$$

$$-1$$

$$0$$

Fonction tangente

On calcule la tangente d'un nombre en utilisant la fonction prédéfinie **tan**. Maple est intraitable sur le domaine de définition de cette fonction.

```
> tan(0);tan(Pi/6);tan(Pi/4);tan(Pi/2);tan(4*Pi/6);tan(5*
  Pi/6);tan(Pi);tan(3*Pi/2);
```

$$0$$

$$\frac{1}{3}\sqrt{3}$$

$$1$$

```
Error, (in tan) numeric exception: division by zero
```

$$-\sqrt{3}$$

$$-\frac{1}{3}\sqrt{3}$$

$$0$$

```
Error, (in tan) numeric exception: division by zero
```

Fonction cotangente

On calcule la cotangente d'un nombre en utilisant la fonction prédéfinie **cot**.

```
> cot(0);cot(Pi/6);cot(Pi/4);cot(Pi/2);cot(4*Pi/6);cot(5*
  Pi/6);cot(Pi);cot(3*Pi/2);
Error, (in cot) numeric exception: division by zero
```

$$\sqrt{3}$$

$$1$$

$$0$$

$$-\frac{1}{3}\sqrt{3}$$

$$-\sqrt{3}$$

```
Error, (in cot) numeric exception: division by zero
```

$$0$$

Fonctions trigonométriques inverses : `arcsin()`, `arccos()`, `arctan()`, `arccot()`

Les fonctions trigonométriques inverses **`arcsin`**, **`arccos`**, **`arctan`** et **`arccot`** retournent un angle en radian.

```
> arcsin(1/2);arccos(1/2);evalf(arctan(1/2));evalf(arccot
  (1/2));
```

$$\frac{1}{6}\pi$$

$$\frac{1}{3}\pi$$

$$0.4636476090$$

$$1.107148718$$

Conclusion

Maple calcule vite et bien, avec le degré de précision souhaité. C'est la moindre des choses. Son rayon d'action ne s'arrête pas là. En tant que logiciel de calcul formel, il peut manipuler des mélanges de nombres et de symboles ou même seulement des symboles. Ainsi, toutes les opérations et les fonctions prédéfinies exposées dans ce chapitre acceptent des symboles, par exemple :

```
> Pi*a+2*m+k^i;
  conjugate(a+b*I);evalc(%);
  abs(a+b*I);evalc(%);
```

$$\pi a + 2 m + k^{i}$$

$$\overline{a + \mathrm{I} b}$$

$$a - \mathrm{I} b$$

$$|a + \mathrm{I} b|$$

$$\sqrt{a^2 + b^2} \qquad \textbf{(5.1)}$$

Evidemment, de nouvelles règles doivent être maîtrisées. L'exemple qui suit se sert de la commande universelle **`simplify`** qui sera abondamment commentée et utilisée par la suite. Elle permet d'effectuer de nombreuses simplifications de résultats en faisant intervenir les propriétés usuelles des fonctions utilisées.

```
> restart;
  exp(a)*exp(b);simplify(%);
  exp(a)*exp(b)-exp(a+b);simplify(%);
```

$$e^{a} e^{b}$$

$$e^{a+b}$$

$$e^{a}\,e^{b} - e^{a+b}$$

$$0$$