**Chapitre 2**

**Notions de base de calcul formel en MAPLE**

**2.1 C’est quoi Maple ?**

Maple est un puissant logiciel qui peut être utilisé pour résoudre des problèmes mathématiques d’une manière générale. Les dans les domaines des mathématiques, science et engineering (et plus) peuvent être investigués en utilisant soit des commandes fournies par Maple, soit en utilisant le puissant langage de programmation natif de Maple pour créer ces propres programmes personnalisés.

Nous allons présenter, dans ce qui suit, une entrée en la matière qui fournit ce dont on a besoin pour apprendre les bases de Maple ; de plus amples informations peuvent être trouvées sur le web.

**2.2. Pour Commencer**

La première session Maple s’ouvre avec un dialogue de démarrage qui explique la différence entre le Mode Document et le Mode Feuille de Travail (Worksheet). En utilisant l’un des deux modes, on peut créer des documents mathématiques interactifs de haute qualité. Chaque mode offre les mêmes éléments et fonctionnalité ; la seule différence est la région d’entrée par défaut de chaque mode.

**2.2.1. Mode Document**

Le mode Document est conçu pour effectuer rapidement des calculs. On peut entrer une expression mathématique, et puis évaluer, manipuler, résoudre, ou la tracer avec un petit nombre de touches clavier ou de clics de souris.

Le mode document utilise les Blocks de Document comme région d’entrée par défaut pour cacher la syntaxe de Maple. Une région d’un Bloc de Document est indiquée par deux triangles localisés dans la colonne verticale des Balises le long du volet gauche du document Maple. Ceci permet de se concentrer sur le problème au lieu des commandes pour résoudre le problème. Par exemple, lorsqu’on utilise le menu de contexte en Maple input dans le Mode Document (invoqué en cliquant le bouton droit), l’entrée et la sortie sont reliées en utilisant une flèche ou un signe d’égalité avec un texte auto-documentant indiquant le calcul qui est a été mis en place. La commande utilisée pour résoudre cette expression est cachée.

**2.2.2. Mode Feuille de Travail (Worksheet)**

Le mode Worksheet (ou Feuille de travail) est conçu pour un usage interactif à travers l’utilisation de commandes et la programmation en utilisant le langage Maple.Le mode Worksheet supporte les éléments disponibles en modes Document.

Le mode Worksheet utilise l’invite Maple comme la région d’entrée par défaut.L’invite d’entrée Maple est un crochet rouge. Lorque les menus de contexte sont utilisés en entrée dans le mode Worksheet, toutes les commandes sont affichées.

**2.2.3. Modes Document et Worksheet**

Indépendamment du mode dans lequel on travail, on a la possibilité d’afficher ou cacher les calculs. On peut cacher les commandes dans le Mode Worksheet en ajoutant un bloc document (*document block*) à partir du menu : Format.Creer un Bloc de Document, ou bien on peut afficher les commandes en mode Document en ajoutant une invite Maple à partir du menu : Insertion.Groupe d’exécution.Avant/Après Curseur.

**Pour lancer une session Maple**

1. Dans le dialogue de démarrage, sélectionner "Démarrage avec un Document vide" ou "Démarrage avec une Feuille de

Travail vide.

Ou bien

2. Fermer le dialogue de démarrage.

3. A partie du menu File, sélectionner Nouveau, puis Mode Document ou Mode Feuille de Calcul. Un document vide s’affiche. A chaque fois qu’on ouvre un nouveau document, Maple affiche une liste d’aide rapide des touches de raccourcis les plus importantes. Pour invoquer cette liste, on presse sur la touche F1.

**2.3. Entrée des Expressions**

**2.3.1. Groupes d’exécution**

Un groupe d’exécution est un groupement d’entrées Maple avec leurs sorties Maple correspondantes. Il est distingué par un large crochet, appelé bordure de groupe, à gauche. Un groupe d’exécution pourrait aussi contenir un tracé, un texte, des composants incrustés, une feuille, et un canevas de dessin.

Les groupes d’exécution sont les éléments fondamentaux des calculs et de la documentation dans le document. Si on place le curseur dans une commande d’entrée et on presse la touche Enter ou Return, Maple exécute toutes les commandes saisies dans le groupe d’exécution courant.

**2.3.2. Mode Math vs. Mode Texte**

Le mode d’entrée par défaut dans le Mode Document ou Feuille de Travail est le Mode Math, qui affiche les entrées en 2-D Math. Dans les versions plus anciennes de Maple, les commandes et expressions étaient entrées en utilisant Maple input ou bien 1-D Math.

|  |  |
| --- | --- |
| **Mode Math** | **Mode Texte** |
| *Paramétrage par défaut de Maple*. Notation math standard exécutable.  Elle est aussi appelée **Entrée Math 2-D**. | *Paramétrage par défaut de Maple*. Notation math standard exécutable.  Elle est aussi appelée **Entrée Math 2-D**. |
| Accès à partir du menu **Insertion** 🡪 **Math 2-D** | Accès à partir du menu Insertion 🡪 **Entrée Maple** |
| Lorsqu’on utilise **Math 2-D**, l’icône du mode math est mise en surbrillance dans la barre d’outils :  Math  Texte | Lorsqu’on utilise **Entrée Maple**, l’icône de mode texte est mise en surbrillance dans la barre d’outils :  Math  Texte |
| En mode Document (ou un bloc de Document), l’entrée est saisie dans bloc de document avec un curseur penché : / | En mode Document (ou un bloc de Document), l’entrée est saisie avec un curseur droit, comme un texte plein :  : | entrer le texte source |
| En Mode Worksheet, la saisie est faite là où il y a une invite d’entrée avec un curseur penché | En Mode Worksheet, la saisie est faite là où il y a une invite d’entrée avec un curseur droit |
| Pour convertir une expression **2-D Math** en une expression **1-D Math** cliquer sur l’expression avec le bouton droit de la souris et sélectionner :  2-D Math 🡪 Convet-to 🡪 1-D Math | Pour convertir une expression **1-D Math** en une expression **2-D Math** cliquer sur l’expression avec le bouton droit de la souris et sélectionner :  1-D Math 🡪 Convet-to 🡪 2-D Math |
| Auncun symbol de terminaison n’est nécessaire | Toutes les entrées doivent se terminer par ; ou : |
| Les palettes rendent la saisie des expressions dans un notation familière plus facile que la saisie dans une syntaxe étrengère et réduisent les erreurs de saisie | L’utilisation des palettes en mode 1-D Math nous renseigne sur la syntaxe des commandes correspondantes. |

**Palettes**

Les palettes sont des collections d’éléments relies qu’on peut insérer dans un document by clicking or drag-and-dropping. L’environnement Maple donne accès à plus de 30 palettes contenant des éléments tels que les symboles, modèles, opérations mathématiques et plus encore.

**Voir et Arranger les Palettes**

Par défaut, les palettes sont affichées dans les bans de palettes à droite et à gauche de la fenêtre Maple. Pour voir et gérer les palettes et les bans de palettes :

**Entrer une Expression en Utilisant les Palettes**

Soit l’expression suivante qu’on veut obtenir avec Maple :

Dans cet exemple, on va saisir l’expression à gauche (en utilisant les palettes) puis l’évaluer.

**Commandes**

Bien Maple offre plusieurs éléments pour résoudre les problèmes et manipuler les résultats sans entrer aucune commande, on pourrait trouver plus commode d’avoir plus de contrôle et de flexibilité en utilisant l’ensemble des commandes et le langage de programmation que Maple offre.

**La Bibliothèque Maple**

Les commandes sont contenues dans la bibliothèque Maple, qui est divisée en deux groupes : la bibliothèque principale et les packages. La bibliothèque principale contient les commandes Maple les plus fréquemment utilisées. Les packages contiennent des commandes en relation pour exécuter des taches dans disciplines telles que les Statistiques, la Géométrie Différentielle ..etc. Par exemple, le package Optimisation contient les commandes pour résoudre numériquement des problèmes d’optimisation.

**Entrée des Commandes**

Si on veut interagir avec Maple en utilisant les commandes, alors il faut simplement entrer les en utilisant 2-D Math. Les commandes et variables sont affichées en italique. Les commandes Maple sont construites dans un format similaire à celui de *commande(arguments)*, basé sur la commande utilisée.

**Package de Commandes**

Il y a deux manières d’accéder aux commandes dans un package, en utilisant la forme longue de la commande package ou bien la forme courte.

**La Forme-Longue d’accès aux Commandes de Package:**

La forme longue spécifie à la fois le nom du package et celui de la commande en utilisant la syntaxe package[commande](arguments).

**La Forme-Courte d’accès aux Commandes de Package:**

La forme courte rend toutes les commandes du package disponibles en utilisant la commande , . Si on veut utiliser un certain nombre de commandes d’un package, charger le package en entier est recommandé. Lorsqu’on exécute la commande , une liste de toutes les commandes dans le package est affichée. Pour supprimer l’affichage des noms des commandes, terminer la commande par ":". Comme alternative, on peut charger les packages à travers le menu Outils, en sélectionnant Charger un Package, puis le nom du package.

Aprés le chargement d’un package, on peut utilizer les noms de la forme-courte, c.à.d. les noms des commandes sans le nom du package.

**Les Etiquettes d’Equations**

Les étiquettes d’équations aident à économiser le temps de saisie des expressions en référençant les sorties de Maple. Par défaut, les étiquettes des équations sont affichées. Si elles ne sont pas affichées, alors

1. A partir du menu Outils, sélectionner Options, et cliquer sur le volet Affichage. S’assurer que boite "Afficher les étiquettes des équations" est cochée.

4. Press Enter to obtain the result.

**Pour changer le format des étiquettes des équations:**

• Sélectionner Format . Etiquettes d’Equations . Affichage des Etiquettes. Dans le dialogue Format des étiquettes,

sélectionner un des schémas de l’étiquette de numérotation.

• Optionnellement, saisir un préfixe de numérotation d’étiquettes.

L’élément de menu "Référence d’étiquettes" permet de basculer entre le nom d’étiquette et le contenu de sa référence. Placer le curseur sur l’étiquette d’équation rérérencée et sélectionner : Format . Etiquettes d’équations. Références d’étiquettes.

L’étiquette est associée à la dernière sortie dans un groupe d’exécution.

On ne peut pas appliquer les étiquettes d’équations à ce qui suit :

• Erreur (Error), Avertissement (warning), et les messages d’information

• Tables, images, traces, Feuilles, dessins

**2.3.3. Blocs de Document**

En mode Document, le contenu est créé comme une série de blocs de document. Les blocs de Document permettent de cacher la syntaxe utilisée pour effectuer les calculs, ce qui en retour permet de se concentrer sur le concept présenté au lieu de la commande utilisée pour manipuler ou résoudre le problème. On peut aussi créer des blocs de document en mode Feuille de calcul pour arriver à la même fin. Les blocs de document sont typiquement réduits pour cacher le code Maple, mais ces régions peuvent être élargies pour révéler ce code.

**Pour créer un bloc document :**

Depuis le menu Format, sélectionner Créer Bloc de Document. Si un ou plusieurs groupes d’exécution est sélectionné, alors un bloc de document est créé contenant ces groupes d’exécution. Sinon, un nouveau bloc de document est créé après le groupe d’exécution courant (voir exemple).

Les régions des blocs de document sont identifiées par des balises localisées dans la barre verticale le long du volet gauche du document (voir figure). En plus des bordures des blocs de documents, ces balises (icones) indiquent la présence d’attributs cachés dans de document tels qu’annotations, formatage numérique, renvois.

**Pour active les balises:**

A partir du menu Affichage, sélectionner Balises.

Pour voir le code dans un bloc de document :

1. Placer le curseur dans un bloc de document à développer.

2. A partir du menu Affichage , sélectionner Développer Bloc de Document.

Avec le Bloc de Document développé, on peut voir la commande Maple qui a été utilisée pour faire le calcul. A noter aussi l’invite rouge (>) avant l’expression originale et la commande de résolution. La saisie de commandes en dehors de la région d’un bloc de document est faite à cette région d’entrée. Pour insérer une région d’entrée, cliquer sur le bouton dans la barre d’outils.

**Pour faire disparaître un Bloc de Document :**

• Avec le curseur à l’intérieur du bloc de document, sélectionner "Affichage. Réduire Bloc de Document".

On peut utiliser ce processus de développement des blocs de document pour voir et éditer les commandes Maple à l’intérieur d’un bloc de document.

**Changer l’Affichage:**

On peut spécifier quelles parties de l’entrée ou la sortie sont affichées lorsque le bloc de document est réduit. Pour chaque groupe d’exécution dans le bloc, on peut choisir d’afficher l’une ou l’autre de l’entrée ou la sortie.

• Placer le curseur dans le groupe d’exécution.

• A partir du menu Affichage, sélectionner Basculer l’affichage de l’entrée/sortie.

On peut choisir, aussi, d’afficher la sortie dans la ligne ou bien centrée sur une nouvelle ligne.

• A partir du menu Affichage, sélectionner Sortie dans la ligne.

**Création d’un Bloc de Document dans le Mode Feuille de Calcul**

En mode Feuille de Calcul, on peut créer un contenu en utilisant des commandes, et puis utiliser un bloc de document pour choisir la quantité d’information à afficher. Saisir la phrase suivante en utilisant l’entrée/sortie Texte et 2-D Math :

|  |  |  |
| --- | --- | --- |
| 1. A partir de l’invite d’entrée utilisateur cliquer sur  l’icône pour saisir du texte plein.  Enter "La réponse à".  **Note**: ces instructions sont pour le mode Feuille de Calcul. |  |  |
| 2. Cliquer l’icône de l’invite d’entrée, , pour saisir les commandes Maple. Saisir , puis appuyer sur **Enter** pour exécuter la commande.  [> |  |  |
| 3. Encore une fois cliquer sur l’incône Texte pour saisir le reste du texte, "est", puis saisir une autre invite d’entrée.  S’assurer de mettre des blancs tout autour du texte pour la phrase s’affiche proprement. |  |  |
| 4. Pour afficher, encore, la même sortie, utiliser la commande **value** et avec l’étiquette d’équation. Cela permet s’insérer du texte entre l’entrée et la sortie d’une seule commande, il y a réellement deux commandes.  Saisir et exécuter la commande comme indiqué. |  |  |
| 5. Pour finir la phrase, cliquer sur l’icône de texte dans le dernier groupe d’exécution, puis saisir un point. |  |  |
| 6. Sélectionner la phrase entière, puis à partir du menu **Format** sélectionner **Créer un Bloc de Document**. Par défaut seuls te texte et la sortie restent visibles, la sortie est centrée sur une nouvelle ligne. |  |  |
| 7. Pour afficher le texte et la sortie sur la même ligne, placer le curseur dans le bloc de document ; à partir du menue **Affichage** sélectionner **Sortie dans la ligne**. |  |  |
| 8. Si on veut afficher l’entrée au lieu de la sortie pour la première expression ; placer le curseur devant la première expression et à partir du menu **Affichage** sélectionner **Basculer l’affichage de l’entrée/sortie.** Seule la première région affiche l’entrée. |  |  |

**2.4. Calculs de Base**

**2.4.1. Calcul Symbolique et Numérique**

Le calcul symbolique est la manipulation mathématique des expressions impliquant des quantités symboliques ou abstraites, telles que les variables, fonctions, et opérateurs, et des nombres exactes, tels ques les entiers, rationnels,

, et . Le but d’une telle manipluation pourrait être de transformer une expression en une forme plus simple ou bien de relier l’expression à d’autres formules mieux comprises.

Le calcul numérique est la manipulation des expressions dans le contexte de l’arithmétique à précision finie. Les expressions impliquant des nombres exacts, par exemple, sont remplacées par des approximations utilisant des nombres à virgule flottante, par exmple, 1.41421. Ces calculs impliquent en général des erreurs. Comprendre et contrôler ces erreurs est souvent d’une importance comparables à celle des résultats calculés.

En Maple, le calcul numérique est normalement effectué si on utilise des nombres en virgule flottante (nombres contenant un point décimal) ou bien la commande **eval**. La commande de tracage utilise le calcul numérique, alors que les commandes telles que int, limit, et gcd, ainsi que la résolution de problèmes mathématiques utilisent le calcul symbolique pour arriver à leurs résultats.

**Calculs Exacts**

En Maple, les entiers, rationnels, les constantes mathématiques tels que et , ainsi que les strucutres mathématiques, telles que les matrices, avec ces nombres comme entrées sont traités comme des quantités exactes. Les noms, tels que et les fonctions mathématiques, telles que et sont des objets symboliques. Les noms peuvent avoir comme valeurs des quantités exactes, et les fonctions peuvent être évaluées pour des arguments aussi bien exacts ou symboliques.

**Important**: Sauf si on lui demande de faire autrement (voir section suivante), Maple evalue les expressions contenant des quantités exactes en des résultats exacts, comme si on devait effectuer les calculs à la main, et non pas en des approximations, comme on les obtiendrait à l’aide d’une calculatrice standard.

**Calculs en virgule flottante**

Dans certaines situations, une approximation numérique d’une quantité exacte est nécessaire. Par exemple, la commande de tracage exige que l’expression qu’elle trace soit évaluée avec des valeurs numériques qui peuvent être rendue sur écran : ne peut pas être rendue, mais 3.14159 peut l’être. Maple distingue les quantités exactes de celles approximées grâce à la présence ou l’absence du point décimal : 1.9 est approximer, alors que est exact.

**Note**: Une représentation alternative des nombres en virgule flottante, appelée e-notation, pourrait ne pas inclure un point décimal explicite : .

En présence d’une quantité en virgule flottante (approximer) dans une expression, Maple calcul généralement en utilisant une approximation numérique. L’arithmétique impliquant un mélange de quantiés exactes et en virgule flottante les résultats sont virgule flottante.

Si on passe un argument en virgule flottante à une fonction, elle tente normalement de produire une approximation en virgule flottante du résultat.

**Convertir Les Quantités Exactes en Qauntités en Virgule Flottante**

Pour convertir une quantité exacte en une approximation numérique de cette quantité, on utilise la commance **eval** ou bien l’opération **Approximate** du menu contexte.

Par défaut, Maple calcul de telles approximations en utilisant 10 chiffres arithmétiques. On peut modifier ceci de deux manières:

• Localemnt, on peut passer la précision, à l’appel **eval**, comme un indexe.

• Globalement, on peut paramétrer la valeur de la variable d’environnement **Digits**.

**Note**: Si approprié, Maple effectue les calculs en virgule flottante en utilisant directement le hardware sous-jacent de l’ordinateur.

**Sources d’erreur**

Par sa nature, le calcul en virgule flottante comporte normalement quelques erreurs. Contrôler l’effet de ces erreurs est le sujet de recherches actives en Analyse Numérique. Quelques sources d’erreur sont : sont des exemples.

• Une quantité exacte pourrait ne pas être représentable en forme décimale : et sont des exemples.

• Les petites erreurs peuvent s’accumuler aprés plusieurs opérations arithmétiques.

• La soustraction de quantités presque égales peut résulter en une information essentiellment non utile. Par exemple considérer le calcul : for .

Aucun chiffre correcte ne reste. Si, cependant, on utilise Maple pour analyser cette expression, et on remplace cette forme par une représntation plus précise pour de petites valeurs un résultat précis de 10 chiffres peut être obtenu.

**2.4.2. Résolution d’Equations et d’Inéquations**

En utilisant Maple, on peut résoudre symobiliquement des équations et des inéquations. On peut, aussi,résoudre numériquement des équations.

Pour résoudre une équation ou un ensemble d’équations en utilisant les menus de contexte :

1. Cliquer le bouton droit de la souris sur les équations.

2. A partir du menu de contexte, sélectionner Résoudre (ou Résoudre Numériquement).

En mode Worksheet, Maple insert une séquance d’appel qui résoud l’équation suivie des solutions.

Si on sélectionne résoudre, Maple calcul les solutions exactes.

Si on sélectionne Résoudre Numériquement, Maple calcul les solutions en virgule flottante.

**Résoudre Symboliquement les Equations et Inéquations**

La commande solve est un résolveur général qui determine les solutions exacte symboliques des équations ou inéquations. Les solutions d’une seule équation ou inéquation sont retournées comme une séquence d’expressions. Si Maple ne trouve aucune solution, la commande solve retourne une séquence d’expressions vide.

En général, solve calcul les solutions dans le champs de nombres complexes. Pour restriendre le problème aux seules solutions réelles on utilise le package RealDomain (DomaineRéel), pour forcer Maple à effectuer les calculs dans le champs des nombres réels.

Après le chargement du package RealDomain (DomaineRéel), Maple considère que toutes les variables sont réelles. Les commandes retournent des résultats simplifiés appropriés au champs des nombres réels.

**Verifier**

Il est recommandé de vérifier les solutions retournées par la commande solve en utilisant la commande eval.

**Attribuer la valeur d’une solution à une variable:**

Pour attribuer la valeur d’une solution à une variable correspondante comme une expression, on utilise la commande **assign**.

**2.4.3. Résoulution de Problèmes Mathématiques**

Ce chapitre se focalise sur la résolution de problèmes dans des disciplines mathématiques spécifiques.

**2..4.3.1 Algèbre Polynomiale**

Un polynome Maple est une expression en des puissances d’une inconnue. Les polynomes univarié sont des polynomes d’une inconnue, exemple,, . Les polynomes multivariés sont des polynomes de plusieurs inconnues, tel que . Les coefficients peuvent être des entiers, rationnels, irrationnels, des nombres en point-flottant, des nombres complexes, variables, ou une combinaison de ces types.

**Arithmétique Polynomiale**

Les opérateurs arithmétiques pour les polynomes sont les opérateurs arithmétiques standards de Maple sauf l’opérateur de division (/). L’opérateur de division accepte des arguments polynomiaux, mais n’effectue pas la division polynomiale. La division polynomiale est opération importante. Les commandes ***quo*** et ***rem*** trouvent le quotient et le reste d’un division polynomiale. Les commandes ***iquo*** et ***irem*** trouvent le quotient et le reste d’un division entière.

Pour développer un polynome, on utilise la commande ***expand***.

Si on veut déterminer si oui ou non un polynome divise au autre, mais qu’on ne s’intéresse pas au quotient, alors on peut utiliser la commande ***divide***.

La commande divide teste la division polynomiale exacte.

**Important**: On doit insérer un espace ou l’opérateur de multiplication (.) entre les noms de variables adjacentes ; Sinon, elles seront interprétées comme une seule variable.

|  |  |  |
| --- | --- | --- |
| Opération | Opérateur | Exemple |
| Addition |  |  |
| Soustraction |  |  |
| Multiplication |  |  |
| Division Quotient et Reste |  |  |
| Exponentiation |  |  |
| On peut spécifier la multiplicaton en saisissant directemment \* , qui s’affichera en 2-D Math comme .  En 2-D Math onp eut multiplier implicitement en placant un espace entre deux expressions.  Dans certains cas l’espace est optionnel, par exemple Maple interprète un nombre suivit par un nom comme un multiplication implicite.  En 2-D Math l’exponentiation s’affichier comme un accent circonflexe. | | |

**Trier les Termes**

Pour trier les termes d’un polynome, on utilise la commande ***sort***.

Pour spécifier les inconnues du polynome et leur ordre, on inclut une liste de noms.

Par défaut, la commande sort trie un polynome par ordre décroissant du total des degrés des termes.

Le premier terme a un total des degrés égal à 4. Les deux autres termes ont un total de 3. L’ordre des deux derniers termes est déterminé par l’ordre de leurs noms dans la liste.

Pour trier les termes par ordre lexicographique pur, c.à.d., en dimiuant l’odre de la première inconnue dans liste d’option, et puis en dimunuant l’odre de l’inconnue suivante dans la liste d’option, on spécifie l’option ‘plex’.

**Regrouper les termes**

Pour collecter les termes d’un polynome, on utilse la commande ***collect***.

**Coefficients et Degrés**

Maple a plusieurs commandes qui retournent les valeurs des coefficients et degrés d’un polynome.

|  |  |  |
| --- | --- | --- |
| Commande | Description | Exemple |
| coeff | Coéfficent du terme de degré spécifié |  |
| lcoeff | Le coéfficient de tête |  |
| tcoeff | Le coéfficient de queu |  |
| coeffs | Séquence de tous les coéfficients avec correspondance avec les termes un à un |  |
| degree | Degré (Le plus haut) |  |
| ldegree | Degré le plus bas |  |

**Factorisation**

Pour exprimer un polynome dans une forme totalement factorisée, on utilse la commande ***factor***.

**2.4.3.2. La Commande limit**

Par défaut, Maple recherche la limite bidirectionnelle. Pour spécifier la direction, inclure l’une des options left,right, ou complex dans un appel de la commande limit.

La commande limit peut aussi calculer des limites multidimentionnelles.

**Calcul Numérique d’une limite**

Pour calculer numériquement une limite :

• Utiliser la séquence d’appel evalf(Limit(arguments)).

**Important**: Utiliser la commande inerte Limit, non pas la commande limi. La commande Limit accepte les mêmes arguments que ceux de la commande limit.

Par exemple:

La commande Limi ne calcul pas la limit. Elle retourne une limite non évaluée.

**2.4.3.3. La Commande diff**

Maple calcul les dérivées en utilisant la commande ***diff***. Pour utiliser directement la commande ***diff***, spécifier l’expression à dériver et la variable.

On peut calculer des dérivées d’ordre supérieur en spécifiant une séquence de variables dérivation. Maple appelle récursivement la commande ***diff*** .

Pour calculer des dérivées partielles, on utilise la même sntaxe. Maple que les dérivées commutent.

Pour saisir des dérivées d’ordre plus grand, il préférable d’utiliser la syntaxe diff(f,x$n). Cette syntaxe peut être utilisée pour calculer la dérivée symbolique d’odre n.

Par exemple:

**2.4.3.4. Les Series**

Pour générer le développemnt en séries de Taylor d’une fonction en point, on utilise la commande ***taylor***.

**Note:** Si les séries de taylor n’existent pas, on utilise la commande series pour trouver un dévelopement en séries général.

Par exemple, la fonction intégrale de cos n’a pas de développement en séries de taylor en 0.

Pour générer un dévelopement en séries tronqué d’une fonction en un point, on utilise la commande series.

Par défaut, Maple effectue les calculs jusqu’à l’ordre 6. Pour utiliser un ordre différent, on spécifie un entier non négatif dans le troisème argument.

Pour paramétrer l’ordre pour tous les calculs, on utilise la variable d’environnement **Order**.

**2.4.3.5 La Commande int**

and utilisent la commande ***int***. Pour utiliser cette commande directement, spécifier les arguments suivants :

• Expression à intégrer

• Variable d’intégration

Pour une intégrale définie, on initialise la variable d’intégration avec l’intervalle d’intégration.

**Intégration Numérique**

Pour efectuer un intégration numérique :

• Utiliser la séquence d’appel ***evalf(Int(arguments))***.

**Important**: Utiliser la commande inerte Int, non pas la commande int.

En plus des arguments acceptés par la commande int, on peut inclure des arguments optionnels tel que la method, qui spécifie la méthode d’intégration numérique.

**2.4.4. Programmation Basique**

Jusqu’ à présent nous avns utiliser Maple de manière inteeractive, en effectuant des opérations de manière séquentielle, comme l’exécution d’un seule commande. Maple offre aussi constructions sophistiquées pour la programmation.

En Maple, on peut écrire des programes appelés procédurs, et les sauvegarder dans des modules. Ces modules peuvent être utilisés et distribués de la même manière que les packages.

**Important**: Il est fortement consillé d’utiliser le mode Worksheet et le mode de saisie 1-D Math lorsqu’on programe ou utilise les commandes de programmation. Par conséquent, dans ce qui suit les saisies seront faites en 1-D Math.

**2.4.4.1. Strucutres de Control**

Deux strucutres de control de base dans Maple sont l’instruction if, qui controle l’exécution conditionnelles d’un séquence d’instructions, et l’instruction for, qui controle l’exécution répétée d’une séquence d’instructions.

**2.4.4.2 Exécution Conditionnelle (Instruction if)**

On peut spécifier que Maple effectue une action seulement si une condition est vérifiée. On peut aussi effectuer une action parmi un ensemble de plusieurs actions, selon laquelle des conditions est satisfaite.

En utilisant l’instruction if, on peut exécuter une instruction parmi une série d’instructions sur la base d’une condition booléene(vrai, faux, ou FAIL). Maple teste chaque condition dans l’ordre. Lorsqu’une condition est satisfaite, Maple exécute l’instruction correspondante, et puis quitte l’instruction if.

**Syntaxe**

L’instruction if a la syntaxe suivante en Maple:

Les expressions conditionnelles (conditional\_expression1, conditional\_expression2, ...) peuvent être n’impporte quelle expression boléenne. On peut construire les expressions booléenne en utilisant :

• Les opérateurs relationnels : <, <=, =, >=, >, <>

• Les opérateurs logiques : and, or, xor, implies, not

• Les noms logiques : true, false, FAIL

La séquences d’instructions (statement\_sequence1, statement\_sequence2, ..., statement\_sequenceN) peuvent être n’importe quelle instruction Maple, l’instruction if incluse.

La clause elif est optionnelle. On peut en spécifier n’importe quel nombre.

La clause else est optionnelle.

**Instructions if Simples**

L’instruction if la plus simple a une seule expression conditionnelle seulement.

Si l’expression conditionnelle est évaluée à true, la séquence d’instructions est exécutée. Sinon, Maple quitte immédiatemment l’instruction if.

Exemple:

**Clause elses**

Dans une instruction if simple avec une clause else, si l’évaluation des expressions conditionnelles retourne false ou FAIL, Maple exécute la séquence d’instructions dans la clause else.

Exemple :

**Clauses elif**

Dans une instruction if avec des clauses elif, Maple évalue les expressions conditionnelles dans l’ordre jusqu’à ce que l’une d’elles retourne true. Maple exécute la séquence d’instructione correspondante, et quitte l’instruction if. Si aucune évaluation ne retourne true, Maple quitte l’instruction if.

**L’ordre des clauses elif :** Une séquence d’instructions d’une clause elif n’est exécutée que lorsque l’évaluations des expressions conditionnelles précédentes retourne false ou FAIL, et l’évaluation de sa condition conditionnelles retourne true. Ceci veut dire que le changement de l’ordre des clause elif pourrait changer le comportement de l’instruction if.

Dans ce qui suit l’ordre des clauses elif faux.

**Clauses elif et else**

Dans un instruction avec des clauses **elif** et **else**, Maple évalue les expressions conditionnelles dans l’ordre jusqu’à ce qu’une d’elles retourne true. Maple exécute la séquence correspondante, et quitte le **if**. Si aucune évaluation ne retourne true, Maple exécute la séquence de la clause **else**.

**2.4.4.3. Repetition (Boucle for)**

En utilisant l’instruction de répétition, on peut exécuter de manière répétée une séquence d’instructions. On peut répéter les instructions de trois manières.

• Jusqu’à ce que la valeur d’une variable cmpteur dépasse une limite (boucle for/from)

• Pour chaque opérande d’une expression (boucle for/in)

• Jusqu’à ce qu’une condition booléenne ne tienne plus (boucle while)

**Boucle for/from**

La boucle for/from répéte une séquence d’instructions jusqu’à ce que la valeur d’une variable cmpteur dépasse une limite.

**Syntaxe**

**Exemple**

**Boucle for/in**

La boucle for/in répéte une séquence d’instructions pour chaque composant (opérande) d’une expression, par exemple, les éléments d’une liste.

**Syntaxe**

The for clause must appear first.

**Exemple**

La boucle suivante retourne l’approximation en virgule flottante de la fonction sinus des angles (en degré) de la liste L.

**2.4.4.4. Boucle while**

La boucle while répéte une séquence d’instructions jusqu’à ce qu’une condition booléenne ne tienne plus.

**Synta****xe**

Une boucle while répéte jusqu’à ce qu’une son expression booléenne conditional\_expression retourne false ou FAIL.

**Exemple**

La boucle suivante calcul les chiffres de 872,349 en base 7 (dans l’ordre croissant de la signification).

**2.4.4.5. Boucles Générales**

On peut inclure une boucle while dans une boucle for/from ou for/in.

La boucle for/from générale a la syntaxe suivante.

La boucle for/in générale a la syntaxe suivante.

Après le teste de la condition de limite au début de chaque itération de la boucle, Maple évalue conditional\_expression.

• Si conditional\_expression est évaluée à false ou FAIL, Maple quitte la boucle.

• Si conditional\_expression est évaluatée à true, Maple exécute statement\_sequence.

**2.4.5. Commandes Itératives**

**Création d’une Séquence**

La commande seq crée un séquence de valeurs en évaluant une expression spécifiée sur une plage de valeurs d’indexes ou les opérandes d’une expression.

**seq(***expression***,** *name* **=** *initial* **..** *final***);**

**seq(***expression***,** *name* **in** *expression***);**

**Ajouter et Multiplier des Expressions**

Les commandes add et mul ajoutent et multiplient des séquences d’expressions sur une plage de valeurs d’indexes ou d’opérandes d’une expression.

**add(***expression***,** *name* **=** *initial* **..** *final***);**

**mul(***expression***,** *name* **=** *initial* **..** *final***);**

Les limites de la plage d’indexes (initial and final) dans add et mul doivent être évaluées à des constantes numériques. Pour les calculs symbolique on utilise les commandes sum et product.

**Exemple :**



**> **

**Sélectionner les Opérandes d’une Expresson**

Les commandes **select**, **remove**, et **selectremove** appliquent une procédure à valeur booléenne ou une commande aux opérandes d’une expression.

• La commande select retourne les opérandes pour lesquels la procédure or commande retourne true.

• La commande remove retourne les operandes pour lesquels la procédure or commande retourne false ou FAIL.

• La commande selectremove retourne deux expressions de même type que l’expression d’entrée.

- La première consiste en les operandes pour lesquels la procédure or commande retourne true.

- La seconde consiste en les operandes pour lesquels la procédure or commande retourne false.

La structure de la sortie est la même que la strucutre de l’entrée.

**select(***proc\_cmd***,** *expression***);**

**remove(***proc\_cmd***,** *expression***);**

**selectremove(***proc\_cmd***,** *expression***);**

**La commande map sur un Set ou List**

La commande map applique un nom, procédure, ou commande à chaque élément d’un ensemble ou liste.

**map(***name\_proc\_cmd***,** *expression***);**

**La commande map binaire sur deux Listes or Vecteurs**

La commande zip applique un nom ou une procédure binaire ou commande à deux liste ou vecteurs par pair de composants.

Par défaut, la longueur de l’objet retourné est celle de la liste ou vecteur le plus court. Si on spécifie une valeur comme le (optionnel) quatrième argument, elle est utilisée comme valeur des éléments manquants de la liste ou vecteur la plus court(e). Dans ce cas, la longueur de la valeur retournée est celle de la liste ou vecteur le plus long(ue).

**zip(***proc\_cmd***,** *a***,** *b***);**

**zip(***proc\_cmd***,** *a****,*** *b***,** *fill***);**

**2.4.6. Procédures**

Une procédure Maple est un programme qui consiste en des instructions Maple. En utilisant les procédures, on peut rapidemment exécuter les séquences d’instructions contenues.

**Définir et exécuter des procédures simples**

Pour définir une procédure, enfermer une séquence d’instructions entre les instructions ***proc()*** et ***end proc***. En général, on attribue une définition de procédure à un nom.

La procédure suivante retourne la recine carrée de 2.

**Note**: Maple retourne la définition de la procedure

Pour améliorer la lisibilité des procédures, il est recommendé de définir une procédure sur plusieurs lignes, et faire des indentations avec des espaces. Pour commencer une nouvelle ligne (sans évaluer la définition incomplète de la procédure), appuyer sur **Shift+Enter**. Lorsqu’on termine la saisie de la procédure, on appuie sur **Enter** pour créer la procédure.

**Exemple:**

Pour exécuter la procédure p, on saisi son nom suivi des parenthèses ( ).

**Procedures avec paramètres**

On peut définir des procédures qui acceptent des paramètres. Entre les parenthèses de l’instruction proc, on spécifie les noms des paramètres, séparés par des virgules.

Lorsque l’utilisateur exécute la procédure, les noms des paramètres sont remplacés par les valeurs des arguments.

**Valeurs Retournées par une Procédure**

Lorsqu’une procédure est exécutée, Maple retourne seulement la valeur du résultat calculé de la dernière instruction. Maple ne retourne pas la sortie de chaque instruction de la procédure. C’est sans importance d’utiliser " ;" ou ":" comme séparateus d’instructions.

**2.4.7. Modules**

Les procédures Maple associent une séquence de commandes avec une seule commande. Le module, une strucure de programmation plux complexe, permet d’associer des procédures et données en relation.

Un élément clé des modules est qu’ils exportent des variables. Ce qui implique que les variables sont disponibles en dehors du module dans lequel elles sont céées. La plus part des packages Maple sont implémentés comme des modules. Les commandes du package sont des exportations du module.

**2.4.8. Objects**

Les objets étendent l’idée d’associer des procédures et données au delà de ce que les modules permettent. Avec les objets, plusieurs instances d’une classe d’objets peuvent être créées. Chaque objet individuel peut avoir ses propres données, bien qu’il puisse partager d’autres valeurs et procédures avec l’ensemble de objets de la classe. Une classe Maple bien implémentée peut être utilisée aussi naturellement qu’un type propre de Maple.