# ÉCRITURE ET LECTURE DE FICHIERS DE DONNÉES

Maxima peut écrire des données dans des fichiers qui peuvent ensuite être lus soit par Maxima soit par d'autres applications. De même, il peut lire des données provenant de fichiers créés par d'autres applications.

La documentation concernant l'écriture et la lecture de jeux de données se trouve dans le manuel de Maxima :

https ://maxima.sourceforge.io/ext/maxima.pdf

chapitre 81, package numericalio. Ce package de Maxima est défini ainsi : « numericalio est un ensemble de fonctions permettant de lire et d'écrire des fichiers et des flux. Les fonctions d'entrée et de sortie en texte simple peuvent lire et écrire des nombres (entiers, flottants ou bigfloat), des symboles et des chaînes de caractères. Les fonctions d'entrée et de sortie en mode binaire ne peuvent lire et écrire que des nombres en virgule flottante.

Ce tutoriel traite des entrées et sorties en texte simple pour les nombres et les chaînes de caractères. Il est organisé comme suit.

La première section montre comment créer un ensemble de données à l'aide de deux commandes de Maxima, makelist et cons (pour « construire »).

La section suivante montre comment écrire les données dans un fichier.

La troisième et dernière section montre différentes façons d'accéder aux données à partir d'un fichier. Le fichier de données consulté peut avoir été créé précédemment avec Maxima, ou bien avoir été créé avec une autre application.

Certaines des commandes ci-dessous sont précédées de commentaires. Les commentaires placés dans des cellules contenant des entrées et leurs sorties doivent être contenus entre des délimiteurs de ce type : /* */. Pour placer des commentaires dans des cellules séparées comme celle-ci, au-dessus ou au-dessous de cellules contenant des entrées et leurs sorties, placez le curseur avant la cellule puis faites :

- cliquer sur le bouton texte dans la palette de boutons Cellules si elle est affichée
- sélectionner Cellule/Insérer une cellule de texte/Insérer une cellule de texte, ou
- utiliser le raccourci CTRL+1.

## 1 Création d'un tableau de données

Les commandes ci-dessous génèrent un ensemble de six valeurs pour chacune de trois variables. Les variables sont des entiers. La variable x varie de 4 à 9, et nous générons deux fonctions de la variable x : le carré de x et la factorielle de x.

Les listes L1, L2 et L3 contiennent les valeurs numériques correspondantes. Elles sont créées avec la commande makelist de Maxima.Ensuite des noms sont ajoutés aux listes à l'aide de la commande cons de Maxima. Les listes avec leurs noms sont LN1, LN2 et LN3.

```
/* Create three lists */                    L1 : makelist(i, i, 4, 9);
L2 : makelist(i^ 2, i, 4, 9);                L3 : makelist(i !, i, 4, 9);
/* Rajout de noms aux listes ' data */       LN1 : cons("Variable x", L1);
LN2 : cons("x carré", L2);                   LN3 : cons("x factorielle", L3);
```

(L1) $[4\,,5\,,6\,,7\,,8\,,9]$

(L2) $[16\,,25\,,36\,,49\,,64\,,81]$

(L3) $[24\,,120\,,720\,,5040\,,40320\,,362880]$

(LN1) $[\text{"}Variablex\text{"}\,,4\,,5\,,6\,,7\,,8\,,9]$

(LN2) $[\text{"}xcarr\text{"}\,,16\,,25\,,36\,,49\,,64\,,81]$

(LN3) $[\text{"}xfactorielle\text{"}\,,24\,,120\,,720\,,5040\,,40320\,,362880]$

Les listes sont maintenant utilisées pour générer un tableau. La création d'un tableau standard (colonnes de données verticales) se fait en deux étapes simples. D'abord, regrouper les trois listes dans une matrice à l'aide de la commande matrix de Maxima. Ensuite, transposer la matrice afin que les valeurs des variables apparaissent en colonnes. Donner un nom à la matrice n'est pas indispensable. On pourrait la laisser sans nom et créer la transposée avec la commande transpose(%), tant qu'aucune autre opération ne soit effectuée entre la création et la transposition de la matrice.

```
->  /* Mettre les données dans une matrice */                M : matrix(LN1,LN2,LN3)$
    /* Transposer la matrice M pour générer une table T */    T : transpose(M)$
    print ("Matrice M, composée de trois listes : ", M)$
    print ("Table T, créée en transposant  M : ", T)$
```

"Matrice M, composée de trois listes : "

$$\begin{bmatrix} "Variablex" & 4 & 5 & 6 & 7 & 8 & 9 \\ "xcarr" & 16 & 25 & 36 & 49 & 64 & 81 \\ "xfactorielle" & 24 & 120 & 720 & 5040 & 40320 & 362880 \end{bmatrix}$$

"Table T, créée en transposant M : "

$$\begin{bmatrix} "Variablex" & "xcarr" & "xfactorielle" \\ 4 & 16 & 24 \\ 5 & 25 & 120 \\ 6 & 36 & 720 \\ 7 & 49 & 5040 \\ 8 & 64 & 40320 \\ 9 & 81 & 362880 \end{bmatrix}$$

## 2 Écriture du tableau de données dans un fichier

Cette section montre comment écrire la table T dans un fichier nommé datamatrix.csv. Ce fichier est placé dans un dossier dont il faut donner le nom et le chemin complet. Pour ce document, le dossier sera :

/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data

pour mon ordinateur sous Linux. Pour windows, on utilisera par exemple C :/WXMAXIMADATA comme chemin. Dans tous les cas, cette commande doit être modifiée afin d'indiquer à wxMaxima où le fichier doit être enregistré avant l'exécution de la commande.

Pour une présentation des différentes options concernant la spécification des dossiers de données, voir Woollett, Maxima by Example, chapitre 2, sur csulb.edu/∼woollett/.

Avant d'exécuter la commande write_data, il est conseillé d'exécuter d'abord la commande file_search qui la précède. Cette commande indique si le fichier existe déjà. Si un fichier portant ce nom existe, on peut modifier le nom du fichier à enregistrer. Si aucun fichier de ce nom n'existe, Maxima renvoie le message : false.

Utilisateurs Windows : remarquez que le chemin du fichier utilise la barre oblique ( / ) et non la barre oblique inversée ( \ ) habituelle sous Windows. Autrement dit, Maxima suit la convention Unix à ce sujet.

*Avertissement* : remplacez C :/WXMAXIMADATA par le chemin de votre dossier avant d'exécuter la commande ci-dessous.

```
->  file_search("/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv");
```

(% o44) $false$

La commande Maxima write_data transfère les données du tableau T vers le fichier datamatrix.csv. Le suffixe csv indique que le fichier doit contenir des variables séparées par des virgules (Comma-Separated Variables).

La commande write_data accepte deux ou trois arguments : le nom de l'objet à copier (le tableau T dans cet exemple) ; le nom du fichier vers lequel copier le contenu de l'objet, chemin et nom de dossier inclus ; et (optionnellement) le type de séparateur.

Si le troisième argument est omis, Maxima utilise un espace comme séparateur. Le séparateur peut être un espace, une virgule, une tabulation, une barre verticale ( | ) ou un point-virgule. Si un séparateur autre qu'une virgule est utilisé, le suffixe du nom de fichier devrait être autre chose que csv, par exemple txt ou dat. L'utilisation des fichiers csv présente de nombreux avantages. En particulier, tous les tableurs (Excel, OpenOffice Calc, Gnumeric, etc.) peuvent lire et écrire des fichiers csv.

*Rappel* : remplacez C :/WXMAXIMADATA par le chemin de votre dossier avant d'exécuter la commande ci-dessous.

```
->      write_data(T,"/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv",comma);
```

(% o45) done

## 3 Lecture de données à partir d'un fichier texte

La commande printfile confirme le contenu du fichier nouvellement créé. Si la commande avait été suivie du symbole dollar plutôt que du point-virgule, le contenu du fichier aurait été affiché mais le chemin et le nom du dossier n'auraient pas été affichés.

*Rappel* : remplacez /home/... par le chemin de votre dossier avant d'exécuter la commande ci-dessous.

```
(% i1)  printfile("/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv");
```

$$"Variable x","xcarr","xfactorielle"4,16,245,25,1206,36,7207,49,50408,64,403209,81,362880$$

(% o1)

"/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv"

Pour de nombreux usages, importer les données sous forme de tableau ou de liste est plus utile que de simplement les afficher à l'écran. Les commandes read_table et read_nested_list, exécutées ci-dessous, lisent les données dans Maxima sous les formes indiquées.

En pratique, bien sûr, on n'utilise généralement qu'une seule de ces deux commandes. Les deux commandes ci-dessous donnent un nom aux objets lus : T2 pour le tableau et LT4 pour la liste. Des noms plus explicites pourraient être utilisés. On pourrait aussi ne donner aucun nom.

*Rappel* : remplacez /home/... par le chemin de votre dossier avant d'exécuter la commande ci-dessous.

```
(% i13) T2 : read_matrix("/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv")$
        LT4 : read_nested_list("/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv")$
        print("Voici les données saisies en tant que table : ", T2)$
        print("Voici les mêmes données, saisies sous forme de liste : ",LT4)$
```

"Voici les données saisies en tant que table : "

$$\begin{bmatrix} "Variable x" & "xcarr" & "xfactorielle" \\ 4 & 16 & 24 \\ 5 & 25 & 120 \\ 6 & 36 & 720 \\ 7 & 49 & 5040 \\ 8 & 64 & 40320 \\ 9 & 81 & 362880 \end{bmatrix}$$

"Voici les mêmes données, saisies sous forme de liste : "

[ [”$Variablex$” , ”$xcarr$” , ”$xfactorielle$”] , [4 , 16 , 24] , [5 , 25 , 120] , [6 , 36 , 720] , [7 , 49 , 5040] , [8 , 64 , 40320] , [9 , 81 , 362880] ]

Avant d'examiner une troisième méthode pour importer des données, exécutez la commande values. Elle montre ce qui a été ajouté à la mémoire de travail de Maxima. Le contenu de T et T2 est identique, mais Maxima ne le sait pas.

**(% i14)** values ;

(% o14) [T2 , LT4]

Une troisième option de lecture existe : la commande read_list lit les données sous forme d'une seule liste sans imbrication de listes. Autrement dit, les données ne sont pas imbriquées et ne sont pas placées dans un tableau.

*Rappel* : remplacez /home/... par le chemin de votre dossier avant d'exécuter la commande ci-dessous.

**(% i15)** read_list("/home/mgosse2/hubiC/logiciels/wxmaxima/documentation/data/datamatrix.csv") ;

(% o15)
[”$Variablex$” , ”$xcarr$” , ”$xfactorielle$” , 4 , 16 , 24 , 5 , 25 , 120 , 6 , 36 , 720 , 7 , 49 , 5040 , 8 , 64 , 40320 , 9 , 81 , 362880]