Maxima pour Calcul matriciel, manuel minimal

Nous utiliserons avant tout Maxima comme « calculatrice » mathématique et pas pour programmer; ensuite nous verrons.

1. Petit départ

1. Lorsqu'on lance la **console** Maxima elle affiche (%i1) (le i veut dire input), tapez une commande (par exemple: 1+5), ajoutez ; puis return.

Maxima exécute et affiche (%o1) (le o veut dire output) suivi du résultat : 6

Puis la console affiche (%i2) et attend la deuxième commande......

2. Maxima distingue majuscules et minuscules; faites attention.

3. Pour quitter vous tapez quit(); et return.

Opérations arithmétiques

1) + ,-, *, / (pour la division), ^ ou ** pour l'exponentiation, . pour la multiplication des matrices, sqrt(x) pour la racine carrée.

Les calculs se font sous forme exacte, c'est à dire que le résultat de l'opération 1/10+ 1/101 sera le nombre rationnel 201/10100 et pas la valeur approchée; si toutefois vous désirez une valeur approchée vous pouvez taper

bfloat(3/10); suivi de return

qui vous donnera une valeur approchée avec 16 chiffres significatifs

et si vous voulez décider le nombre de chiffres significatifs, voici la suite de commandes

fpprec:20; return

bfloat(3/10); return

vous donnera le résultat avec 20 chiffres significatifs...

2)

Comme Maxima privilégie la valeur exacte, voyez ce qui se passe si vous tapez

((1+sqrt(2))^5; return

il ne se passe rien

si vous voulez développer cette expression vous ajouterez ensuite

expand(%); return

ce qui développera le résultat précédent (% représente le résultat précédent) et si vous ajoutez

bfloat(%); return

vous obtiendrez une valeur approchée avec 20 chiffres significatifs.

3) Attention Maxima n'a pas besoin de bfloat pour effectuer de longs calculs; vérifiez-le en tapant

100!; return

4) créer une liste

(%i1) L:[5,8];return

5) pour récupérer un terme d'une lisite

(%i2) part(L,2); return

(%o2) 8

6) créer une liste de listes

(%i3) M:[[1,26],[3,5]];return

7) pour récupérer un terme

(%i4) part(M,2,2);return

(%o4) 5

Algèbre

Maxima est fait pour les calculs d'expressions

1) voici une suite à l'écran

(%i1) (x+2*y)^3;return

(%o1) (x+2*y)^3

(%i2) expand(%);return

(%o2) x^3 + 6*x^2*y +12*x*y^2+8y^3

si vous voulez affecter à y la valeur 5x, voici comment vous continuez

(%i3) %o2,y=5*x;return

(« %o2 » reprend le résultat qui porte ce nom, et la commande « y=5*x » affecte à y la valeur 8*x)

(%o3) 1331 x^3

2) si « expand » développe on peut aussi factoriser; par exemple

(%i3) factor(x^3 + 3*x^2*y +12*x*y^2+8y^3);return

donnera comme réponse

(%o3) (x+y)^3

Algèbre linéaire

1) Pour définir une matrice

(%i1) a:matrix([1,2,3],[4,8,-9]);return

va créer la matrice $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & -9 \end{pmatrix}$

et Maxima affiche

(%o1) $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & -9 \end{pmatrix}$

2) la multiplication des matrices

a.b

3) l'addition

a+b

4) le produit d'un réel par une matrice

3*a

5) le déterminant d'une matrice carrée

determinant(a)

6) l'inverse d'une matrice carrée inversible

invert(a)

7) la transposée

transpose(a)

Maxima pour (esssayer de) diagonaliser

(%i1)Pol:charpoly(M,t);

(%o1) Pol=....... (moche, on va simplifier)

(%i2) ratsimp(Pol);

(%o2) t^4-6t^2-8t-3

(%i3) solve(Pol,t);

(%o3) t=3, t=-1

On a les valeurs propres mais il nous faut les sous-espaces propres

Détermination des valeurs propres et des vecteurs propres

(%i4) eigenvectors(M);

la réponse est riche

(%o4) [[[3, - 1], [1, 3]], [[[1, 1, 1, 1]], [[1, 0, 0, - 1], [0, 1, 0, - 1], [0, 0, 1, - 1]]]]

détaillons:

les valeurs propres et leurs multiplicités

la première liste [[3, - 1], [1, 3]] détaille les valeurs propres avec les multiplicités

ici -1 est valeur propre de multiplicité 3; 3 est valeur propre de multiplicité 1

(on peut le vérifier $(t+1)^3(t-3)$ est bien la factorisation du polynôme caractéristique)

les bases des sous-espaces-propres

la seconde donne une famille libre de vecteurs propres :

[1, 1, 1, 1] est une base de ker(M-3I)

, [1, 0, 0, - 1], [0, 1, 0, - 1], [0, 0, 1, - 1] est une base de ker(M+I)

d'où une matrice de passage $P = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}$

et $P^{-1}\mathrm{MP} = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

Compléments

1) affecter une valeur à une variable (au passage si elle n'était pas encore définie cela la définit

(%i1) a:7;

(ùo1) 7

2) si vous voulez faire exécuter une opération mais ne pas afficher le résultat (plus loin vous l'utiliserez) remplacer le « ; » par le symbole dollar

3) de même si a est une matrice à m lignes et p colonnes on pourra modifier certains termes

par exemple:

(%i1) a:matrix([1,5,9],[2,-8,10]);

(%o1) $\begin{pmatrix} 1 & 5 & 9 \\ 2 & -8 & 10 \end{pmatrix}$

(%i2) for k thru 3 do (a[2,k]:a[2,k]-2*a[1,k]);a;;

(%o2) $\begin{pmatrix} 1 & 5 & 9 \\ 0 & -18 & -28 \end{pmatrix}$

au passage vous avez appris à écrire une « boucle for »

4)

(%i1) x:6; for k thru 5 do (x:x+1,x:x^2); if (x<17) then x else 21;

(%o1) 21

5) while (condition) do (instruction1,instruction2) ...