Characteristic cone for equation module

This program takes the generators of the cone and the partial difference equations of discrete autonomous *n*D system as inputs and gives a 'yes/no' certificate according to whether the given cone is a characteristic cone for the given system or not.

First the dimension of the system, a kernel representation matrix and the number of cone generating vectors are taken as input. Then the matrix and the vectors are given by the user. As output the program will give if the given cone is a characteristic cone or not.

The dimension of the system determines the number of free variables of the system. According to the input given by the user, the free variables are defined in x, as x1, x2, x3... . So the variables that are valid during matrix input are x1, x1^-1, x2, x2^-1, x3, x3^-1 ... . To make the input more readable, and reduce typo errors, user can use y variables instead of x inverses i.e. user can give input in form of x1, y1, x2, y2, x3, y3 ..., where y1, y2, y3... are considered as x1^-1, x2^-1, x3^-1.... Transposed column vectors are taken as input for generating the cone.
For example -c1 = 0 1 0 is actually the vector [0 1 0]^T

A few examples are shown below:

EXAMPLE 1:
- Dimension of the system: 3
- Dimension of the kernel representation matrix: (nxm): 3x2
- Number of vectors for generating the cone: 3

-Please enter the Kernel Representation matrix:
--> x3, x2^3
--> x2, y2*y3 -1
--> y1*y3, x1+x2

-please enter the transposed column vectors to define the cone:
-c1= 1 0 0
-c2= 0 1 0
-c3= 0 0 1

-This is a characteristic cone

Now in the next example we will use inverse of x, in places of y.

EXAMPLE 2:
- Dimension of the system: 3
- Dimension of the kernel representation matrix: (nxm): 3x2
- Number of vectors for generating the cone: 3

-Please enter the Kernel Representation matrix:
--> x2, x2^-1*x3^-1 -1
--> x3, x2^3

Authors: Ayan Sengupta, Mousumi Mukherjee and Debasattam Pal

--> x1^-1*x3^-1, x1+x2

-please enter the transposed column vectors to define the cone:
-c1= 1 0 0
-c2= 0 1 0
-c3= 0 0 1

-This is a characteristic cone

CAUTION:

As the implementation of the code depends on computing the Groebner basis of the module, there might be cases where the Groebner basis is too massive. In such cases computation gets too heavy and, SAGE or particularly, SINGULAR crashes after some time. Such an example is shown below:

EXAMPLE:

Dimension of the system: 3
Size of the kernel representation matrix: (nxm): 3x2
Number of vectors for generating the cone: 3

Please enter your Kernel Representation matrix:
Example: -> x1, x2, x3

-> x1^5*x3+x2^6+x3^7*x2, x3*y1 -1
-> x2, y2*y3 -1
-> y1*y3, x1+x2

please enter the transposed column vectors of the cone:
c1= 1 0 0
c2= 0 1 0
c3= 0 0 1

HOW TO RUN THE CODE IN SAGE:

First open SAGE from the terminal. Then you can load a file in a Sage session with the command "load".

EXAMPLE:

if your file is in the current directory where you started sage then:
load("your_file.sage")

if your file is in another directory then :
load("~/the_file_location/your_file.sage")

Authors: Ayan Sengupta, Mousumi Mukherjee and Debasattam Pal