Cyclic Vectors for Certain D-Modules

Faculty Mentor: Aron Heleodoro
Project Leader: Sungwoo Nam
IGL Scholars: Dhruv Kush, Tony Tan, Hantao Zhang, Ashwin Rajinish

May 15th, 2020

Definitions and Introduction:

Our project is Cyclic Vectors for Certain D-Modules, which leads to the question: What is a D-Module? Well, a D-Module is a module over the ring "D" of differential operators, and in this project we are primarily concerned with a left module over the Weyl Algebra.

A left R-Module is defined as follows: A left R-module is a set M with an addition operation + and a scalar multiplication operation .: RxM $--$ > M called such that (M,+) is an abelian group and that for all r and s in R and x and y in M we have r.(x+y) = r.x + r.y, (r+s).x = r.x + s.x, (rs).x = r.(s.x), and 1.x = x.

And the Weyl Algebra is simply the ring of differential operators with polynomial coefficients (over a field F of characteristic 0).

For instance an element in this ring looks like the following:

$$P = a_n(x_1, x_2...x_n)\partial^n + ... + a_1(x_1, x_2...x_n)\partial^1 + a_0(x_1, x_2...x_n)$$

Most importantly, we impose the commutator relation $[\partial_i, x_i] = \partial_i x_i - x_i\partial_i = 1$ which implies $[\partial_i, f] = \partial f/\partial x_i$.

In particular, our project aims to find an elementary proof of the statement that our D-module is cyclic.

Specifically, we aim to prove, using elementary techniques, that the D-module $F[x_1, ..., \ x_k][x_{ij}^{-1}]$ is generated by the element f^{-1} where $f = \prod_{0\leq i<j\leq k}(x_{ij})$.

Naive Approach:

One direction of attempt we tried was to work out a few examples and express the results as sums of certain elements in a matter similar to how elements in a finite-dimensional vector space are written as linear combinations of vectors of some basis and then find how the generator creates those "basis" elements. We noticed that, in a lot of cases, there are only a few options of operators we can hit our polynomial with, if it is to obey the symmetry of the element we are trying to generate. This led to our utilization of SageMath to create a library of python functions to generate different permutations of derivatives on a variable f. For example on $f = (x_1 - x_2)(x_2 - x_3)(x_1 - x_3)$, we can specify the parameters of f, the variables within f (in this case $x_1, x_2, x_3, x_{12}, x_{13}, and x_{23}$) and n to generate all permutations of derivatives up to n amount of derivatives.

Example Generations

$(dx_1)f^{-1} = -\frac{1}{x_{12}x_{13}^2x_{23}} - \frac{1}{x_{12}^2x_{13}x_{23}}$

$(dx_2)f^{-1} = -\frac{1}{x_{12}x_{13}x_{23}^2} + \frac{1}{x_{12}^2x_{13}x_{23}}$

$(dx_3)f^{-1} = \frac{1}{x_{12}x_{13}x_{23}^2} + \frac{1}{x_{12}x_{13}^2x_{23}}$

$(dx_1)(dx_1)f^{-1} = \frac{2}{x_{12}x_{13}^3x_{23}} + \frac{2}{x_{12}^2x_{13}^2x_{23}} + \frac{2}{x_{12}^3x_{13}x_{23}}$

$(dx_2)(dx_1)f^{-1} = \frac{1}{x_{12}x_{13}^2x_{23}^2} + \frac{1}{x_{12}^2x_{13}x_{23}^2} - \frac{1}{x_{12}^2x_{13}^2x_{23}} - \frac{2}{x_{12}^3x_{13}x_{23}}$

$(dx_3)(dx_1)f^{-1} = -\frac{1}{x_{12}x_{13}^2x_{23}^2} - \frac{1}{x_{12}^2x_{13}x_{23}^2} - \frac{2}{x_{12}x_{13}^3x_{23}} - \frac{1}{x_{12}^2x_{13}^2x_{23}}$

$(dx_1)(dx_2)f^{-1} = \frac{1}{x_{12}x_{13}^2x_{23}^2} + \frac{1}{x_{12}^2x_{13}x_{23}^2} - \frac{1}{x_{12}^2x_{13}^2x_{23}} - \frac{2}{x_{12}^3x_{13}x_{23}}$

$(dx_2)(dx_2)f^{-1} = \frac{2}{x_{12}x_{13}x_{23}^3} - \frac{2}{x_{12}^2x_{13}x_{23}^2} + \frac{2}{x_{12}^3x_{13}x_{23}}$

$(dx_3)(dx_2)f^{-1} = -\frac{2}{x_{12}x_{13}x_{23}^3} - \frac{1}{x_{12}x_{13}^2x_{23}^2} + \frac{1}{x_{12}^2x_{13}x_{23}^2} + \frac{1}{x_{12}^2x_{13}^2x_{23}}$

$(dx_1)(dx_3)f^{-1} = -\frac{1}{x_{12}x_{13}^2x_{23}^2} - \frac{1}{x_{12}^2x_{13}x_{23}^2} - \frac{2}{x_{12}x_{13}^3x_{23}} - \frac{1}{x_{12}^2x_{13}^2x_{23}}$

$(dx_2)(dx_3)f^{-1} = -\frac{2}{x_{12}x_{13}x_{23}^3} - \frac{1}{x_{12}x_{13}^2x_{23}^2} + \frac{1}{x_{12}^2x_{13}x_{23}^2} + \frac{1}{x_{12}^2x_{13}^2x_{23}}$

$(dx_3)(dx_3)f^{-1} = \frac{2}{x_{12}x_{13}x_{23}^3} + \frac{2}{x_{12}x_{13}^2x_{23}^2} + \frac{2}{x_{12}x_{13}^3x_{23}}$

Here is an example of the latex formatted from the library when $n = 2$.

For this specific case, we noticed many interesting terms such as:

$$-\frac{6}{x_{12}x_{13}^2x_{23}^3} - \frac{6}{x_{12}^2x_{13}x_{23}^3} - \frac{6}{x_{12}x_{13}^3x_{23}^2} + \frac{6}{x_{12}^3x_{13}x_{23}^2} + \frac{6}{x_{12}^2x_{13}^3x_{23}} + \frac{6}{x_{12}^3x_{13}^2x_{23}}$$

$$\frac{6}{x_{12}x_{13}^2x_{23}^3} + \frac{6}{x_{12}^2x_{13}x_{23}^3} + \frac{6}{x_{12}x_{13}^3x_{23}^2} - \frac{6}{x_{12}^3x_{13}x_{23}^2} - \frac{6}{x_{12}^2x_{13}^3x_{23}} - \frac{6}{x_{12}^3x_{13}^2x_{23}}$$

$$\frac{2}{x_{12}x_{13}^2x_{23}^3} + \frac{2}{x_{12}^2x_{13}x_{23}^3} + \frac{2}{x_{12}x_{13}^3x_{23}^2} - \frac{2}{x_{12}^3x_{13}x_{23}^2} - \frac{2}{x_{12}^2x_{13}^3x_{23}} - \frac{2}{x_{12}^3x_{13}^2x_{23}}$$

However, we were not able to follow implement the linear combination like algorithm to produce our wanted basis. Given a stronger computer algebra system/more time allocated towards the project, we hope to be able to examine some sort of pattern within basis terms of any f, and eventually uncover an elementary proof.

Here is the github link for those interested in the library:
`https://github.com/goldenxuett/Cyclic-vectors-for-certain-D-modules-IGL-2020`

Computer Algebra:

At some point, this problem turned into an algorithmic problem rather than a pure mathematical one. One might wonder whether there is an algorithm that finds a generator of a cyclic D-module. The answer is positive in some cases. A recent finding by Leytin[1] proves the existence of and provides an implementation for an algorithm that finds a generator for a holonomic left module M over the Weyl algebra given the annihilators L(a) and L(b) of two arbitrary elements a and b in M, where, for every x in M, L(x) is the set of elements in the underlying ring, in this case the Weyl algebra, sending x to 0 under left multiplication.

The algorithm uses Gröbner bases. A Gröbner basis is a specific kind of a generating set of an ideal in a polynomial ring over multiple indeterminates; this is a popular technique in computer algebra and computational algebraic geometry. It is rigorous and more efficient than taking powers of arbitrary elements in the module in brute-force fashion, which is both nondeterministic and unlikely to yield a solution in feasible time.

1. https://arxiv.org/pdf/math/0204303.pdf

Concluding Remarks

Unfortunately, given the circumstances and the state of the problem, we realized that the idea of an elementary proof existing could possibly be a dubious notion to begin with. After all, the amount of computation needed to prove that an element is a generator is unknown and the methodology to assuredly come about this proof is nebulous. Because of these constraints, developing a new proof is likely to involve algorithmic proficiency rather than pure theoretical proficiency. Future research with algorithms and tested techniques in computational algebraic geometry such as Gröbner bases seem promising as an alternative for brute force.

Acknowledgements

We thank Professor Aron Heleodoro and Sungwoo Nam for leading this project in these unfortunate circumstances. We also thank the Illinois Geometry Lab for providing this mathematically enriching opportunity.

References

https://arxiv.org/abs/1806.08754v2

https://arxiv.org/abs/1503.01055v1

https://arxiv.org/pdf/math/0204303.pdf

A primer of algebraic D-modules /