Proceedings of The National Conference
On Undergraduate Research (NCUR) 2020
Montana State University, Bozeman MT
March 26-28, 2020

Field Extensions in Mathematica

Giacomo Viazzo
Mathematics
Wingate University
Wingate, North Carolina 28174 USA

Faculty Advisor: Dr. Kaitlyn Perry

Abstract

Every ideal $I \subseteq k[x_1, \ldots, x_n]$ has a Gröbner Basis where k is a field. A Gröbner basis is a particular polynomial set that generates the ideal I. When using Mathematica[1] as a software to compute the Gröbner bases, we see that Mathematica only allows us to compute Gröbner bases of ideals when $k = \mathbb{Q}$. In this research, we develop a method so that we can work in $\mathbb{Q}(i)$, or more general $\mathbb{Q}(\sqrt{n})$ where $n \in \mathbb{R}$, while still using Mathematica. This allows for a broader use of the software as well as a significant simplification of the computations. We will introduce our method and discuss the reasons why it is logically valid to apply our method to all field extensions of $\mathbb{Q}$.

Key words: Gröbner Bases, Field Extensions, Mathematica

1. Introduction

When dealing with Gröbner bases, it is necessary to define the concept of an ideal. Let I be a set contained in $k[x_1, \ldots, x_n]$, where k is a field. If we let f and g be elements in I and h be any element in our field $k[x_1, \ldots, x_n]$, then I is an ideal if we have the following three properties:

- 0 is in I;
- $f+g$ is in I;
- and hf is in I.

Given this, a Gröbner basis is a subset of the ideal I, such that $< LT(g_1), \cdots, LT(g_t) > = < LT(I) >$. It is important to notice that the computation of Gröbner bases implies tedious and relatively complex calculations (Buchberger's Algorithm[2]). Therefore, in order to compute Gröbner bases more efficiently, a computational software of choice (Wolfram Mathematica 12) was used for this type of calculations. However, the capability of the Mathematica is limited, as it can only handle ideals where $k = \mathbb{Q}$ or a subset of $\mathbb{Q}$ (e.g. $\mathbb{Z}$). Therefore, we designed a method that would allow us to compute the Gröbner bases starting from fields that are expansions of the rational numbers. The project started observing the behavior of $\mathbb{Q}(i)$ and designing the algorithm based on that expansion. However, we soon noticed that a more general version of the same method could be applied to computations involving other field expansions, as long as those are finite.

2. Method

2.1. The Mapping

In order to compute Gröbner bases in Mathematica, the mapping links the fields $\mathbb{Q}(\sqrt{n})[x_1, \dots, x_{m-1}]$ and $\mathbb{Q}[x_1, \dots, x_m]$, with $n \in \mathbb{R},\ m \in \mathbb{N}$. Starting from the expansion field, an ideal

$$< f_1(x_1, \dots, x_{m-1}), \cdots, f_i(x_1, \dots, x_{m-1}) >$$

is uniquely mapped to a corresponding ideal

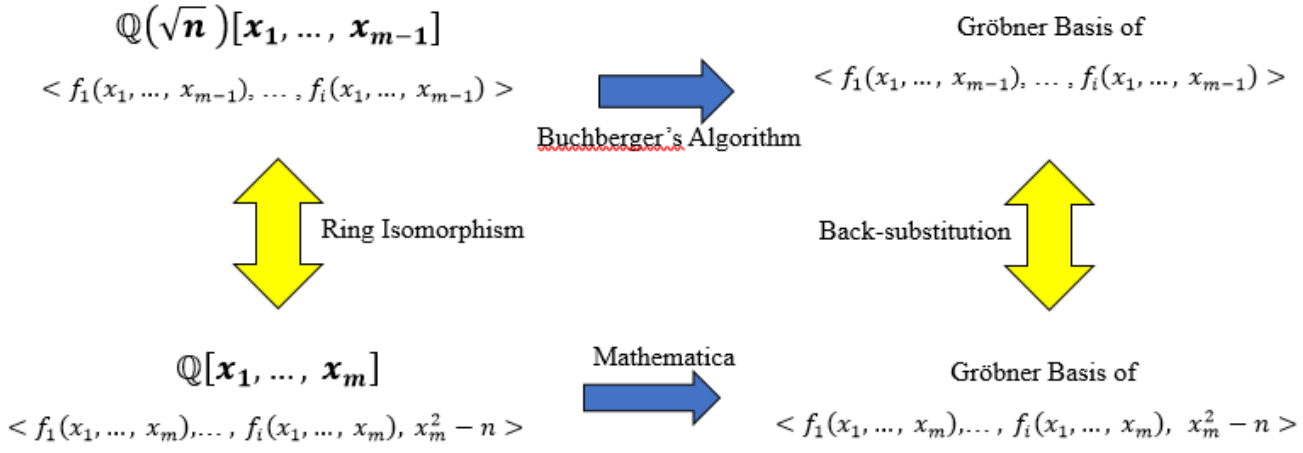
$$< f_1(x_1, \dots, x_m),\ \dots,\ f_i(x_1, \dots, x_m),\ x_m^2 - n >$$

by introducing a new variable (x_m) in the rational field and placing the restriction $x_m^2 - n$. Adding the restriction guarantees that x_m behaves like a "fake variable" and it is the key of the success of this method. An algebraic proof shows that this mapping is bijective, therefore validating the algorithm.

2.2. Mathematical Proof

The requirement for this mapping to be applicable is that the mapping between the two ideals is a ring isomorphism[3]. As a matter of fact, as shown in the diagram below, all those transitions need to be injective, so that two ideals in the first field are not mapped to the same ideal. Since the condition $x_n^2 - n$ is placed on the image ideal, all the polynomials in the ideal can be reduced to degree one, as far as the variable x_n is concerned. Therefore, all the polynomials in the starting ideal are uniquely mapped to a polynomial in the new ideal. On the other hand, the bijection part of the proof is intuitively solved, as by substituting with parameter $\sqrt{n}$ a polynomial in the starting ideal is always generated.

2.3. Visualizing the Concept



The diagram above shows graphically what was explained in the previous section. The first row represents ideals in the expansion field, while the second row deals only with rational numbers. The transition between ideals and Gröbner bases are the calculations required to move between the ideals and the basses. In other words, instead of computing Gröbner bases on

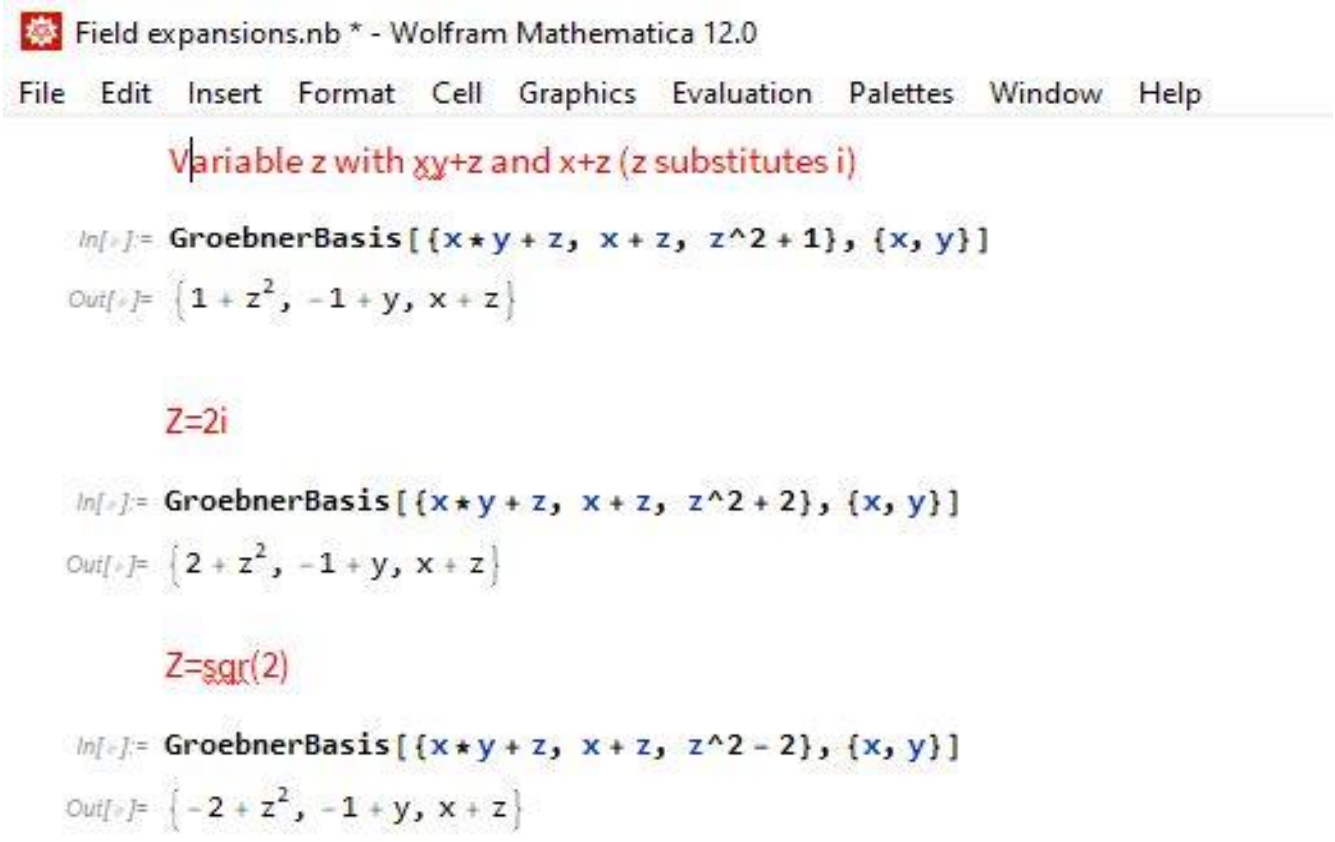
field expansions manually, we are able to break down the process so that the software can handle the calculations. Eventually, we just need to get rid of the "fake variable" and the result is the Gröbner basis of the original ideal.

3. Multiple Finite Expansions

We are able to expand this to any field extension, including those with higher roots and/or more than one extension (e.g. $\mathbb{Q}[\sqrt{2}, \sqrt{3}, \sqrt{5}]$), as long as it is a finite extension. Consequently, we cannot compute Gröbner bases in $\mathbb{R}$ using this method. The proof and the algorithm can be adjusted to meet those conditions by adding the correct number of variables and proper restrictions to build the ring isomorphism with multiple expansions.

4. Use in Mathematica

An example of our method is provided below. Starting from the ideal $< xy + \sqrt{n}, x + \sqrt{n} >$, we computed its Gröbner basis manually, using Buchberger's Algorithm, and with Mathematica, using our substitution. The ideal of choice was simple so that the Buchberger's algorithm would not get excessively convoluted. The calculated Gröbner basis was $\{y - 1; x + \sqrt{n}\}$. As shown in the screenshot below, regardless of how z was defined, it did not affect the outcome and the computed Gröbner basis matched the expected result. In more detail, three cases in which the variable z was used to substitute three different values, respectively $n = i, n = 2i, n = \sqrt{2}$, are reported below. Observing the results, two main ideas are important to notice. Firstly, the polynomial $z - n^2$ is intact in the computed basis, maintaining the ring isomorphism with the original Gröbner basis. Secondly, the variable z behaves more like a parameter, becoming almost a placeholder for the root of interest.



5. Future Directions

To continue this project, we will use other computing software, such as Pari and SAGE. Since Pari in particular can compute Gröbner Bases over field extension, our intent is to compare our method to the software's algorithm to identify similarities between the two, or optimizations that can be made.

6. Acknowledgements

The author wishes to express their appreciation to my faculty mentor Dr. Kaitlyn Perry for her valuable lead and constructive suggestions throughout the study. I would also like to thank the Reeves Summer Research Program for funding this project, and Wingate University for all the opportunities I was given.

7. References

1. Wolfram Mathematica 12, Version 12.0.0.0
2. Cox D., Little J., O'Shea *D. Ideals, Varieties, and Algorithms : an Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer, 2015.
3. Joseph Gallian. *Contemporary Abstract Algebra* (9th Edition). Cengage Learning.