A Gröbner Basis Database

Jelena Mojsilović[1]

[1] Department of Applied Mathematics, Illinois Institute of Technology

In response to the increased interest in machine learning to solve mathematical problems combined with the complexity inherent in computing Gröbner bases, we create a Gröbner basis database. Through this database we aim to provide a platform where researchers and practitioners can: obtain accurate and verified data on Gröbner bases, submit interesting and important examples of Gröbner bases, and obtain curated datasets for benchmarking, learning, and other common problems in the research community. Given the nascent stage of the database, new features are likely to be added on as it grows.

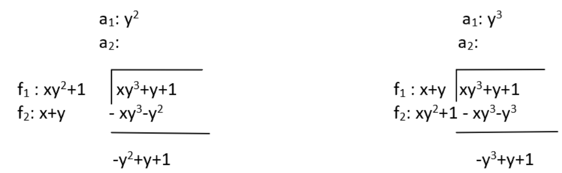

Fig. 1. The division algorithm dividing $xy^3 + y + 1$ by the set $\{x + y, xy^2 + 1\}$, utilizing two orderings on the divisors $(xy^2 + 1, x + y)$ and $(x + y, xy^2 + 1)$, yields two different remainders.

Gröbner bases are an approach to answering the *ideal membership problem* arising from computational algebraic geometry. Before we explain what this is, let us first define an ideal. An ideal, I, is a subset $I \subset k[x_1, ..., x_n]$ that satisfies:

1. $0 \in I$
2. If $f, g \in I$, then $f + g \in I$.
3. If $f \in I$ and $h \in k[x_1, ..., x_n]$, then $hf \in I$.

Thus, an ideal I is an infinite collection of polynomials and the ideal membership problem is to determine whether a polynomial f lies in an ideal I. An analogous problem in linear algebra would be determining if a vector, v, lies in a subspace V. The solution to the linear algebra problem is also analogous to the solution of the ideal membership problem. The standard method in linear algebra is to find a basis for V and attempt to find if $H_V x = v$ has a solution x, where H_V is the matrix whose columns consist of the basis. The approach in computational algebraic geometry begins with finding a finite basis for the ideal. The Hilbert Basis Theorem tells us that every ideal $I \subset k[x_1, ..., x_n]$ has a finite basis. We denote this as $I = \langle f_1, ..., f_s \rangle$, . That is, every polynomial $g \in I$ can be expressed as: $g = h_1 f_1 + h_2 f_2 + ... + h_s f_s$ for some set $\{h_1, ..., h_s\} \subset k[x_1, ..., x_n]$.

A problem arises from this definition: If we are given a polynomial, f, how do we determine whether $f \in I$? This is synonymous with the ideal membership problem.

The Division Algorithm

The division algorithm provides a partial solution to the ideal membership problem, regardless of the basis.

Fix a monomial order $>$. Let $F = (f_1, ..., f_n)$ be an ordered s-tuple of polynomials in $k[x_1, ..., x_n]$. Then every $f \in k[x_1, ..., x_n]$ can be expressed as $f = a_1 f_1 + ... + a_s f_s + r$ where $a_i, r \in k[x_1, ..., x_n]$.

Combining the division algorithm with the ideal membership problem, we see that a polynomial $f \in I$ if $r = 0$. However, the division algorithm does not specify in which order the divisors must be utilized. This allows the remainder upon division to differ depending on the ordering of divisors. The example in Figure 1 is a demonstration of this, determining whether $f = xy^3 + y + 1 \in \langle xy^2 + 1, x + y \rangle$.

This property is problematic because the ordering of divisors does not provide a unique remainder. We want a basis that retains the property that the ordering of divisors does not change the remainder. We must ask: is there a better basis for I such that the remainder, r, is uniquely determined? A Gröbner basis is such a basis.

Gröbner Bases

A Gröbner basis is a generating set of an ideal in a polynomial ring, $k[x_1, ..., x_n]$, over a field, k. More formally letting LT denote the leading term of a polynomial, this basis is defined as follows.

Definition 1 Under a specific term order, a finite subset $G = \{g_1, ..., g_t\}$ of an ideal I is said to be a **Gröbner basis** if $\langle LT(g_1), .., LT(g_t) \rangle = \langle LT(I) \rangle$.

A consequence of this definition is that every non-zero ideal has a Gröbner basis. The following two properties of a Gröbner basis allow for it to be the generating set for I that we are looking for.

Theorem 1 *Let $G = \{g_1, ..., g_t\}$ be a Gröbner basis for an ideal $I \subset k[x_1, ..., x_n]$ and let $f \in k[x_1, ..., x_n]$. Then there is a unique remainder, $r \in k[x_1, ..., x_n]$ with the following two properties.*

1. *No term of r is divisible by any of $LT(g_1), ..., LT(g_t)$.*
2. *There is $g \in I$ such that $f = g + r$.*

In particular, r is the remainder on division of f by G no matter how the elements of G are listed when using the division algorithm.

For this reason, Gröbner bases solve the problem of finding a solution to a multivariate, non-linear system of polynomials.

Computation and Application of Gröbner Bases . Gröbner bases are a cornerstone of computation with polynomials and have a wide variety of applications such as robotics, engineering, statistics, and biology. Gröbner bases provide a solution to an EXSPACE hard ideal membership problem, which can be reduced to the word equivalence problem [2].

A Gröbner Basis Database. Given the complexity of Gröbner bases combined with an increase in interest using machine learning to solve mathematical problems, a Gröbner basis database would provide the research community with a robust set of Gröbner basis data. The goal of the database is two-fold: to generate a set for potential future learning, and to provide a way for researchers to cross reference to see if their physical problem is equivalent to some other known problem.

Methods

To create the database, we use MySQL in order to form a table of rows and columns. Each column is one of seventeen fields: starting generating set, reduced Gröbner basis, field, term order, number of variables, size of the reduced GB, max total degree, min total degree, max multi-degree, min multi-degree, minimum number of generators, code, name, description, submission by, verified by, and submission date. These fields are chosen because they are well-defined properties of Gröbner bases.

The database is connected to a website, Groebner Basis Database, where users can find a search function to query the database, pre-made datasets for machine learning, benchmarking problems, and programming, as well as a submissions page in order to submit their own Gröbner basis data. In addition to inputting our own Gröbner basis data, we encourage others to submit data so that we can have a robust dataset. Our database can be found at: http://math.iit.edu/~groebnerdatabase/index.html.

Fields of the Database. All of the mathematical definitions below are taken from [1] before the list begins.

- Starting Generating Set: We organize our database by the starting generating set because it defines the polynomial system from which we will construct a (reduced) Gröbner basis.
- Reduced Gröbner Basis: A **reduced Gröbner basis** for a polynomial ideal I is a Gröbner basis G for I such that:
  - LC(p)= 1 for all $p \in G$, where $LC(p)$ denotes the leading constant of the polynomial p.
  - For all $p \in G$, no monomial of p lies in $\langle LT(G-\{p\}\rangle$.

  We choose to include the reduced Gröbner basis because it is unique given a monomial ordering. Thus for any two ideals, in order to verify if they have the same basis given a monomial order, in fact, to determine whether they are the same ideal, a reduced Gröbner basis must be found.
- Field: The field of coefficients overwhich a Gröbner basis is calculated must be specified by definition.
- Term Order: A **term order** on $k = [x_1, ..., x_n]$ is any relation on $Z^n_{\geq 0}$, or equivalently, any relation on the set of monomials x^α satisfying:
  - $>$ is a total (linear) ordering on $Z^n_{\geq 0}$;
  - if $\alpha > \beta$ and $\gamma \in Z^n_{\geq 0}$, then $\alpha + \gamma > \beta + \gamma$;
  - $>$ is a well-ordering on $Z^n_{\geq 0}$. In other words, every nonempty subset of $Z^n_{\geq 0}$ has a smallest element under $>$.

  A term order is necessary in order to compute a Gröbner basis by definition, thus it must be included as a field in our database. For every term order, we standardize the variables. For example, for a system given in (x, y, z) we convert the system to x_1, x_2, x_3. This is done in order to prevent double entries in the database of the type that have the same initial generating set and same reduced Gröbner basis.
- Size of the Reduced Gröbner Basis: The size of the reduced Gröbner basis does change depending on term order. This information could distinguish the same ideal double-entered into the database whose Gröbner bases are calculated under different term orders.
- Maximum Total Degree: For a given monomial order, the maximum of the degrees of the polynomials in the Gröbner basis, where the **degree** of a polynomial p whose leading term is $LT(p) = c_\alpha x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is defined to be $deg(p) = \alpha_1 + \cdots + \alpha_n$, is $maxdeg(g_1), ..., deg(g_t)$.
- Minimum Total Degree: For a given monomial order, the minimum of the degrees of the polynomials in the Gröbner basis, where the **degree** of a polynomial p whose leading term is $LT(p) = c_\alpha x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is defined to be $deg(p) = \alpha_1 + \cdots + \alpha_n$, is $mindeg(g_1), ..., deg(g_t)$.
- Maximum Multidegree: For a given monomial order, the maximum of the multidegrees of the polynomials in the Gröbner basis, where the **multidegree** of a polynomial p whose leading term is $LT(p) = c_\alpha x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is defined to be the vector $deg(p) = (\alpha_1, \dots, \alpha_n)$, is the vector $MaxMdeg = (max\{\alpha_{11}, \dots, \alpha_{1t}\}, \dots, max\{\alpha_{n1}, \dots, \alpha_{nt}\})$
- Minimum Multidegree: For a given monomial order, the minimum of the multidegrees of the polynomials in the Gröbner basis, where the **multidegree** of a polynomial p whose leading term is $LT(p) = c_\alpha x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is defined to be the vector deg$(p) = (\alpha_1, \dots, \alpha_n)$, is the vector $MinMdeg = (min\{\alpha_{11}, \dots, \alpha_{1t}\}, \dots, min\{\alpha_{n1}, \dots, \alpha_{nt}\})$
- Minimum Number of Generators: We include the minimum number of generators because it specifies the size of the input system of polynomials.
- Code: For the time being this is not an active field. However, this field will become a source for users to find code for widely used programs, such as Macaulay2 or Sage, in order to perform calculations for a given entry in the database.
- Name: Name is an optional field which is included in order to provide the names of ideals which are more commonly known by some name.
- Verified By: The verified by field will allow the user to know that the data of an entry in the database has been checked for correctness.

Current Method. The small examples we have included for the start of the database were calculated in Macaulay2. Macaulay2 is open-source software devoted to supporting research in computational algebraic geometry and commutative algebra. Given the complexity of the problem, the future method of computation may differ.

Discussion

This database is a long-term project that will require editing and maintenance over time. We hope this database will serve as a useful tool for the research community. As such, we encourage input from those who will use it in an effort to improve this database.

A major challenge in creating this database was determining what properties of Gröbner bases would be useful to the research community and if these properties should be included. Thus, we have to determine what the ideal database would be and what can be made realistically. We acknowledge that our database will not contain all properties of a Gröbner basis that may be used for research, but our database will contain those properties of a Gröbner basis that will be most often sought after by the research community.

Conclusion

We have set the foundation of our database with a basic outline of what we want the database to contain. This database should be viewed as a major work in progress that will change over time. For our purposes, we hope to use the dataset we generate through this database in future work in machine learning on Gröbner bases.

Future Work. In the future we plan to work on handling user-generated datasets, creating an automated method of transferring uploads to the database, and user submission of data that is too complex for the automated database entry process. User generated datasets may contain missing information and most likely will require some calculations in order to fill null entries. The current method of adding entries into the database requires physical entry from a submission. This is inefficient and time costly. In the future, it would be more efficient to create an automated process so as to allow more time for handling more complex entries. Complex entries may take a large amount of time to handle, so these entries must be filtered out of the automation process. However, this does not solve the issue of how the data for these entries will be verified and/or calculated.

ACKNOWLEDGMENTS. This database is a joint work with Travis Koehring, Sonja Petrović, Despina Stasi, and Sara Jamshidi Zelenberg. We would like to thank the Applied Mathematics Department at the Illinois Institute of Technology for hosting our database. We would also like to thank Upendra Ghandi for his help and support in the creation of our database and website this past summer.

References

[1] David A. Cox , John Little , Donal O'Shea, Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, Springer Publishing Company, Incorporated, 2015

[2] Earnst W. Mayer, Some Complexity Results for Polynomial Ideals, Journal of Complexity 13, 303-325, 1997