Using Algebraic Geometry
David Cox, John Little, Don O'Shea
Update on Maple's `Groebner` Package
March 13, 2010

General Information

Since the second edition of *Using Algebraic Geometry* appeared in 2005, Maple's `Groebner` package has undergone further revisions and extensions. Hence some of the discussion in the text and some examples are no longer up-to-date. The purposes of this update are to indicate the most important changes, to illustrate some of the increased functionality now provided, and to suggest some alternatives when calculations are difficult or awkward using Maple. This update refers to the version of the `Groebner` package supplied with Maple 13, the most recent version currently available. However some of the changes described below first appeared in earlier versions.

Defining Monomial Orders, Computing Gröbner Bases, etc.

The basic syntax of monomial order definitions, the `LeadingCoefficient, LeadingMonomial, and LeadingTerm` commands, the `NormalForm` command for the remainder on division, the `SPolynomial` command, and the `Basis` command for computing Gröbner bases are described in the new version of the Maple appendix for our text *Ideals, Varieties, and Algorithms*, available online at `http://www.cs.amherst.edu/~dac/iva/appc.pdf`. That appendix also discusses the `IsZeroDimensional, UnivariatePolynomial, Solve,` and `HilbertPolynomial` commands and the `PolynomialIdeals` package, which contains procedures implementing the methods described in *Ideals, Varieties, and Algorithms* for computing ideal intersections, ideal quotients and saturations, primary decompositions, and so forth.

Computations in Finite-Dimensional Algebras

There are new names and revised syntax for two commands dealing with computations in the quotient ring $k[x_1, \ldots, x_n]/I$, where I is a zero-dimensional ideal, as discussed in Chapter 2 of *Using Algebraic Geometry.* First, the command for computing the basis for $k[x_1, \ldots, x_n]/I$ as a k-vector space consisting of the monomials in the complement of $\langle LT(I)\rangle$ is now called `NormalSet`. The syntax is

```
> NormalSet(G,monomialorder)
```

where G is a Gröbner basis for I (or alternatively, an ideal defined using the `PolynomialIdeals` package, in which case a Gröbner basis will be computed in the course of execution). The output consists of a sequence with two parts: first the monomial basis as a list arranged in increasing order according the monomial order specified (this is different from the way the monomial basis is ordered in an example in the text); second, a table that facilitates extracting the vector of coefficients of a polynomial representative for a coset in $k[x_1, \ldots, x_n]/I$ with respect to the monomial basis.

A matrix m_f for the multiplication mapping by $f \in k[x_1, \ldots, x_n]/I$ is now computed by a command called `MultiplicationMatrix`. (But note, the Maple command computes the *transpose* of the matrix as defined in the text.) The syntax for this command requires the user to supply both parts of the output of the `NormalSet` command as inputs. Hence, the easiest way to use these is to first compute the `NormalSet` and give names to the two terms in the output sequence, then use them as inputs to `MultiplicationMatrix`:

```
> monlist,montable:=NormalSet(G,monomialorder)
```

then

```
> MultiplicationMatrix(f,monlist,montable,G,monomialorder)
```

The output is a `Matrix` defined via the `LinearAlgebra` package. Hence the interface with the matrix algebra and numerical routines provided there is seamless.

Gröbner Basis Conversion

The `Groebner` package now contains procedures implementing both the FGLM basis conversion for zero-dimensional ideals described in §3 of Chapter 2 and the general Gröbner Walk algorithm described in §5 of Chapter 8. The `FGLM` command

```
> FGLM(G,monomialorder1,monomialorder2)
```

takes as input a Gröbner basis G with respect to `monomialorder1`, and computes the reduced Gröbner basis for the ideal generated by G with respect to `monomialorder2`. The input G may also be a `PolynomialIdeal`. In that case, a Gröbner basis for the ideal generated by G with respect to the first monomial order will be computed if that is not already known. Various optional arguments may also be specified, including one which will terminate execution by means of a degree bound on the monomials processed by the algorithm. With this option, the `FGLM` procedure will also terminate when the ideal generated by G is not zero-dimensional, but the output may not be a Gröbner basis.

The syntax of the Gröbner Walk procedure is similar:

```
> Walk(G,monomialorder1,monomialorder2)
```

As for the `FGLM` procedure, there are additional optional arguments that may be specified, including one which implements the "sudden death" strategy for computing elimination ideals discussed on page 445 of the text when `monomialorder2` is a Maple `lexdeg` elimination order.

The Current Maple "Gröbner Engine"

The `Basis` command of the Maple 13 `Gröbner` package incorporates five different algorithms for computing Gröbner bases:

- a compiled implementation of Faugère's F4 algorithm (`FGb`), which is often the fastest, but is somewhat less general in that it does not support coefficients in a rational function field, provides only some monomial orders, and does not allow for coefficients mod p for large primes p,

- an interpreted Maple implementation of F4 which is completely general,
- the traditional Buchberger algorithm, with the normal pair selection strategy (which is actually superior in some special cases, e.g. for the case where a single polynomial is adjoined to a known Gröbner basis),
- the FGLM basis conversion algorithm,
- the Gröbner Walk conversion algorithm.

The default `Basis` procedure relies on heuristics for selecting which method will be used. For instance, for many problems `FGb` (or the Maple F4) will be the default choice. However, for *lex* Gröbner bases for zero-dimensional ideals, a *grevlex* Gröbner basis computation, followed by FGLM basis conversion is the usual current default method. These defaults are subject to updates in future versions as the state of the art evolves. Options for specifying exactly how the computation will be performed are also provided.

Local Orders and Standard Bases

In addition to the usual monomial orders, Maple also supports general semigroup orders and in particular the local orders discussed in Chapter 4 of *Using Algebraic Geometry*. These can be defined directly as matrix orders, for example. However the Mora Normal Form algorithm is not implemented, and hence `Basis` computations using these orders may not terminate (unless the input polynomials are homogeneous). It is, of course, possible to use the homogenization process described in Definition (3.6) from §3 of Chapter 4, and the homogenous Mora Normal Form algorithm in Theorem (3.10) from that section to compute standard bases. Still, for this type of computation, the package `Singular` is probably a better option than Maple.

Gröbner Bases for Modules

Computation of Gröbner bases for modules over $R = k[x_1, \ldots, x_n]$ can be performed in Maple by a (built-in) implementation of the idea from Exercise 6 in §2 of Chapter 5 of *Using Algebraic Geometry*. The process involves defining a polynomial algebra using the `Ore_Algebra` package, then specifying a monomial order on that polynomial algebra which treats the extra variables (or alternatively, monomials in a single additional variable) representing the standard basis vectors in R^m as "module placeholder variables." The usual `Basis` command then computes module Gröbner bases. The details are laid out clearly in the Maple interactive help facility. As is the case for standard bases, however, other symbolic algebra packages such as COCOA, `Singular`, and *Macaulay 2* are probably still better choices than Maple for these computations.