

XIDEAL

Gröbner Bases for Exterior Algebra

David Hartley ^{*†}

Institute for Algorithms and Scientific Computing
GMD — German National Research Center
for Information Technology
St Augustin, Germany

Philip A. Tuckey [‡]

Laboratoire de Physique Moléculaire
UFR des Sciences et Techniques
Université de Franche-Comté
25030 Besançon
France

Version 2.4

1 Description

The method of Gröbner bases in commutative polynomial rings introduced by Buchberger (e.g. [1]) is a well-known and very important tool in polynomial ideal theory, for example in solving the ideal membership problem. XIDEAL extends the method to exterior algebras using algorithms from [2] and [3].

There are two main departures from the commutative polynomial case. First, owing to the non-commutative product in exterior algebras, ideals

^{*}Postal address: GMD-SCAI, D-53754 St Augustin, Germany.

[†]Email: Hartley@gmd.de

[‡]Email: pat@rs3.univ-fcomte.fr

are no longer automatically two-sided, and it is necessary to distinguish between left and right ideals. Secondly, because there are zero divisors, confluent reduction relations are no longer sufficient to solve the ideal membership problem: a unique normal form for every polynomial does not guarantee that all elements in the ideal reduce to zero. This leads to two possible definitions of Gröbner bases as pointed out by Stokes [4].

XIDEAL constructs Gröbner bases for solving the left ideal membership problem: Gröbner left ideal bases or GLIBs. For graded ideals, where each form is homogeneous in degree, the distinction between left and right ideals vanishes. Furthermore, if the generating forms are all homogeneous, then the Gröbner bases for the non-graded and graded ideals are identical. In this case, XIDEAL is able to save time by truncating the Gröbner basis at some maximum degree if desired.

XIDEAL uses the exterior calculus package EXCALC of E. Schrüfer [5] to provide the exterior algebra definitions. EXCALC is loaded automatically with XIDEAL. The exterior variables may be specified explicitly, or extracted automatically from the input polynomials. All distinct exterior variables in the input are assumed to be linearly independent – if a dimension has been fixed (using the EXCALC `spacedim` or `coframe` statements), then input containing distinct exterior variables with degrees totaling more than this number will generate an error.

2 Declarations

xorder

`xorder` sets the term ordering for all other calculations. The syntax is

```
xorder k
```

where `k` is one of `lex`, `gradlex` or `deglex`. The lexicographical ordering `lex` is based on the prevailing EXCALC kernel ordering for the exterior variables. The EXCALC kernel ordering can be changed with the REDUCE `korder` or EXCALC `forder` declarations. The graded lexicographical ordering `gradlex` puts terms with more factors first (irrespective of their exterior degrees) and sorts terms of the same grading lexicographically. The degree lexicographical ordering `deglex` takes account of the exterior degree of the variables, putting highest degree first and then sorting terms of the same

degree lexicographically. The default ordering is `deglex`.

xvars

It is possible to consider scalar and 0-form variables in exterior polynomials in two ways: as variables or as coefficient parameters. This difference is crucial for Gröbner basis calculations. By default, all scalar variables which have been declared as 0-forms are treated as exterior variables, along with any EXCALC kernels of degree 0. This division can be changed with the `xvars` declaration. The syntax is

```
xvars U,V,W,...
```

where the arguments are either kernels or lists of kernels. All variables specified in the `xvars` declaration are treated as exterior variables in subsequent XIDEAL calculations with exterior polynomials, and any other scalars are treated as parameters. This is true whether or not the variables have been declared as 0-forms. The declaration

```
xvars {}
```

causes all degree 0 variables to be treated as parameters, and

```
xvars nil
```

restores the default. Of course, p -form kernels with $p \neq 0$ are always considered as exterior variables. The order of the variables in an `xvars` declaration has no effect on the REDUCE kernel ordering or XIDEAL term ordering.

3 Operators

xideal

`xideal` calculates a Gröbner left ideal basis in an exterior algebra. The syntax is

```
xideal(S:list of forms[,V:list of kernels][,R:integer])
      :list of forms.
```

`xideal` calculates a Gröbner basis for the left ideal generated by `S` using the current term ordering. The resulting list can be used for subsequent

reductions with `xmod` as long as the term ordering is not changed. Which 0-form variables are to be regarded as exterior variables can be specified in an optional argument `V` (just like an `xvars` declaration). The order of variables in `V` has no effect on the term ordering. If the set of generators `S` is graded, an optional parameter `R` can be given, and `xideal` produces a truncated basis suitable for reducing exterior forms of degree less than or equal to `R` in the left ideal. This can save time and space with large problems, but the result cannot be used for exterior forms of degree greater than `R`. The forms returned by `xideal` are sorted in increasing order. See also the switches `trxideal` and `xfullreduction`.

xmodideal

`xmodideal` reduces exterior forms to their (unique) normal forms modulo a left ideal. The syntax is

```
xmodideal(F:form, S:list of forms):form
```

or

```
xmodideal(F:list of forms, S:list of forms)
      :list of forms.
```

An alternative infix syntax is also available:

```
F xmodideal S.
```

`xmodideal(F,S)` first calculates a Gröbner basis for the left ideal generated by `S`, and then reduces `F`. `F` may be either a single exterior form, or a list of forms, and `S` is a list of forms. If `F` is a list of forms, each element is reduced, and any which vanish are deleted from the result. If the set of generators `S` is graded, then a truncated Gröbner basis is calculated using the degree of `F` (or the maximal degree in `F`). See also `trxmod`.

xmod

`xmod` reduces exterior forms to their (not necessarily unique) normal forms modulo a set of exterior polynomials. The syntax is

```
xmod(F:form, S:list of forms):form
```

or

`xmod(F:list of forms, S:list of forms):list of forms.`

An alternative infix syntax is also available:

`F xmod S.`

`xmod(F,S)` reduces `F` with respect to the set of exterior polynomials `S`, which is not necessarily a Gröbner basis. `F` may be either a single exterior form, or a list of forms, and `S` is a list of forms. This operator can be used in conjunction with `xideal` to produce the same effect as `xmodideal`: for a single homogeneous form `F` and a set of exterior forms `S`, `F xmodideal S` is equivalent to `F xmod xideal(S,exdegree F)`. See also `trxmod`.

xauto

`xauto` autoreduces a set of exterior forms. The syntax is

`xauto(S:list of forms):list of forms.`

`xauto S` returns a set of exterior polynomials which generate the same left ideal, but which are in normal form with respect to each other. For linear expressions, this is equivalent to finding the reduced row echelon form of the coefficient matrix.

excoeffs

The operator `excoeffs`, with syntax

`excoeffs(F:form):list of expressions`

returns the coefficients from an exterior form as a list. The coefficients are always scalars, but which degree 0 variables count as coefficient parameters is controlled by the command `xvars`.

exvars

The operator `exvars`, with syntax

`exvars(F:form):list of kernels`

returns the exterior powers from an exterior form as a list. All non-scalar variables are returned, but which degree 0 variables count as coefficient

parameters is controlled by the command `xvars`.

4 Switches

`xfullreduce`

on `xfullreduce` allows `xideal` and `xmodideal` to calculate reduced, monic Gröbner bases, which speeds up subsequent reductions, and guarantees a unique form for the Gröbner basis. off `xfullreduce` turns of this feature, which may speed up calculation of some Gröbner basis. `xfullreduce` is on by default.

`trxideal`

on `trxideal` produces a trace of the calculations done by `xideal` and `xmodideal`, showing the basis polynomials and the results of the critical element calculations. This can generate profuse amounts of output. `trxideal` is off by default.

`trxmod`

on `trxmod` produces a trace of reductions to normal form during calculations by XIDEAL operators. `trxmod` is off by default.

5 Examples

Suppose XIDEAL has been loaded, the switches are at their default settings, and the following exterior variables have been declared:

```
pform x=0,y=0,z=0,t=0,f(i)=1,h=0,hx=0,ht=0;
```

In a commutative polynomial ring, a single polynomial is its own Gröbner basis. This is no longer true for exterior algebras because of the presence of zero divisors, and can lead to some surprising reductions:

```
xideal {d x^d y - d z^d t};
```

```

{d t^d z + d x^d y,
 d x^d y^d z,
 d t^d x^d y}

f(3)^f(4)^f(5)^f(6)
xmodideal {f(1)^f(2) + f(3)^f(4) + f(5)^f(6)};

0

```

The heat equation, $h_{xx} = h_t$ can be represented by the following exterior differential system.

```

S := {d h - ht*d t - hx*d x,
      d ht^d t + d hx^d x,
      d hx^d t - ht*d x^d t};

```

`xmodideal` can be used to check that the exterior differential system is closed under exterior differentiation.

```

d S xmodideal S;

{}

```

`xvars`, or a second argument to `xideal` can be used to change the division between exterior variables of degree 0 and parameters.

```

xideal {a*d x+y*d y};

      d x*a + d y*y
      {-----}
      a

xvars {a};
xideal {a*d x+y*d y};

      {d x*a + d y*y, d x^d y}

xideal({a*d x+y*d y},{a,y});

      {d x*a + d y*y,

```

```

      d x^d y*y}

xvars {};          % all 0-forms are coefficients
excoeffs(d u - (a*p - q)*d y);

      {1, - a*p + q}

exvars(d u - (a*p - q)*d y);

      {d u,d y}

xvars {p,q};      % p,q are no longer coefficients
excoeffs(d u - (a*p - q)*d y);

      { - a,1,1}

exvars(d u - (a*p - q)*d y);

      {d y*p,d y*q,d u}

xvars nil;

```

Non-graded left and right ideals are no longer the same:

```

d t^(d z+d x^d y) xmodideal {d z+d x^d y};

0

(d z+d x^d y)^d t xmodideal {d z+d x^d y};

- 2*d t^d z

```

Any form containing a 0-form term generates the whole ideal:

```

xideal {1 + f(1) + f(1)^f(2) + f(2)^f(3)^f(4)};

{1}

```


References

- [1] B. Buchberger, *Gröbner Bases: an algorithmic method in polynomial ideal theory*, in *Multidimensional Systems Theory* ed. N.K. Bose (Reidel, Dordrecht, 1985) chapter 6.
- [2] D. Hartley and P.A. Tuckey, *A Direct Characterisation of Gröbner Bases in Clifford and Grassmann Algebras*, Preprint MPI-Ph/93-96 (1993).
- [3] J. Apel, *A relationship between Gröbner bases of ideals and vector modules of G -algebras*, Contemporary Math. **131**(1992)195–204.
- [4] T. Stokes, *Gröbner bases in exterior algebra*, J. Automated Reasoning **6**(1990)233–250.
- [5] E. Schrüfer, *EXCALC, a system for doing calculations in the calculus of modern differential geometry, User's manual*, (The Rand Corporation, Santa Monica, 1986).