# Tutorial FGb

Jean-Charles Faugère[1,2,3]

June 2015

FGb is a C library for computing Gröbner bases and polynomial system solving with algebraic computations. FGb is freely distributed for academic use only. It can be used as a C library through a C program or through its interface with the Maple computer algebra system. Most of the functions are available for these two usage modes.

This tutorial presents FGb through its Maple interface. The various functionalities compute Gröbner bases or related objects which are useful for solving polynomial systems with algebraic methods (symbolic computation). The structure of the library is described in [4].

Most of the functions provided by FGb are based on the state-of-the-art algorithms for Gröbner bases such as the $F_4$ and $F_5$ algorithms [2, 3]. These algorithms are based on linear algebra routines which are given in [5]: coefficients of the polynomials are stored in row vectors according to a *monomial ordering* ; matrices are formed by taking such row vectors obtained by multiplying the input polynomials with monomials up to some degree and performing row echelon form computations.

In order to use FGb, it is not necessary to know more about Gröbner bases and monomial orderings than the few lines below. However for an expert usage, learning about such notions is really useful: such knowledge is crucial to know how to use efficiently these functions. The interested reader can refer to [1, Chapter 2].

The main monomial orderings provided by FGb are the degree reverse lexicographical ordering (denoted by DRL(vars)) and an elimination ordering DRL(vars1) > DRL(vars2) (here we mean that variables in vars1 are eliminated). You can use FGb without knowing the definition of the DRL-ordering. Gröbner bases are special families of polynomials that generate the same *ideal* (i.e. set of algebraic combinations) of the input polynomials. Their special properties allow to extract various useful information about the solution set of the input polynomial system except a description of the projection of the solution set on the space spanned by a subset of the input variables.

For this latter task, you may use the aforementioned elimination ordering: eliminating some variables leads to compute the projection on the space spanned by the other variables. In the

[1]INRIA Paris-Rocquencourt, PolSys Project, France.
[2]Sorbonne Universités, UPMC Univ Paris 06, Equipe PolSys, LIP6, F-75005, Paris, France.
[3]CNRS, UMR 7606, LIP6, France.

(common) case where the solution set (in some algebraic closure) is finite, special change of ordering algorithms are used [7, 6].

## Contents



# 1 Getting started

The package FGb must be loaded in your maple session with the following command.

```
> with(FGb);
```

All FGb commands available through its Maple interface appear. These commands provide functionalities to solve polynomial systems with coefficients in a field $\mathbb{K}$ through Gröbner bases computations.

The fields which are supported are the prime fields $\mathbb{Z}/p\mathbb{Z}$ with $p < 2^{16}$ or the field $\mathbb{Q}$ of rational numbers.

The first task that FGb allows to tackle is the following: given a polynomial system F with coefficients in a field $\mathbb{K}$ as above, decide if it has solutions in the algberaic closure of $\mathbb{K}$. To do that, we compute a Gröbner basis using the main function of the FGb package which is fgb_gbasis. The Gröbner basis is [1] if and only if there is no solution in the algebraic closure.

```
# The following toy example is known as Katsura-3 problem.

> F:=[2*x+2*y+2*z+t-1, 2*x*z+z^2+2*y*t-y,
2*x*y+2*y*z+2*z*t-z, 2*x*z+2*y*t+z^2-y,
2*x^2+2*y^2+2*z^2+2*t^2-t];

#F considered with rational coefficients

> fgb_gbasis(F, 0, [], [x,y,z,t]);

[135230915*t^4+2604491-216024792*t^3+3881504*y*t-11331680*z*t+
126340686*t^2-537232*y+5402944*z-31985976*t,
2494*y*t^2-57+210*t^3-2640*y*t+244*z*t-453*t^2+718*y-104*z+270*t,
43*z*t^2-1+15*t^3+8*y*t-44*z*t-17*t^2-4*y+11*z+7*t,
2*y^2+y*t-2*z*t-y+z, 8*y*z-1-16*y*t-4*z*t-5*t^2+8*y+4*t,
4*z^2+1+8*y*t+8*z*t+5*t^2-4*y-4*z-4*t, 2*x+2*y+2*z+t-1]

#F considered with coefficients in Z/(65521 Z)
> fgb_gbasis(F, 65521, [], [x,y,z,t]);

[t^4+59696+55447*t^3+9849*y*t+24675*z*t+7763*t^2+37061*y+53842*z+10802*t,
y*t^2+42586+1734*t^3+34362*y*t+18863*z*t+28084*t^2+30265*y+47814*z+39670*t,
z*t^2+59426+25904*t^3+48760*y*t+59425*z*t+27427*t^2+41141*y+1524*z+42665*t,
y^2+32761*y*t+65520*z*t+32760*y+32761*z,
y*z+8190+65519*y*t+32760*z*t+40950*t^2+y+32761*t,
z^2+49141+2*y*t+2*z*t+49142*t^2+65520*y+65520*z+65520*t,
x+32760+y+z+32761*t]
```

We deduce from the above computations that the system F has solutions (in $\mathbb{C}$ and in the algebraic closure of $\mathbb{Z}/65521\mathbb{Z}$). Let us now consider the intersection of the complex solutions of F and the hyperplane defined by $x + 2y + 3z + 4t - 5$ and check if it has solutions.

```
> F:=[op(F), x+2*y+3*z+4*t-5]:

> fgb_gbasis(F, 0, [], [x,y,z,t]);

[1]
```

We deduce that this intersection is empty over $\mathbb{C}$. The same conclusion holds over the algebraic closure of $\mathbb{Z}/65521\mathbb{Z}$.

```
> fgb_gbasis(F, 65521, [], [x,y,z,t]);

[1]
```

We continue this section with additional examples. All of them can be found in maple files of the directory examples provided with the distribution of FGb.

### 1.1 Finitely many solutions.

Let us see how to check if a given polynomial system has a finite number of complex solutions.

```
#The system below is obtained by eliminating the variable t
#from the Katsura-3 problem above
> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];

#We start by computing a Grobner basis for the ordering DRL(x>y>z)

> gb:=fgb_gbasis(F, 0, [], [x,y,z]):

[27675164*z^4-7969044*z^3+3377930*x*z+5745964*y*z+5766613*z^2+
90629*y-1779826*z,
1276*x*z^2+1201*z^3-200*x*z-313*y*z-420*z^2-22*y+93*z,
116*y*z^2-28*z^3+22*x*z+44*y*z+23*z^2+3*y-14*z,
20*x^2+2+66*x*z+24*y*z+49*z^2-12*x-7*y-18*z,
2*x*y-2*y*z-4*x*z-4*z^2+z,
4*y^2+6*x*z+8*y*z+7*z^2-y-2*z]

#Next, we call the function fgb_hilbert: it returns
#a polynomial in u and an integer which is 0 iff the
#system defines a finite set.

> hilb:=fgb_hilbert(gb, 0, [], [x,y,z],'t'):

[u^3+3*u^2+3*u+1, 0]

> hilb[2];

0
```

When the complex solution set is finite (which is the case here), one may want to get the number of solutions (counted with multiplicities). This is done again using the output of hilb

```
> subs(u=1, hilb);
```

Let p be a prime number. It is remarkable that for any choice of p outside a finite subset of the set of prime numbers, running the above computations in the field $\mathbb{Z}/p\mathbb{Z}$ yields the dimension and degree estimates. This is very useful for large examples since computations in $\mathbb{Z}/p\mathbb{Z}$ are much faster when the size of $p$ is less than the word machine. This is what we do below with $p = 65521$.

```
#The system below is obtained by eliminating the variable t
#from the Katsura-3 problem above
> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];

#We start by computing a Grobner basis for the ordering DRL(x>y>z)
> gb:=fgb_gbasis(F, 65521, [], [x,y,z]):

[z^4+52916*z^3+1449*x*z+56695*y*z+35868*z^2+9919*y+13038*z,
x*z^2+32402*z^3+42722*x*z+45238*y*z+11091*z^2+3389*y+54173*z,
y*z^2+47446*z^3+28242*x*z+56484*y*z+59308*z^2+27677*y+23723*z,
x^2+58969+45868*x*z+52418*y*z+36039*z^2+39312*x+22932*y+58968*z,
x*y+65519*x*z+65520*y*z+65519*z^2+32761*z,
y^2+32762*x*z+2*y*z+16382*z^2+16380*y+32760*z]

#Next, we call the function fgb_hilbert:
#it returns a polynomial in u
> hilb:=fgb_hilbert(gb, 65521, [], [x,y,z],'u'):
#and an integer which is 0 iff the system defines a finite set.

[u^3+3*u^2+3*u+1, 0]
> hilb[2];

0

> subs(u=1, hilb[1]);

8
```

In the above example, the polynomial system has a finite number of complex solutions. This can be seen on the Gröbner basis: the set of leading monomials of the polynomials in the Gröbner basis contains pure powers of all variables.

For large examples, it is not so easy to extract these leading monomials of the polynomials in the Gröbner basis. The function fgb_gbasis_lm is useful because it makes this easy. Besides, the leading monomials in the basis are the only ones we really need to run fgb_hilbert.

```
> gb:=fgb_gbasis_lm(F, 65521, [], [x,y,z]):

> gb[2];

[z^4, x*z^2, y*z^2, x^2, x*y, y^2]

> hilb:=fgb_hilbert(gb[2], 65521, [], [x,y,z], 'u');

[u^3+3*u^2+3*u+1, 0]
```

At this point, we know that the above system F has finitely many complex solutions and we know how to count the number of complex solutions (with multiplicities). To go further we may want to know what is the projection of the solution set of F on the $x$-axis. Hence we want to compute a polynomial in $\mathbb{Q}[x]$ This is done by *eliminating* the variables $y$ and $z$ using the function fgb_gbasis_elim.

```
> xpol:=fgb_gbasis_elim(F, 0, [y,z], [x]);

[3470240*x^8+1-4582784*x^7+2519968*x^6-750640*x^5+
139000*x^4-19216*x^3+2202*x^2-146*x]
```

Finally, we may want a *nice* representation of this complex solution set in terms of univariate polynomials. More precisely, we would like to express the $z$- and $y$-coordinates with respect to the $z$-coordinates of the complex solution set. This is done with the function fgb_matrixn.

```
> param:=fgb_matrixn_radical(F,0,[z,y,x]);
```

On this example, the concrete output is

```
[
6940480*z-56720*x^7+25352*x^6+12756*x^5-14940*x^4+5626*x^3-
936*x^2+32*x+5,
13880960*y-1343680*x^7+1848208*x^6-1016328*x^5+284036*x^4-
42824*x^3+3676*x^2-222*x+9,
3470240*x^8-4582784*x^7+2519968*x^6-750640*x^5+139000*x^4-
19216*x^3+2202*x^2-146*x+1
]
```

We set below $c_1 = 6940480$, $c_2 = 13880960$

$$q_1 = -56720\,x^7 + 25352\,x^6 + 12756\,x^5 - 14940\,x^4 + 5626\,x^3 - 936\,x^2 + 32\,x + 5,$$
$$q_2 = -1343680\,x^7 + 1848208\,x^6 - 1016328\,x^5 + 284036\,x^4 - 42824\,x^3 + 3676\,x^2 - 222\,x + 9$$

and

$$q_3 = 3470240\,x^8 - 4582784\,x^7 + 2519968\,x^6 - 750640\,x^5 + 139000\,x^4 - 19216\,x^3 + 2202\,x^2 - 146\,x + 1.$$

Note that $q_3$ has degree 8 which is the number of complex solutions counted with multiplicities found using fgb_hilbert.

This output must me interpreted as follows: the complex solution is defined by the following representation

$$q_3(x) = 0, \qquad z = q_1(x)/c_1 q_0(x), \qquad y = q_2(x)/c_2 q_0(x)$$

where $q_0 = \frac{\partial q_3/\partial x}{\deg(q_3)}$. Note that isolating the roots of $q_3$ with sufficient accuracy allows to isolate the roots of the system.

### 1.2 Infinitely many solutions.

We focus now on what can be done for systems with infinitely many complex solutions; in this case, one says that the system has positive dimension. The following commands which, as above, use fgb_hilbert allow to compute the dimension $d$ and the *degree* of the ideal generated a given polynomial system.

```
> F:=[x^2-2*x*u+u^2+y^2-2*y*v+v^2-1, u*y+v*u+v*x, x^2+y^2+v^2+u^2-1, w*v-1]:


> gb:=fgb_gbasis(F, 0, [], [u,v,w,x,y]):
> hilb:=fgb_hilbert(gb, 0, [], [u,v,w,x,y],'z');

[2*z^3+5*z^2+4*z+1, 1]

#Degree
> deg:=subs(z=1, hilb[1]);

12
#Dimension
> dim:=hilb[2];

1
```

The above degree is the number of points counted with multiplicity obtained when augmenting the polynomial system with $d$ generically chosen linear forms.

```
> forms:=[seq(randpoly([u,v,w,x,y],degree=1,dense),i=1..dim)]:
> gb0:=fgb_gbasis([op(F), op(forms)], 0, [], [u,v,w,x,y]);
> hilb0:=fgb_hilbert(gb0, 0, [], [u,v,w,x,y],'z');

[2*z^3+5*z^2+4*z+1, 1]
```

```
#Degree
> deg0:=subs(z=1, hilb0[1]);

12
#Dimension
> dim0:=hilb0[2];

0
```

As aleady mentioned, when running all the above computatations over a prime field $\mathbb{Z}/p\mathbb{Z}$ for a prime $p$ chosen after a finite subset of prime numbers.

Eliminating variables is useful in positive dimension: it allows to compute a subset of polynomials which define, up to taking closure, the projection of the complex solution on the space of the remaining variable. Elimination orderings are crucial for this operation.

```
> gb:=fgb_gbasis(F, 0, [u,v,w], [x,y]):

[u^2-2*w*y^3+w*y-2*x^2+y^2, v*u+2*v*x+2*w*x*y^2-w*x+x*y,
v^2-1+2*w*y^3-w*y+3*x^2, w*v-1, 2*w^2*x*y^2-w^2*x+u+w*x*y+2*x,
2*w^2*y^3-w^2*y+v+3*w*y^2-w+3*y, x*u+x^2-y^2,
u*y-v*x-2*w*x*y^2+w*x-x*y, v*x^2+2*w*y^4-w*y^2+2*x^2*y+y^3-y,
y*v-x^2+y^2, 2*w*y^5-w*y^3+x^4+x^2*y^2+y^4-y^2, w*x^2-w*y^2-y,
x^6-x^2*y^2+y^6]
```

Note that in the above example, one can extract expressions of $u, v, w$ with respect to $x, y$ by looking at the polynomials which depend only on $w, x, y$, $v, x, y$ and $u, x, y$ (they have degree 1 in $w$, $v$ and $u$ respectively). This is not always true but rather common.

Note also when one wants to compute only the projection and when computations are performed over $\mathbb{Q}$, it is faster to use fgb_gbasis_elim.

```
> proj:=fgb_gbasis_elim(F, 0, [u,v,x], [x,y]);

[x^6-x^2*y^2+y^6]
```

This function can be used for the implicitization of curves or surfaces:

```
> F1:=[x+77+45*u+34*v-94*u^2-67*v*u-95*v^2,
       y-2*u+4*v-5*v*u+40*v^2-5*u^3-4*u^2*v,
       z+38*v+50*u^2+75*v^3-11*u^4-80*u*v^3-14*v^4]:

> impl1:=fgb_gbasis_elim(F1, 0, [u,v], [x,y,z]):
```

Though the runtime for the above call is less than a second on a modern computer, its output is too large to be printed here: one polynomial of degree 12 with 169 monomials.

Using fgb_gbasis with an elimination ordering for the implicitization problem is not appropriate: since it computes the whole Gröbner basis, the runtime is much larger than the one of fgb_gbasis_elim.

## 2 Installation instructions

**Requirements:**

- Unix/Linux: libc?
- Mac OS: xcode?

## 3 Basic commands

### 3.1 fgb_gbasis(F,char,vars1,vars2,opts)

**Input description.**

- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 is a list of variables (vars1 may be empty);
- vars2 is a list of variables ;
- opts is an optional argument (see Section 4).

**Assumptions.**

- F has either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables in F should appear either in vars1 or in vars2 (else an error is raised) ;
- vars1 and vars2 must have an empty intersection.

**Output description.** The command returns a Gröbner basis of the ideal generated by F with respect to the elimination ordering DRL(vars1)>DRL(vars2).

**Examples**

```
> F:=[x, x*y-1];
> fgb_gbasis(F, 0, [], [x,y]);

[1]
> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];
:
> fgb_gbasis(F, 9001, [], [x,y,z]):
[z^4+1476*z^3+4294*x*z+8165*y*z+3056*z^2+8840*y+6072*z,
x*z^2+1574*z^3+7195*x*z+684*y*z+1608*z^2+5742*y+2730*z,
y*z^2+8380*z^3+8846*x*z+8691*y*z+5975*z^2+388*y+4190*z,
x^2+8101+6304*x*z+7202*y*z+4953*z^2+5400*x+3150*y+8100*z,
x*y+8999*x*z+9000*y*z+8999*z^2+4501*z,
y^2+4502*x*z+2*y*z+2252*z^2+2250*y+4500*z]
```

### 3.2 fgb_gbasis_lm(F,char,vars1,vars2,opts)

**Input description.**

- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 is a list of variables ;
- vars2 is a list of variables ;
- opts is an optional argument (see Section 4).

**Assumptions.**

- F has either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables in F should appear either in vars1 or in vars2 (else an error is raised) ;
- vars1 and vars2 must have an empty intersection.

**Output description.** The command returns a list of two elements [gb, lm] where

- gb is a Gröbner basis of the ideal generated by F with respect to the elimination ordering DRL(vars1)>DRL(vars2).
- lm is the list of leading monomials of the polynomials in gb

**Examples**

```
> fgb_gbasis_lm([x,x*y-1], 0, [], [x,y]);

[[1],[1]]

> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];
:
> fgb_gbasis_lm(F, 9001, [], [x,y,z]):
[[z^4+1476*z^3+4294*x*z+8165*y*z+3056*z^2+8840*y+6072*z,
x*z^2+1574*z^3+7195*x*z+684*y*z+1608*z^2+5742*y+2730*z,
y*z^2+8380*z^3+8846*x*z+8691*y*z+5975*z^2+388*y+4190*z,
x^2+8101+6304*x*z+7202*y*z+4953*z^2+5400*x+3150*y+8100*z,
x*y+8999*x*z+9000*y*z+8999*z^2+4501*z,
y^2+4502*x*z+2*y*z+2252*z^2+2250*y+4500*z],
[z^4, x*z^2, y*z^2, x^2, x*y, y^2]]
```

### 3.3 fgb_gbasis_elim(F,char,vars1,vars2,opts)

**Input description.**

- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 is a list of variables ;
- vars2 is a list of variables ;
- opts is an optional argument (see Section 4).

**Assumptions.**

- F has either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables in F should appear either in vars1 or in vars2 (else an error is raised) ;
- vars1 and vars2 must have an empty intersection.

**Output description.** The command returns a Gröbner basis with respect to the ordering DRL(vars2) of the ideal obtained by eliminating vars1 from the one generated by F.

**Examples**

```
> F:=[x+28*u-16*v-30*u*v, y-72+87*u^2-47*u*v, z+48-53*u+28*u^2]:
> fgb_gbasis_elim(F, 0, [],[]);

[865928*x^2*z+618110485632-1105440*x*y*z+352800*y^2*z+3434760*x*z^2-
2192400*y*z^2+3406050*z^3+41564544*x^2-68684672*x*y+29716736*y^2+
234388201*x*z-243520594*y*z+653661404*z^2+4461842352*x-5321983120*y+
39759575952*z]

> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1]:
> fgb_gbasis_elim(F, 9001, [],[]);

[z^7+8361*z^6+7278*z^5+6894*z^4+1051*z^3+4625*z^2+5358*z]
```

### 3.4 fgb_hilbert(gb,char,vars1,vars2,v)

**Input description.**

- gb is a list of polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 is a list of variables ;
- vars2 is a list of variables ;
- v is an used variable.

**Assumptions.**

- char is the charactistic of the field over which gb has been computed (char is 0 when rational coefficients have been used, else it is a prime field of characteristic $< 2^{16}$) ;
- gb is a Gröbner basis with respect to the ordering DRL(vars1)>DRL(vars2) ;
- v is an used variable.

**Output description.** The command returns a list of two elements

- the first one is a univariate polynomial $P$ which is the Hilbert polynomial of the ideal generated by gb ;
- the second one is a natural integer dim which is the dimension of the ideal gb.

Note that, as a consequence, the Hilbert series of the ideal generated by gb is given by $\frac{P}{(1-\mathsf{v})^{\mathsf{dim}}}$.

**Examples**

```
> fgb_hilbert([1],0,[], [x,y]);

> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y, 2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];
:
> lgb:=fgb_gbasis_lm(F, 9001, [], [x,y,z]):
> fgb_hilbert(lgb[1],9001,[],[x,y,z],'u');

[u^3+3*u^2+3*u+1, 0]
> fgb_hilbert(lgb[2],9001,[],[x,y,z], 'u');

[u^3+3*u^2+3*u+1, 0]
```

### 3.5 fgb_interface()

**Input description.** This function rakes no argument

**Output description.** It returns a list of 5 elements as follows:

- an integer giving the maple release number ;
- a string giving the operating system ;
- an expression `"FGb_modp" = <number>` where `<number>` gives the build number of FGb_modp.
- an expression `"FGb_INT" = <number>` where `<number>` gives the build number of FGb_INT.
- an integer giving the FGb maple interface release number.

**Examples**

```
> fgb_interface();
[16.01, "APPLE UNIVERSAL OSX", "FGb_modp" = 12801, "FGb_INT" = 12802, 1.62]
```

### 3.6 fgb_matrixn_radical(F,char,vars,num,opts)

**Input description.**

- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars is a list of variables ;
- num is a natural number ;
- opts is an optional argument (see Section 4).

**Assumptions.**

- F has either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- vars is the list of variables appearing in F.

The assumptions below are satisfied in generic coordinates (i.e. they are true up to substituting the last variable with a random linear combination of vars) and checked by the function.

- The solution set $Z$ of F (in the algebraic closure) is finite ;
- The set of polynomials that vanish on $Z$ is in the so-called shape lemma position, i.e. for the lexicographical ordering, the Gröbner basis has the following form:

$$\mathsf{vars}[1] - t_1(\mathsf{vars}[\mathsf{n}]), \mathsf{vars}[2] - t_2(\mathsf{vars}[\mathsf{n}]), \ldots, \mathsf{vars}[\mathsf{n}-1] - t_{n-1}(\mathsf{vars}[\mathsf{n}]), t_n(\mathsf{vars}[\mathsf{n}]).$$

- Another technical assumption is requested on the DRL-Gröbner basis of the set of polynomials that vanish on $Z$ (see [6]).

**Output description.** A list of polynomials of the form

$$c_1\mathsf{vars}[1] - q_1(\mathsf{vars}[\mathsf{n}]), c_2\mathsf{vars}[2] - q_2(\mathsf{vars}[\mathsf{n}]), \ldots, c_{n-1}\mathsf{vars}[\mathsf{n}-1] - q_{n-1}(\mathsf{vars}[\mathsf{n}]), q_n(\mathsf{vars}[\mathsf{n}])$$

where the $c_i$'s are coefficients and the $q_i$'s are univariate polynomials in vars[n]. These polynomials describe the set $Z'$ defined by

$$\mathsf{vars}[\mathsf{i}] = \frac{q_i(\mathsf{vars}[\mathsf{n}])}{c_i q_0(\mathsf{vars}[\mathsf{n}])} \quad \text{for } 1 \leq i \leq \mathsf{n}-1, \quad q_n(\mathsf{vars}[\mathsf{n}]) = 0$$

where $q_0 = \frac{\partial q_n / \partial \mathsf{vars}[n]}{\deg(q_n)}$.

When num is 0, $Z'$ equals $Z$, else $Z'$ is the set of points of multiplicity num in $Z$.

**Examples**

```
> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];

> param:=fgb_matrixn_radical(F,0,[z,y,x]);

[
6940480*z-56720*x^7+25352*x^6+12756*x^5-14940*x^4+5626*x^3-
936*x^2+32*x+5,
13880960*y-1343680*x^7+1848208*x^6-1016328*x^5+284036*x^4-
42824*x^3+3676*x^2-222*x+9,
3470240*x^8-4582784*x^7+2519968*x^6-750640*x^5+139000*x^4-
19216*x^3+2202*x^2-146*x+1
]
```

---

## 3.7 pseudo_fgb_normalForm(gb,F,char,vars1,vars2)

---

**Input description.**

- gb is a list of polynomials ;
- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 is a list of variables ;
- vars2 is a list of variables ;

---

**Assumptions.**

- gb and F have either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables in gb and F should appear either in vars1 or in vars2 (else an error is raised) ;

- vars1 and vars2 must have an empty intersection.
- gb is a Gröbner basis with respect to the ordering DRL(vars1)>DRL(vars2).

---

**Output description.** A list of the form

$$[c_1, p_1, c_2, f_2, \ldots, c_k, p_k]$$

where $k$ is the cardinality of F the $c_i$'s are coefficients and the $p_i$'s are polynomials such that $p_i$ is the normal form of $c_i f_i$ (where $f_i$ is the $i$-th element of F.

---

**Examples**

>

# 4 Advanced usage

We describe now options, given as elements of a set, that are available for fgb_gbasis, fgb_gbasis_lm, fgb_gbasis_elim and fgb_matrixn_radical and more advanced functions for expert users.

## 4.1 Options

Options are given as a set of the following form

$$\{< \mathsf{string}_1 >=< \mathsf{num}_1 >, \ldots, < \mathsf{string}_k >=< \mathsf{num}_k >\}$$

where $< \mathsf{string}_i >$ is a string and $< \mathsf{num}_i >$ is a natural number.

**Verbosity :** "verb"=<num>**.**
Here <num> is an integer in $\{0, 1, 2, 3\}$ and it controls the verbosity of FGb. Its default value is 0.

Informations that are printed are related to the size of the matrices appearing during the Gröbner bases reconstructions, progress in linear algebra routines (for Gröbner bases computations and algorithms for change of orderings) and rational reconstruction.

**Size of matrices :** "index"=<num>**.**
Recall that algorithms for Gröbner bases in FGb are based on row echelon form computations. The default limit value for the number of rows/columns matrices generated by FGb is limited to 500 000.

That may not be sufficient for very large computations and the user can increase this value (up to the limit of the available memory on its computer). Note also that increasing this value when it is not necessary may slow down the run time of FGb.

**Degree restrictions :** "verb"=<num>.
Recall also that algorithms for computations of Gröbner bases in FGb are performed by increasing degree until the requested Gröbner basis is obtained.

One can limit the degree up to which the computations are run. The limit degree is instantiated to the integer <num>.

---

**Example:**

```
> fgb_gbasis(F, 0, [], {\sf vars}, {``verb''=3, ``index''=1000000}):
```

---

## 4.2 Advanced functions

---

### 4.3 fgb_multi(F1,F2,vars1,vars2,vars3,opts)

---

**Input description.**

- F1 and F2 are lists of multivariate polynomials with rational coefficients ;
- vars1, vars2 and vars3 are lists of variables ;
- num is a natural number ;
- opts is an optional argument (see Section 4).

---

**Assumptions.**

- F1 and F2 have either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables appearing in F1 or F2 must appear in either vars1 or vars2 ;
- vars1 is the list of variables appearing in F2 ;
- vars1 and vars2 must have an empty intersection.

---

**Output description.** This function simulates the following sequence of computations provided that all assumptions for fgb_matrix_radical are satisfied by the system newF below.

```
> gb:=fgb_gbasis_elim(F1, 0, vars1, [op(vars2), op(vars3)]):
> newF:=[op(gb), op(F2)]:
> gb:=fgb_gbasis_elim(newF, 0, vars2, vars3):
```

The function returns gb.

**Examples**

```
> pol:=3844*x^4-10092*x^2*y^2+2804*x^2*y*z+5041*x^2*z^2+1496*x*y^2*z-
2414*x*y*z^2+9409*y^4-14162*y^3*z+5618*y^2*z^2+1488*x^2*y+8804*x^2*z+
7216*x*y^2-20792*x*y*z+11360*x*z^2-10864*y^3+10964*y^2*z-2720*y*z^2-
6944*x^2-10168*x*y+9920*x*z+26738*y^2-25822*y*z+6400*z^2-9744*y+7569;

> F1:=F1:=[L*diff(pol,x)-1, L*diff(pol,y)-1, L*diff(pol,z)-1]:

> F2:=[pol]:

> param:=fgb_multi(F1,F2,[L],[x,y],[z]):
```

## 4.4 fgb_matrixn_radical2(F1,F2,char,vars1,vars2,num,opts)

**Input description.**

- F1 and F2 are lists of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars1 and vars2 are lists of variables ;
- num is a natural number ;
- opts is an optional argument (see Section 4).

**Assumptions.**

- F1 and F2 have either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- All variables appearing in F1 or F2 must appear in either vars1 or vars2 ;
- vars1 is the list of variables appearing in F2 ;
- vars1 and vars2 must have an empty intersection.

**Output description.** This function simulates the following sequence of computations provided that all assumptions for fgb_matrix_radical are satisfied by the system newF below.

```
> gb:=fgb_gbasis_elim(F1, char, vars1, vars2):
> newF:=[op(gb), op(F2)]:
> param:=fgb_matrix_radical(newF, char, vars2, num):
```

The function returns param.

---

**Examples**

```
> pol:=3844*x^4-10092*x^2*y^2+2804*x^2*y*z+5041*x^2*z^2+1496*x*y^2*z-
2414*x*y*z^2+9409*y^4-14162*y^3*z+5618*y^2*z^2+1488*x^2*y+8804*x^2*z+
7216*x*y^2-20792*x*y*z+11360*x*z^2-10864*y^3+10964*y^2*z-2720*y*z^2-
6944*x^2-10168*x*y+9920*x*z+26738*y^2-25822*y*z+6400*z^2-9744*y+7569;

> F1:=F1:=[L*diff(pol,x)-1, L*diff(pol,y)-1, L*diff(pol,z)-1]:

> F2:=[pol]:

> param:=fgb_matrixn_radical2(F1,F2,0,[x,y,z],0):

> param1:=fgb_matrixn_radical2(F1,F2,0,[x,y,z],1):

> param2:=fgb_matrixn_radical2(F1,F2,0,[x,y,z],2):
```

---

## 4.5 fgb_matrixn(F,char,vars,opts)

---

**Input description.**

- F is a list of multivariate polynomials ;
- char is a natural number indicating the characteristic of the ground field; for computations over the rationals, char is 0 ;

  If char is 0 computations are performed over $\mathbb{Q}$ else they are performed over $\mathbb{Z}/\mathsf{char}\mathbb{Z}$.
- vars is a list of variables ;
- num is a natural number ;
- opts is an optional argument (see Section 4).

---

**Assumptions.**

- F has either rational coefficients or coefficients in a prime field of characteristic $< 2^{16}$ ;
- vars is the list of variables appearing in F.

The assumptions below are satisfied in generic coordinates (i.e. they are true up to substituting the last variable with a random linear combination of vars) and checked by the function.

- The solution set $Z$ of F (in the algebraic closure) is finite ;
- The set of polynomials that vanish on $Z$ is in the so-called shape lemma position, i.e. for the lexicographical ordering, the Gröbner basis has the following form:

$$\mathsf{vars}[1] - t_1(\mathsf{vars}[\mathsf{n}]), \mathsf{vars}[2] - t_2(\mathsf{vars}[\mathsf{n}]), \ldots, \mathsf{vars}[\mathsf{n}-1] - t_{n-1}(\mathsf{vars}[\mathsf{n}]), t_n(\mathsf{vars}[\mathsf{n}]).$$

- Another technical assumption is requested on the DRL-Gröbner basis of the set of polynomials that vanish on $Z$ (see [6]).

---

**Output description.** A list of polynomials of the form

$$c_1\mathsf{vars}[1] - q_1(\mathsf{vars}[\mathsf{n}]), c_2\mathsf{vars}[2] - q_2(\mathsf{vars}[\mathsf{n}]), \ldots, c_{n-1}\mathsf{vars}[\mathsf{n}-1] - q_{n-1}(\mathsf{vars}[\mathsf{n}]), q_n(\mathsf{vars}[\mathsf{n}])$$

where the $c_i$'s are coefficients and the $q_i$'s are univariate polynomials in vars[n]. These polynomials describe the set $Z'$ defined by

$$\mathsf{vars}[\mathsf{i}] = \frac{q_i(\mathsf{vars}[\mathsf{n}])}{c_i q_0(\mathsf{vars}[\mathsf{n}])} \quad \text{for } 1 \leq i \leq \mathsf{n}-1, \quad q_n(\mathsf{vars}[\mathsf{n}]) = 0$$

where $q_0 = \frac{\partial q_n / \partial \mathsf{vars}[n]}{\deg(q_n)}$; hence when there is no multiple root for F, $Z = Z'$.

---

**Examples**

```
> F:=[2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
2*x*y-2*y*z-4*x*z-4*z^2+z,
2*x*z+z^2-4*x*y-4*y^2-4*y*z+y,
10*x^2+10*y^2+10*z^2+16*x*y+16*x*z-6*x+16*y*z-6*y-6*z+1];

> param:=fgb_matrixn(F,0,[z,y,x]);

[6940480*z-56720*x^7+25352*x^6+12756*x^5-14940*x^4+5626*x^3-
936*x^2+32*x+5,
13880960*y-1343680*x^7+1848208*x^6-1016328*x^5+284036*x^4-
42824*x^3+3676*x^2-222*x+9,
3470240*x^8-4582784*x^7+2519968*x^6-750640*x^5+139000*x^4-
19216*x^3+2202*x^2-146*x+1]
```

# References


[1] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.

[2] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4).-. *Journal of Pure and Applied Algebra*, 139(1–3):61–88, 1999.

[3] J.-C. Faugère. A new efficient algorithm for computing Gröbner without reduction to zero (F5). In *Proceedings of ISSAC 2002*, pages 75 – 83. ACM Press, 2002.

[4] J.-C. Faugère. FGb: A Library for Computing Grbner Bases. In Komei Fukuda, Joris Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 84–87, Berlin, Heidelberg, September 2010. Springer Berlin / Heidelberg.

[5] J.-C. Faugère and S. Lachartre. Parallel Gaussian Elimination for Grbner bases computations in finite fields. In M. Moreno-Maza and J.L. Roch, editors, *Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, PASCO '10, pages 89–97, New York, NY, USA, July 2010. ACM.

[6] J.-C. Faugère and C. Mou. Fast Algorithm for Change of Ordering of Zero-dimensional Grbner Bases with Sparse Multiplication Matrices. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 115–122, New York, NY, USA, 2011. ACM.

[7] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.