MGrobner Package

Peter Stadelmeyer

April 28, 1997

Abstract

MGrobner is a MAPLE V Package for doing Gröbner basis and syzygy computation not only for ideals of polynomials but for modules over polynomial rings. The package consists of five procedures: `mgbasis` and `mgbasisx` to compute the Gröbner basis of a module, `mnormalf` to get the normal form of a polynomial tuple modulo a module, `msolveGB` and `msolveSP` to solve a system of linear equations with polynomial coefficients (syzygy computation).

1 Problem Statement

Whenever we talk about the module of F we mean the submodule of the free module over $K[X]$ generated by F. Elements $\vec{f}$ of the free module are represented as column vectors (i.e.$\vec{f} = (f_1, \ldots, f_r)^T$, where $f_j \in K[X]$).

Some notation:

K	a field (in our case the rational numbers)
$[X]$	stands for $[x_1, \ldots, x_n]$ the set of all power products in $x_1, \ldots, x_n$
$[X]^r$	the set of power product tuples (i.e. the set of all r-tuples of the form $(0, \ldots, 0, x_1^{i_1} \cdots x_n^{i_n}, 0, \ldots, 0)$ and $i_1, \ldots, i_n \in \mathbf{N}_0$)
$K[X]$	denotes $K[x_1, \ldots, x_n]$, the ring of all n-variate polynomials over K
$K[X]^r$	the free module over $K[X]$ of dimension r
$\vec{f}, \vec{g}, \vec{f_i}$	elements of $K[X]^r$
F, G, H	finite subsets of $K[X]^r$
Module(F)	the module generated by F
GB(F)	the normed reduced Gröbner basis of (the module generated by) F
nf$(\vec{f}, G)$	the normal form of $\vec{f}$ modulo G, where G is a normed reduced Gröbner basis

In the following, a few problems are listed which can be solved using the MGrobner Package.

1.1 Classical Gröbner basis problems

Problem I: Equality of submodules

given: F and H

decide: Module(F)=Module(H)?

method: compute $G_F = \mathrm{GB}(F)$ and $G_H = \mathrm{GB}(H)$. Then $\mathrm{Module}(F) = \mathrm{Module}(H) \Leftrightarrow G_F = G_H$.

Problem II: Submodule membership

given: F and $\vec{f}$

decide: Is $\vec{f}$ an element of Module(F)?

method: compute $G = \mathrm{GB}(F)$. Then $\vec{f} \in \mathrm{Module}(F) \Leftrightarrow \mathrm{nf}(\vec{f}, G) = 0$.

Problem III: Congruence modulo a submodule

given: $F, \vec{f}, \vec{h}$

decide: Is $\vec{f} \equiv_F \vec{h}$?

method: compute $G = \mathrm{GB}(F)$. Then $\vec{f} \equiv_F \vec{h} \Leftrightarrow \mathrm{nf}(\vec{f}, G) = \mathrm{nf}(\vec{h}, G)$ or $\mathrm{nf}(\vec{f} - \vec{h}, G) = 0$.

1.2 Solving systems of linear equations with coefficients in $K[X]$

Problem IV: Solve a system of linear homogeneous equations

given: $\vec{f_1}, \ldots, \vec{f_s}$

find: all **syzygies** of $\vec{f_1}, \ldots, \vec{f_s}$ i.e. all s-tuples $\vec{z} = (z_1, \ldots, z_s)^T \in K[X]^s$ which satisfy

$$\vec{f_1} z_1 + \cdots + \vec{f_s} z_s = 0 \tag{1}$$

or equivalent (written in matrix notation)

$$\begin{pmatrix} f_{11} \\ \vdots \\ f_{r1} \end{pmatrix} z_1 + \cdots + \begin{pmatrix} f_{1s} \\ \vdots \\ f_{rs} \end{pmatrix} z_s = \begin{pmatrix} f_{11} & \cdots & f_{1s} \\ \vdots & \ddots & \vdots \\ f_{r1} & \cdots & f_{rs} \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

method: since it is obvious that the syzygies constitute a module, we will look for a generating *basis* of this module. Hilbert's Basissatz (see [Hilbert], p. 210) guarantees, that there always exists a finite basis. Therefore our goal is to find finitely many s-tuples of polynomials $\vec{y_1}, \ldots, \vec{y_m}$ such that

1. $\vec{y_i}$ is a solution of (1), for $i = 1, \ldots, m$
2. every other solution $\vec{z}$ of (1) can be written as

$$\vec{z} = p_1 \vec{y_1} + \cdots + p_m \vec{y_m}, \qquad p_1, \ldots, p_m \in K[X]$$

As shown in section 2 there are two different methods to compute $\vec{y_1}, \ldots, \vec{y_m}$. They are implemented as `msolveGB` and `msolveSP`, respectively.

Problem V: Solve a system of linear inhomogeneous equations

given: $\vec{f_1}, \ldots, \vec{f_s}$ and $\vec{f_0}$

find: all s-tuples $\vec{z} = (z_1, \ldots, z_s)^T \in K[X]^s$ which satisfy

$$\vec{f_1} z_1 + \cdots + \vec{f_s} z_s = \vec{f_0} \tag{2}$$

method: to get all solutions of (2) we have to find a basis $\vec{y_1}, \ldots, \vec{y_m}$ of syzygies of $\vec{f_1}, \ldots, \vec{f_s}$ (see Problem IV) and a particular solution $\vec{y_0}$ of (2). All solutions of (2) can then be written as

$$\vec{z} = y_0 + p_1 \vec{y_1} + \cdots + p_m \vec{y_m}, \qquad p_1, \ldots, p_m \in K[X]$$

1.3 Resolution of modules — Chain of syzygies

Problem VI:

This problem is thoroughly described in [Hilbert] on pp.215-233.

given: a system of linear homogeneous equations with coefficients in $K[X]$:

$$\vec{f}_1^{(1)} z_1^{(1)} + \cdots + \vec{f}_{m^{(1)}}^{(1)} z_{m^{(1)}}^{(1)} = 0 \qquad , \vec{f}_i^{(1)} \in K[X]^m \tag{3}$$

find: the chain of syzygies, which is defined as follows:

- let $\vec{f}_1^{(2)}, \ldots, \vec{f}_{m^{(2)}}^{(2)}$ be a basis of the submodule of syzygies of (3).
- the *derived system of linear equations* is then defined as:

$$\vec{f}_1^{(2)} z_1^{(2)} + \cdots + \vec{f}_{m^{(2)}}^{(2)} z_{m^{(2)}}^{(2)} = 0 \qquad , \vec{f}_i^{(2)} \in K[X]^{m^{(1)}} \tag{4}$$

- compute a basis of the submodule of syzygies of (4) and build again the derived system
- iterate this, until a system of equations has no solution.

method: compute a basis of syzygies and take the result as input for the next computation. Hilbert proved in [Hilbert](pp. 217) that this process has to terminate, at least after the n^{th} step, where n is the number of variables.

2 Mathematical Foundations

The concept of Gröbner bases was originally introduced for polynomial ideals (see [Buchberger]). But as shown in [Mora, Möller] and [Furukawa et al] it can naturally be extended for submodules of the free module over $K[X]$. In the following we will show briefly and without proofs how this generalization can be done. (It will be a summary of section 4.2 and 4.3 of [Wall].)

2.1 Gröbner bases for modules over $K[X]$

To be able to generalize the notions of reduction step, normal form, S-polynomial and Gröbner basis, we first have to define an admissible ordering $<_{M(T)}$ of $[X]^r$. Let $<_T$ be an admissible ordering of $[X]$ (e.g. **pure lexicographic** or **total degree**). Furthermore, let $\vec{p} := (0,\ldots,0,p_i,0,\ldots,0)^T$ and $\vec{q} := (0,\ldots,0,q_j,0,\ldots,0)^T$ be two elements of $[X]^r$. Assume that p_i and q_j occur at the i^{th} position in $\vec{p}$ and j^{th} position in $\vec{q}$ respectiveley. Two examples of such orderings $<_{M(T)}$ are:

- the **term-first-ordering** based on an ordering $<_T$ of $[X]$.

  $$\vec{p} <_{M(T)} \vec{q} :\Leftrightarrow [p_i <_T q_j \vee (p_i =_T q_j \wedge i > j)]$$

  Example: ($<_T$ ... pure lexicographic $y <_T x$)

  $$\binom{0}{1} <_{M(T)} \binom{1}{0} <_{M(T)} \binom{0}{y} <_{M(T)} \binom{y}{0} <_{M(T)} \cdots <_{M(T)} \binom{0}{x} <_{M(T)} \binom{x}{0} <_{M(T)} \cdots$$

- the **index-first-ordering** based on an ordering $<_T$ of $[X]$.

  $$\vec{p} <_{M(T)} \vec{q} :\Leftrightarrow [i > j \vee (i = j \wedge p_i <_T q_j)]$$

  Example: ($<_T$ ... pure lexicographic $y <_T x$)

  $$\binom{0}{1} <_{M(T)} \binom{0}{y} <_{M(T)} \cdots <_{M(T)} \binom{0}{x} <_{M(T)} \cdots <_{M(T)} \binom{1}{0} <_{M(T)} \binom{y}{0} <_{M(T)} \cdots$$

Fixing such an admissible ordering of $[X]^r$, every nonzero r-tuple of polynomials $\vec{f}$ has a unique representation

$$\begin{aligned}
\vec{f} \;&=\; \sum_{i=1}^{m} c(\vec{f},\vec{p_i}) \cdot \vec{p_i} \qquad , c(\vec{f},\vec{p_i}) \in K \setminus \{0\}, \vec{p_i} \in [X]^r \\
&\; \vec{p_1} <_{M(T)} \vec{p_2} <_{M(T)} \cdots <_{M(T)} \vec{p_m} \\
&\; \vec{p_m} = (0,\ldots,0,p_k,0,\ldots,0), \text{ where } p_k \text{ occurs at the } k^{th} \text{ position}
\end{aligned}$$

and we can define:

$\vec{p_m}$	leading power product tuple of $\vec{f}$	$\text{lppt}(\vec{f})$
p_k	leading power product of $\vec{f}$	$\text{lpp}(\vec{f})$
$c(\vec{f},\vec{p_i})$	coefficient of $\vec{p_i}$ in $\vec{f}$	$\text{coeff}(\vec{f},\vec{p_i})$
$c(\vec{f},\vec{p_m})$	leading coefficient of $\vec{f}$	$\text{lc}(\vec{f})$
k	head position	$\text{hp}(\vec{f})$

Definition [Reduction step] Let $F \subseteq K[X]^r$ (F finite), $\vec{g}, \vec{h} \in K[X]^r$. We say: $\vec{g}$ reduces to $\vec{h}$ w.r.t. F ($\vec{g} \rightarrow_F \vec{h}$) iff there are $\vec{f} \in F \setminus \{0\}, u \in [X]$, such that $\mathrm{coeff}(\vec{g}, u \cdot \mathrm{lppt}(\vec{f})) =: c \neq 0$, and

$$\vec{h} = \vec{g} - \frac{c}{\mathrm{lc}(\vec{f})} \cdot u \cdot \vec{f}$$

Definition [Normal form] Let $F \subseteq K[X]^r$ (F finite), $\vec{h} \in K[X]^r$. We say that $\vec{h}$ is irreducible modulo F ($\vec{h}$ is in normal form modulo F) if and only if there exists no $\vec{h'} \in K[X]^r$ such that $\vec{h} \rightarrow_F \vec{h'}$. $\vec{h}$ is a normal form of $\vec{g}$ modulo F iff $\vec{g} \rightarrow_F^* \vec{h}$ and $\vec{h}$ is irreducible modulo F.

Definition [S-polynomial] Let $\vec{f}, \vec{g} \in K[X]^r$. The S-polynomial of $\vec{f}$ and $\vec{g}$ (S-pol($\vec{f}, \vec{g}$)) is defined as

$$\text{S-pol}(\vec{f}, \vec{g}) = \begin{cases} u \cdot \vec{f} - \frac{\mathrm{lc}(\vec{f})}{\mathrm{lc}(\vec{g})} \cdot v \cdot \vec{g} & \text{if } \mathrm{hp}(\vec{f})\text{=}\mathrm{hp}(\vec{g}) \\ 0 & \text{otherwise} \end{cases}$$

where u, v are such that $\mathrm{lcm}(\mathrm{lpp}(\vec{f}), \mathrm{lpp}(\vec{g})) = u \cdot \mathrm{lpp}(\vec{f}) = v \cdot \mathrm{lpp}(\vec{g})$.

Theorem I Let M be a submodule of $K[X]^r$, $G = \{\vec{g_1}, \ldots, \vec{g_t}\}$ a finite subset of M. The following conditions are equivalent:

1. G is a Gröbner basis for M.
2. $\vec{f} \rightarrow_G^* 0$ for all $\vec{f} \in M$.
3. For all $\vec{f}, \vec{g} \in M$: If $\vec{f} - \vec{g} \in M$ and $\vec{f}$ and $\vec{g}$ are in normal form modulo G, then $\vec{f} = \vec{g}$.
4. $\rightarrow_G^*$ has the Church-Rosser-proerty.
5. S-pol$(\vec{g_i}, \vec{g_j}) \rightarrow_G^* 0$ for all $1 \leq i < j \leq t$.

A proof can be found in [Mora, Möller].

2.2 Systems of linear equations over $K[X]$

First we attack Problem IV not for arbitrary $\vec{f_1}, \ldots, \vec{f_s}$, but we make the additional assumption that we have a Gröbner basis $G := \{\vec{g_1}, \ldots, \vec{g_t}\}$ of which we want to compute a basis of syzygies. Such a basis can then be obtained by reducing all the S-polynomials of G to 0 modulo G.

Theorem II Let $\{\vec{g_1}, \ldots, \vec{g_t}\}$ be a Gröbner basis. For all $1 \leq i < j \leq t$ with $\mathrm{hp}(\vec{g_i}) = \mathrm{hp}(\vec{g_j})$ let $u_{ij}, v_{ij}, k_{ij}^1, \ldots, k_{ij}^t \in K[X]$ be such that

$$\text{S-pol}(\vec{g_i}, \vec{g_j}) = u_{ij}\vec{g_i} + v_{ij}\vec{g_j} = k_{ij}^1 \vec{g_1} + \cdots + k_{ij}^t \vec{g_t},$$

where the k_{ij}^l are extracted from the reduction of S-pol$(\vec{g_i}, \vec{g_j})$ to 0. For all $1 \leq i \leq t$ let $\vec{e_i}$ be the t-tuple $(0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 occurs at the i^{th} position. Then

$$\{u_{ij}\vec{e_i} + v_{ij}\vec{e_j} - (k_{ij}^1, \ldots, k_{ij}^t) \mid 1 \leq i < j \leq t, \mathrm{hp}(\vec{g_i}) = \mathrm{hp}(\vec{g_j})\}$$

is a basis for the module of syzygies of $\vec{g_1}, \ldots, \vec{g_t}$.

A proof is given in [Furukawa et al].

We show now, how we can use this result to compute a basis of syzygies of an arbitrary finite set $F := \{\vec{f_1}, \ldots, \vec{f_s}\}$ of r-tuples of polynomials. In this case, we first have to compute the Gröbner basis G of F, apply Theorem II on G and then transform the syzygies of G back to syzygies of F. To do this transformation we need linear representations of the elements of F in terms of elements of G and vice versa. For this purpose, we define what a transformation matrix between two bases is.

Definition [Transformation matrix] Let $F = \{\vec{f_1}, \ldots, \vec{f_s}\}$ and $G = \{\vec{g_1}, \ldots, \vec{g_t}\}$ be bases for the same module (i.e. Module(F)=Module(G)). A matrix $X = (x_{ij})$ of polynomials is called a transformation matrix from F to G iff

$$\begin{pmatrix} \vec{g_1} \\ \vdots \\ \vec{g_t} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1s} \\ \vdots & \ddots & \vdots \\ x_{t1} & \cdots & x_{ts} \end{pmatrix} \cdot \begin{pmatrix} \vec{f_1} \\ \vdots \\ \vec{f_s} \end{pmatrix}$$

The following theorem describes now, how a basis for the module of syzygies of the Gröbner basis G can be transformed to a basis for the module of syzygies of F.

Theorem III Let $F = \{\vec{f_1}, \ldots, \vec{f_s}\}$ be a set of r-tuples of polynomials and $G = \{\vec{g_1}, \ldots, \vec{g_t}\}$ a Gröbner basis for Module(F). Let the rows of the matrix R be a basis for the module of syzygies of G, and let X and Y be transformation matrices from F to G and G to F respectively (i.e. $G = X \cdot F$ and $F = Y \cdot G$). Then the rows of the matrix Q are a basis for the syzygies of F, where

$$Q = \begin{pmatrix} I_s - Y \cdot X \\ \cdots\cdots\cdots\cdots \\ R \cdot X \end{pmatrix}$$

and I_s is the identity matrix with s rows and columns.

For a detailed proof see [Winkler].

3 Algorithm Description

In this section a short overview is given of how the algorithms for computing the Gröbner basis and syzygies are implemented and which improvements were made to speed them up.

3.1 Some general remarks

Most subprocedures in MGrobner have two versions, a normal one and an X-version. An X-version of a procedure is basically the same procedure, but with one extension: for each polynomial tuple an additional reduction vector (for constructing the transformation matrix) is stored.

Throughout the package a few *datastructures* (implemented as lists) are used. There are always two forms of one datastructure, depending on whether or not the datastructure is used in an X-version of a procedure.

normal version	ext.poly.	[hpoly,hht,hhp]
	crit.pair	[LCM,hp,p1poly,p1ht,p2poly,p2ht]
X-version	ext.poly.	[hpoly,hht,hhp,hX]
	crit.pair	[LCM,hp,p1poly,p1ht,p1X,p2poly,p2ht,p2X]
`sbasisSP` **version**	crit.pair	[LCM,hp,i,j]

where the components of the above lists have the following meaning

*poly	the polynomial tuple (represented as a list)
*ht	the leading power product
*hp,hp	the head position
*X	the reduction vector
LCM	the l.c.m. of the leading power products
i,j	indices of the critical pair

3.2 The algorithm for computing the Gröbner basis

Theorem I (5) gives the main idea for the algorithms of `mgbasis` and `mgbasisx`. To hold the number of critical pairs small, the *chain criterion* is used. A proof of the chain criterion can be found in [Mora, Möller].

Algorithm of `mgbasis` (and `mgbasisx`)

input: $F \subseteq K[X]^r$

output: a Gröbner basis G of F

MAPLE V Code	description
	build up the list B of critical pairs
`G := [P[1]];`	$G := \{\vec{p_i}\}$ s.t. $\mathrm{lppt}(\vec{p_i})$ is max. wrt. $<_{M(T)}$
`P := subsop(1=NULL, P);`	$P := P \setminus \{\vec{p_i}\}$
`B := [];`	$B := \{\}$
`while P<>[] do`	**while** $P \neq \{\}$ **do**
`  h := P[1];`	$h \in P$ s.t. $\mathrm{lppt}(\vec{h})$ is max. wrt. $<_{M(T)}$
`  P := subsop(1=NULL, P);`	$P := P \setminus \{\vec{h}\}$

```
  B1 := mgrobner[newCP](h,G,X);

  B2 := mgrobner[oldCP](h,B,X);
  B := mgrobner[sortCP]([op(B1),op(B2)],X);
  G := mgrobner[updateG](h,G,X);
od;
```

$B_1 := \{(\vec{g}, \vec{h}) | \vec{g} \in G, \mathrm{hp}(\vec{g}) = \mathrm{hp}(\vec{h}), \neg((\exists \vec{p} \in G)\mathrm{crit}(\vec{p}, \vec{g}, \vec{h}))\}$
$B_2 := \{(\vec{f}, \vec{g}) | (\vec{f}, \vec{g}) \in B, \neg\mathrm{crit}(\vec{h}, \vec{f}, \vec{g}))\}$
$B := B_1 \cup B_2$, sort B ascending
$G := \{\vec{f} \in G | \mathrm{lppt}(\vec{h}) \not\leq_M \mathrm{lppt}(\vec{f})\} \cup \{\vec{h}\}$
od

main loop

```
while B<>[] do
  pair := B[1];
  B := subsop(1=NULL, B);
  h := mgrobner[spol](pair,X);
  h := mgrobner[reduce](h,G,X);
  if not(mgrobner[zeropoly](h[1])) then
    B1 := mgrobner[newCP](h,G,X);

    B2 := mgrobner[oldCP](h,B,X);
    B := mgrobner[sortCP]([op(B1),op(B2)],X);
  G := mgrobner[updateG](h,G,X);
  fi;
od;
```

while $B \neq \{\}$ **do**
$(\vec{f}, \vec{g}) \in B$
$B := B \setminus \{(\vec{f}, \vec{g})\}$
$\vec{h} := \text{S-pol}(\vec{f}, \vec{g})$
$\vec{h} :=$ reduced form of $\vec{h}$
if $\vec{h} \neq 0$ **then**
$B_1 := \{(\vec{g}, \vec{h}) | \vec{g} \in G, \mathrm{hp}(\vec{g}) = \mathrm{hp}(\vec{h}), \neg((\exists \vec{p} \in G)\mathrm{crit}(\vec{p}, \vec{g}, \vec{h}))\}$
$B_2 := \{(\vec{f}, \vec{g}) | (\vec{f}, \vec{g}) \in B, \neg\mathrm{crit}(\vec{h}, \vec{f}, \vec{g}))\}$
$B := B_1 \cup B_2$, sort B ascending
$G := \{\vec{f} \in G | \mathrm{lppt}(\vec{h}) \not\leq_M \mathrm{lppt}(\vec{f})\} \cup \{\vec{h}\}$
fi
od

where $\mathrm{crit}(\vec{h}, \vec{f}, \vec{g})$ stands for the chain criterion and is defined as:

$$\begin{aligned}
\mathrm{crit}(\vec{h}, \vec{f}, \vec{g}) \quad :\Leftrightarrow \quad & [\mathrm{hp}(\vec{h}) = \mathrm{hp}(\vec{f}) = \mathrm{hp}(\vec{g}), \\
& \mathrm{lpp}(\vec{h}) \leq_M \mathrm{lcm}(\mathrm{lpp}(\vec{f}), \mathrm{lpp}(\vec{g})), \\
& \mathrm{lcm}(\mathrm{lpp}(\vec{h}, \vec{f})) \neq \mathrm{lcm}(\mathrm{lpp}(\vec{f}), \mathrm{lpp}(\vec{g})), \\
& \mathrm{lcm}(\mathrm{lpp}(\vec{h}, \vec{g})) \neq \mathrm{lcm}(\mathrm{lpp}(\vec{f}), \mathrm{lpp}(\vec{g}))]
\end{aligned}$$

3.3 The algorithm for computing a basis for the module of syzygies

The MGrobner package provides two different methods for computing a basis of syzygies:

The first method (**msolveSP**) is a direct application of Theorem II and III (for a detailed algorithm description see [Wall](Alg. 8). First a Gröbner basis along with a transformation matrix is computed (using **mgbasisx**). Then the S-polynomials are reduced to zero. To get a smaller basis (and a faster algorithm) the chain criterion can be used again to detect unnecessary S-polynomials, which do not have to be reduced. Again, a correctness proof can be found in [Wall](Thm. 16).

The other method (**msolveGB**) is based on Theorem 3 in [Wall] and works as follows: One computes a Gröbner basis of a set of tuples of polynomials along with the corresponding transformation matrix. If an S-polynom can be reduced to 0 during the GB-computation, the reduction vector is stored in a list S. When the algorithm terminates, S contains a basis for the module of syzygies. This is described as Alg. 9 in [Wall].

4 Description of the MGrobner Procedures

4.1 mgbasis – compute the normed reduced Gröbner basis

Calling Sequence:

`mgbasis`(F,X)
`mgbasis`(F,X,*torder1*)
`mgbasis`(F,X,*torder1*,*torder2*)

Parameters:

- F – list of polynomial tuples
- X – list of indeterminates
- *torder1* – power-product-tuple ordering: `term` (term first) or `index` (index first – default)
- *torder2* – power-product ordering: `plex` (pure lexicographic) or `tdeg` (total degree – default)

Synopsis:

- The command `mgbasis`(F,X,*torder1*,*torder2*) computes the normed reduced Gröbner basis of F with respect to the indeterminates X and the given orderings. The algorithm used in `mgbasis` is described as Alg.13 in [Wall].
- The polynomial tuples in F are represented as lists of polynomials.
- If X has the form `[x1,x2,...,xn]`, then using the pure lexicographic ordering this is interpreted as `x1` $>$ `x2` $> \ldots >$ `xn`. Within the total degree ordering, ties are broken by inverse lexicographic order.

Examples:

```
> F := [[x*y-1,x+2],[y^2+x+1,y-1]]:
> mgbasis(F,[x,y]);

           2                                        2
       [[x   + x + y, - x - 2 y], [x y - 1, x + 2], [y   + x + 1, y - 1],

                2              2
            [0, x   + x y + 2 y   + 3 x + y + 1]]

> mgbasis(F,[y,x],term,plex);

                  3         2          2                 2
            [[1/4 x   + 1/4 x   + 1/4, y   + 1/2 y + 1/4 x   + 5/4 x],

                      3          2                  2
                [- 1/2 x   - 1/2 x   - 1/2, y x + 1/2 x   + 1/2 x + 1],

                    2
                [y + x   + x, - 2 y - x]]
```

4.2 mgbasisx - compute the normed reduced Gröbner basis along with the transformationmatrix

Calling Sequence:

`mgbasisx`(F,'G','GX',X)
`mgbasisx`(F,'G','GX',X,*torder1*)
`mgbasisx`(F,'G','GX',X,*torder1*,*torder2*)

Parameters:

F – list of polynomial tuples
G – the normed reduced Gröbner basis (output)
GX – the transformation matrix (i.e. $GX \cdot F = G$) (output)
X – list of indeterminates
torder1 – power-product-tuple ordering: `term` (term first) or `index` (index first – default)
torder2 – power-product ordering: `plex` (pure lexicographic) or `tdeg` (total degree – default)

Synopsis:

- The command `mgbasisx`(F,'G','GX',X,*torder1*,*torder2*) computes the normed reduced Gröbner basis of F with respect to the indeterminates X and the given orderings. The result is also returned in the variable G. In GX the transformation matrix is stored. The algorithm used in `mgbasisx` is basically the same as used in `mgbasis`, but the reduction steps are stored to get the transformation matrix GX.
- The polynomial tuples in F are represented as lists of polynomials.
- If X has the form `[x1,x2,...,xn]`, then using the pure lexicographic ordering this is interpreted as `x1` $>$ `x2` $> \ldots >$ `xn`. Within the total degree ordering, ties are broken by inverse lexicographic order.

Examples:

```
> F := [[x*y-1,x+2],[y^2+x+1,y-1]]:
> mgbasisx(F,'G','GX',[x,y]):
> G;

       2                                        2
    [[x  + x + y, - x - 2 y], [x y - 1, x + 2], [y  + x + 1, y - 1],

            2             2
        [0, x  + x y + 2 y  + 3 x + y + 1]]

> GX;

                                          2
           [[- y, x], [1, 0], [0, 1], [y  + x + 1, - x y + 1]]

> mgbasisx(F,'G','GX',[y,x],term,plex):
```

```
> G;

            3         2          2                 2
   [[1/4 x  + 1/4 x  + 1/4, y  + 1/2 y + 1/4 x  + 5/4 x],

                 3        2                    2
       [- 1/2 x  - 1/2 x  - 1/2, y x + 1/2 x  + 1/2 x + 1],

             2
       [y + x  + x, - 2 y - x]]

> GX;

          2                                                  2
  [[1/2 y  + 1/4 + 1/2 x - 1/4 y x, - 1/2 y x + 1/2 + 1/4 x ],

                               2
      [1/2 y x + 1/2, - 1/2 x ], [- y, x]]
```

4.3 mnormalf – compute the normal form of a tuple of polynomials modulo a submodule

Calling Sequence:

`mnormalf`(f,F,X)
`mnormalf`(f,F,X,*torder1*)
`mnormalf`(f,F,X,*torder1*,*torder2*)

Parameters:

f	– a tuple of polynomials
F	– list of polynomial tuples (a Gröbner basis)
X	– list of indeterminates
torder1	– power-product-tuple ordering: `term` (term first) or `index` (index first – default)
torder2	– power-product ordering: `plex` (pure lexicographic) or `tdeg` (total degree – default)

Synopsis:

- The command `mnormalf`(f,F,X,*torder1*,*torder2*) computes the normal form of the tuple of polynomials f modulo the module generated by F with respect to the indeterminates X and the given orderings.
- Note: F must be a Gröbner basis . Therefore, usually first a Gröbner basis for a set of polynomial tuples is computed (with `mgbasis`) and afterwards `mnormalf` is called.
- If X has the form `[x1,x2,...,xn]`, then using the pure lexicographic ordering this is interpreted as `x1` $>$ `x2` $> \ldots >$ `xn`. Within the total degree ordering, ties are broken by inverse lexicographic order.

Examples:

```
> F:=[[x-1,x*y-1],[x*y-1,y],[y,x-1]]:
> G1:=mgbasis(F,[x,y]):
> mnormalf([x^3*y^2+x^3+1,y^3+x+1],G1,[x,y]);

                        2               2
                 [0, x  + 6/5 x y + 2/5 y  - 8/5 x - 2/5 y - 3/5]

> G2:=mgbasis(F,[x,y],term):
> mnormalf([x^3*y^2+x^3+1,y^3+x+1],G2,[x,y],term);

                                              2
                        [- 3 x + 3/2 y + 1/2, y  + 3/2 y]
```

4.4 msolveGB - compute a basis for the module of syzygies

Calling Sequence:

`msolveGB`(F,*rhsf*,X)
`msolveGB`(F,*rhsf*,X,*torder1*)
`msolveGB`(F,*rhsf*,X,*torder1*,*torder2*)

Parameters:

F – list of polynomial tuples
rhsf – right-hand-side of the linear system
X – list of indeterminates
torder1 – power-product-tuple ordering: `term` (term first) or `index` (index first – default)
torder2 – power-product ordering: `plex` (pure lexicographic) or `tdeg` (total degree – default)

Synopsis:

- The command `msolveGB`(F,*rhsf*,X,*torder1*,*torder2*) computes a basis for the module of syzygies using the extended Gröbner-basis-algorithm as described in [Wall] (Alg.9 + Alg.13). The procedure returns a list with two elements: the first element is a particular solution, the second one is a basis for the module of syzygies (represented as a list).
- The polynomial tuples in F are represented as lists of polynomials.
- F and *rhsf* have to be interpreted as:

  e.g.: `F:=[[f11,f21],[f12,f22],[f13,f23]]; rhsf:=[f1,f2,f3];` yields the linear system

$$\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

- If X has the form `[x1,x2,...,xn]`, then using the pure lexicographic ordering this is interpreted as `x1` $>$ `x2` $> \ldots >$ `xn`. Within the total degree ordering, ties are broken by inverse lexicographic order.

Examples:

```
> F := [[x^3-y,-x+y],[x*y-1,x^3-1],[-x+y^2,y-1]]:
> S:=msolveGB(F,[0,0],[x,y]):
> nops(S[2]);

                                    4

> mgbasis(S[2],[x,y]);

                   3  2    4      2          2
                 [[x  y  - x  - x y  + x y - y  + x + y - 1,

                    3      3      2    3    2          2
                   x  y - x  + x y  - y  - x  + x y - y  + y,

                      6    3      3    2         2
                   - x  + x  y + x  - x  y + x y  + x - 2 y]]

> msolveGB(F,[x*y^2-y^2,-x^5+x^2-x+y^2],[x,y],term,tdeg);

              2           3  2    4      2          2
      [[- y, x , - x], [[x  y  - x  - x y  + x y - y  + x + y - 1,

                          3      3      2    3    2          2
                         x  y - x  + x y  - y  - x  + x y - y  + y,

                            6    3      3    2         2
                         - x  + x  y + x  - x  y + x y  + x - 2 y]] ]
```

4.5 msolveSP - compute a basis for the module of syzygies

Calling Sequence:

`msolveSP`(F,*rhsf*,X)
`msolveSP`(F,*rhsf*,X,*torder1*)
`msolveSP`(F,*rhsf*,X,*torder1*,*torder2*)

Parameters:

F – list of polynomial tuples
rhsf – right-hand-side of the linear system
X – list of indeterminates
torder1 – power-product-tuple ordering: `term` (term first) or `index` (index first – default)
torder2 – power-product ordering: `plex` (pure lexicographic) or `tdeg` (total degree – default)

Synopsis:

- The command `msolveSP`(F,*rhsf*,X,*torder1*,*torder2*) computes a basis for the module of syzygies. This procedure is based on Thm.16 and Thm.11 in [Wall]. The procedure returns a list with two elements: the first element is a particular solution, the second one is a basis for the module of syzygies (represented as a list).

- The polynomial tuples in F are represented as lists of polynomials.
- F and *rhsf* have to be interpreted as:

  e.g. `F:=[[f11,f21],[f12,f22],[f13,f23]]; rhsf:=[f1,f2,f3];` yields the linear system

$$\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

- The polynomial tuples in F are represented as lists of polynomials.
- If X has the form `[x1,x2,...,xn]`, then using the pure lexicographic ordering this is interpreted as `x1` $>$ `x2` $> \ldots >$ `xn`. Within the total degree ordering, ties are broken by inverse lexicographic order.

Examples:

```
> F := [[x^3-y,-x+y],[x*y-1,x^3-1],[-x+y^2,y-1]]:
> S:=msolveSP(F,[0,0],[x,y]):
> nops(S[2]);

                                   3

> mgbasis(S[2],[x,y]);

                 3  2    4      2          2
              [[x  y  - x  - x y  + x y - y  + x + y - 1,

                   3      3      2    3    2          2
                  x  y - x  + x y  - y  - x  + x y - y  + y,

                     6    3      3    2        2
                  - x  + x  y + x  - x  y + x y  + x - 2 y]]

> msolveSP(F,[x*y^2-y^2,-x^5+x^2-x+y^2],[x,y],term,tdeg);

           2           4    3  2                       2        2
    [[- y, x , - x], [[x  - x  y  - x y - y + 1 + x y  - x + y ,

                             3      3    2           3    2          2
                          - x  y + x  + x  - x y + y  + y  - y - x y ,

                                3          2        2   3      6
                          - x - x  + 2 y + x  y - x y  - x  y + x ]]   ]
```

5 Examples

5.1 A review of Problem II:

We assume, that $F \subset K[X]^2$ and $\vec{f} \in K[X]^2$ are given and we want to decide, whether $\vec{f}$ is an element of Module(F) or not.

$$F = \left\{ \begin{pmatrix} x-1 \\ xy-1 \end{pmatrix}, \begin{pmatrix} xy-1 \\ y \end{pmatrix}, \begin{pmatrix} y \\ x-1 \end{pmatrix} \right\}$$
$$\vec{f} = \begin{pmatrix} x^2+y \\ x^4-x^3-x^2y-xy^2+y+x^2-x \end{pmatrix}$$

```
> F:=[[x-1,x*y-1],[x*y-1,y],[y,x-1]];

               F := [[x - 1, x y - 1], [x y - 1, y], [y, x - 1]]

> G:=mgbasis(F,[x,y]);

                 2                3      2
        G := [[1, x  - x - y], [0, x  - 2 x  - 2 x y + x + y + 1],

                2            2                    2    2
            [0, x  y - x y - y  - x + 1], [0, x y  - x  + 2 x - y - 1],

                3    2
            [0, y  + x  - 2 x y - 2 x + y + 2]]

> f:=[x^2+y,x^4-x^3-x^2*y-x*y^2+y+x^2-x]:
> mnormalf(f,G,[x,y]);

                                   [0, 0]
```

The normal form of $\vec{f}$ modulo G is 0, so $\vec{f} \in$ Module(F).

5.2 A first syzygy computation

We want to solve the following linear system (cf. [Winkler](p. 401)):

$$\begin{pmatrix} yw-x^2 & zw-yx & zx-y^2 \\ x^2-xw & zw-yx & zy-zw \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

$$\begin{aligned} f_1 &:= 2zw+y^3w-y^2x^2-2yx \\ f_2 &:= 2zw+y^3x^2w-2y^2x^4+2y^2x^3w-2y^2x^2w^2+y^2x^2-y^2xw-2yx^5+5yx^4w-2yx^3w^2-2yx \end{aligned}$$

As an ordering of the set of power product tuples we fix the index first ordering based on the pure lexicographic ordering ($z > y > x > w$). The example shows how to solve such a system using the procedures `msolveGB` or `msolveSP`. (The Gröbner bases of the two solutions must be identical)

```
> f1:=[y*w-x^2,x^2-x*w]:
> f2:=[z*w-y*x,z*w-y*x]:
> f3:=[z*x-y^2,z*y-z*w]:
> F:=[f1,f2,f3]:
> f:=[2*z*w+y^3*w-y^2*x^2-2*y*x,2*z*w+y^3*x^2*w-2*y^2*x^4+2*y^2*x^3*w-
      2*y^2*x^2*w^2+y^2*x^2-y^2*x*w-2*y*x^5+5*y*x^4*w-2*y*x^3*w^2-2*y*x]:
> S:=msolveGB(F,f,[z,y,x,w],index,plex);

S := [

  2      2          2      2  2              2            2    3
[y  - z x  w + 2 y x  w - y  x  + z y x w + y  x w - x z w  - x  y,

          3      3      2             2    4       3      2  2        2
   2 - 2 x  w + x  y - y  x w + y x w  + x , - 2 x  w + x  w  + y x w ],

          2           2          2           2                         2  2
[[- 1/2 w y  z - 1/2 z x  y + 1/2 z  x w - 1/2 z  y w - 1/2 z y x w + 1/2 z  w

              3          2
      + 1/2 x y  + 1/2 z x y ,

          2           2          2          2         2  2           2
    1/2 w y  z - 1/2 z x  y - 1/2 y  x w + z x  w + 1/2 y  x  - 1/2 y z w

              3
        - 1/2 z x ,

        2         2            2            2         2     3
    1/2 y  x w + z x  w - 1/2 x z w  - 1/2 y z w  + 1/2 y x  w - x  y],

          2           2          2           2                         2  2
 [- 1/2 w y  z - 1/2 z x  y + 1/2 z  x w - 1/2 z  y w - 1/2 z y x w + 1/2 z  w

              3          2
      + 1/2 x y  + 1/2 z x y ,

          2           2          2          2         2  2           2
    1/2 w y  z - 1/2 z x  y - 1/2 y  x w + z x  w + 1/2 y  x  - 1/2 y z w

              3
        - 1/2 z x ,

        2         2            2            2         2     3
    1/2 y  x w + z x  w - 1/2 x z w  - 1/2 y z w  + 1/2 y x  w - x  y]

 ]
```

```
]

> mgbasis(S[2],[z,y,x,w],index,plex);

      2         2         2  2      2         2           2              3
   [[z  y w - z  x w - z  w  - z y  x + z y  w + z y x  + z y x w - y  x,

           2           2         2      3          2      2  2    2
       - z y  w + z y x  + z y w  + z x  - 2 z x  w - y  x  + y  x w,

           2         2           2    2            3       2
       z y w  - 2 z x  w + z x w  - y  x w + 2 y x  - y x  w]]

> S:=msolveSP(F,f,[z,y,x,w],index,plex);

 S := [

      2      2           2      2  2                2             2       3
    [y  - z x  w + 2 y x  w - y  x  + z y x w + y  x w - z x w  - y x ,

              3          3    2               2    4         3       2  2        2
      2 - 2 x  w + y x  - y  x w + y x w  + x , - 2 x  w + x  w  + y x w ],

      2         2         2  2      2         2           2              3
    [[z  y w - z  x w - z  w  - z y  x + z y  w + z y x  + z y x w - y  x,

           2           2         2      3          2      2  2    2
       - z y  w + z y x  + z y w  + z x  - 2 z x  w - y  x  + y  x w,

           2         2           2    2            3       2
       z y w  - 2 z x  w + z x w  - y  x w + 2 y x  - y x  w]]             ]

> mgbasis(S[2],[z,y,x,w],index,plex);

      2         2         2  2      2         2           2              3
   [[z  y w - z  x w - z  w  - z y  x + z y  w + z y x  + z y x w - y  x,

           2           2         2      3          2      2  2    2
       - z y  w + z y x  + z y w  + z x  - 2 z x  w - y  x  + y  x w,

           2         2           2    2            3       2
       z y w  - 2 z x  w + z x w  - y  x w + 2 y x  - y x  w]]
```

A basis of the submodule of syzygies is

$$\left\{ \left(\begin{array}{c} z^2yw - z^2xw - z^2w^2 - zy^2x + zy^2w + zyx^2 + zyxw - y^3x \\ -zy^2w + zyx^2 + zyw^2 + zx^3 - 2zx^2w - y^2x^2 + y^2xw \\ zyw^2 - 2zx^2w + zxw^2 - y^2xw + 2yx^3 - yx^2w \end{array} \right) \right\}$$

and a particular solution

$$\left(\begin{array}{c} y^2 - zx^2w + 2yx^2w - y^2x^2 + zyxw + y^2xw - xzw^2 - x^3y \\ 2 - 2x^3w + x^3y - y^2xw + yxw^2 + x^4 \\ -2x^3w + x^2w^2 + yxw^2 \end{array} \right)$$

5.3 A second syzygy computation

The following example is also taken from [Winkler](p. 394):

$$\left(\begin{array}{ccc} xz - xy^2 - 4x^2 - \frac{1}{4} & y^2z + 2x + \frac{1}{2} & x^2z + y^2 + \frac{1}{2}x \end{array} \right) \left(\begin{array}{c} z_1 \\ z_2 \\ z_3 \end{array} \right) = 0$$

```
> f1:=[x*z-x*y^2-4*x^2-1/4]:
> f2:=[y^2*z+2*x+1/2]:
> f3:=[x^2*z+y^2+1/2*x]:
> F:=[f1,f2,f3];

                   2      2           2                   2      2
   F := [[x z - x y  - 4 x  - 1/4], [y  z + 2 x + 1/2], [x  z + y  + 1/2 x]]

> S:=msolveGB(F,[0],[z,y,x],index,plex);

S := [[0, 0, 0],

[

         2         4  2         2  3         4         4         3         2
[1/27 x y  + 2/27 y  x  + 1/27 y  x  + 2/27 y  - 1/27 x  - 4/27 x  - 1/27 x

              5
      - 4/27 x ,

            2  3         2  5           2         2    19  4         6
    - 4/27 y  x  - 2/27 y  x  - 2/27 x y  - 1/18 x  - ---- x  - 8/27 x ,
                                                       54

          2    17  2  2         4  3         4           3         2
    1/54 y  + ---- x  y  + 2/27 y  x  + 2/27 y  x + 1/27 x  + 4/27 x  + 1/27 x
               54
```

```
                 4          2  4
         + 4/27 x  + 8/27 y  x                                                ]

    ,

        4          2    3        2       2          2        2  3      4
[- 1/2 y  - 1/4 x y  + x  + 1/4 x , 3/8 x  + 1/2 x y  + 1/2 y  x  + 2 x ,

           2        4      2              2  2
    - 1/8 y  - 1/2 y  x - x  - 1/4 x - 2 x  y ],

            2  2        3        2        2          2  3       4       2
[3/4 x - 7 x  y  - 5/2 x  - 1/2 y  - 1/2 x  z + 2 z y  x  - 7 z x  + 2 z y  x

           4      2
      + 4 x  + 4 x ,

      3      5                4        6      2
15/2 x  + 7 x  + 1/2 x - 4 z x  - 2 z x  - 2 x  z,

              2      2  3          2                  3        2  4        2  2
- 1/8 - 11/4 x  - 7 y  x  - 1/2 x y  + 1/2 x z + 7 z x  + 2 z y  x  + 2 z y  x

           4      3      5
     - 27 x  + 4 x  + 4 x

],

                   4    2         4      28     2  3         3         2
[- 1/27 x + 8/27 y  z x  + 8/27 y  z - ---- z y  x  - 1/27 x  + 4/27 y
                                         27

              2           2      16     2        2  2    26   4           4
      - 4/27 x  - 2/27 z y  x + ---- x y  - 1/3 x  y  - ---- y  x - 4/27 x
                                 27                      27

         16   2  3
      + ---- y  x ,
         27

    16     2  3         2    5           2      26   2  2    26   2  4
 - ---- z y  x  - 8/27 y  z x  - 8/27 z y  x + ---- x  y  + ---- y  x  - 1/18 x
    27                                          27           27

         19   3         5
      - ---- x  - 8/27 x ,
         54
```

```
          28     2  2    26    4  2         4            4    3        3
 4/27 x + ---- z y   x  - ---- y   x  + 8/27 y  z x + 8/27 y  z x  + 4/27 x
          27              27

             2          2          2       16    2  2    100  2  3    16   2  4
      + 1/27 x  - 5/54 x y  + 2/27 y  z + ---- x  y  - --- y  x  + ---- y  x
                                           27            27          27

      + 1/27                                                              ]

 ,

[

      2  2           2  2  3          2  4          2  2                       4
16/5 x  z  - 64/5 z  y  x  + 224/5 z  x  - 64/5 z  y  x - 24/5 x z - 128/5 z x

           3          2              2  2          2
     + 16 z x  - 128/5 x  z + 224/5 z y  x  + 16/5 y  z,

       2  4          2  6          2  2         3           5
128/5 z  x  + 64/5 z  x  + 64/5 x  z  - 48 z x  - 224/5 z x  - 16/5 x z,

         2  3                    2          2           2  2  2           2
224/5 z y  x  + 4/5 z + 16/5 z y  x + 88/5 x  z - 64/5 z  y  x  - 16/5 x z

             2  3           3           5            4          2  2  4
     - 224/5 z  x  - 128/5 z x  - 128/5 z x  + 864/5 z x  - 64/5 z  y  x   ],

[

       2  2  3                    2  2           2  3          4
364/5 z  y  x  - 26/5 x + 204/5 z y  x  - 208/5 z y  x  + 338/5 y  z x

            3         2    93   2             2                       2
     - 26/5 x  - 52/5 y  + ---- x  - 208/5 z y  x + 13/5 x z + 18 x y
                            20

            2           2  2          4  2          2  2            4   119   3
     - 52/5 y  z - 52/5 x  y  - 104/5 y  z  + 26/5 z  y  x + 87/5 y  + --- z x
                                                                        10

             4  2  2
     - 104/5 y  z  x ,
```

```
 39         73     3          2  2  3          2  2  5          2  2
---- x z + ---- z x  + 208/5 z  y  x  + 104/5 y  z  x  + 104/5 z  y  x
 10         10

                2  2            2  4           2         2  3    87   2
     - 338/5 z y  x  - 338/5 z y  x  - 87/5 x y  - 87/5 y  x  - ---- x
                                                                 10

        87   4           5
     - ---- x  + 17/5 z x ,
        10

           2  4   441              2  2          4    2             2  3
- 208/5 z y  x  + --- x - 208/5 z y  x  + 338/5 y  z x  + 1387/5 z y  x
                   40

              4  2  3          4  2     459  3    87   2         2
     - 104/5 y  z  x  - 104/5 y  z  x + --- x  + ---- y  - 47/5 x  - 13/5 z
                                         10        20

               2             2   119  2     789  2  2         4
     + 13/2 z y  x - 52/5 x y  - --- x  z + --- x  y  + 87/5 y  x
                                  10         10

              2  2  2         4         2  3         2  2
     - 364/5 z  y  x  - 34/5 x  - 52/5 y  x  - 26/5 y  z  - 13/5           ],

[

      2                           2         2        2  2           3       3
13/2 y  z + 13/4 + 13 x - 49/2 x y  - 35/4 x  + 7 z y  x  - 49/2 z x  + 14 x ,

                          3         2         4        5
13/8 - 13/2 x z - 27/2 z x  + 49/2 x  + 49/2 x  - 7 z x ,

        2         2                  2  2         2          2  3          3
13/2 z y  x + 13 x  - 23/8 x - 49/2 x  y  + 49/2 x  z + 7 z y  x  - 189/2 x

           4
     + 14 x                                                                 ]

]

]
```

```
> nops(S[2]);

                                       7

> mgbasis(S[2],[z,y,x],index,plex);

          2                         2         2
      [[z y  + 2 x + 1/2, - z x + y  x + 4 x  + 1/4, 0],

             2    2                        2         2
          [z x  + y  + 1/2 x, 0, - z x + y  x + 4 x  + 1/4],

            4        2         3         2      2  3    2        4         2
          [y  + 1/2 y  x - 2 x  - 1/2 x , - y  x  - y  x - 4 x  - 3/4 x ,

              4        2  2         2       2
              y  x + 4 y  x  + 1/4 y  + 2 x  + 1/2 x],

                 2    2                 2
          [0, z x  + y  + 1/2 x, - z y  - 2 x - 1/2]]
```

5.4 Resolution of modules

In this subsection three examples are given of how to use `msolveGB` or `msolveSP` to compute a chain of syzygies (Problem VI).

Hilbert's example I ([Hilbert](p. 227))

$F = \{wy - x^2, xy - wz, xz - y^2\}$

```
> f1:=[w*y-x^2]:
> f2:=[x*y-w*z]:
> f3:=[x*z-y^2]:
> F:=[f1,f2,f3];

                                  2                            2
                 F := [[y w - x ], [- w z + x y], [x z - y ]]

> S:=msolveGB(F,[0],[w,x,y,z]);

                    S := [[0, 0, 0], [[z, y, x], [y, x, w]]]

> S:=msolveGB(S[2],[0,0,0],[w,x,y,z]);

                              S := [[0, 0], []]
```

Hilbert's example II ([Hilbert](p. 229))

$F = \{w, x, y, z\}$

```
> F:=[[w],[x],[y],[z]];

                         F := [[w], [x], [y], [z]]

> S:=msolveGB(F,[0],[w,x,y,z]);

    S := [[0, 0, 0, 0],

        [[0, 0, z, - y], [0, z, 0, - x], [z, 0, 0, - w], [0, y, - x, 0],

            [y, 0, - w, 0], [x, - w, 0, 0]]                            ]

> S:=msolveGB(S[2],[0,0,0,0],[w,x,y,z]);

 S := [[0, 0, 0, 0, 0, 0],

     [[- x, y, 0, - z, 0, 0], [- w, 0, y, 0, - z, 0], [0, - w, x, 0, 0, - z],

         [0, 0, 0, - w, x, - y]]                                        ]

> S:=msolveGB(S[2],[0,0,0,0,0,0],[w,x,y,z]);

                      S := [[0, 0, 0, 0], [[w, - x, y, - z]]]

> S:=msolveGB(S[2],[0,0,0,0],[w,x,y,z]);

                               S := [[0], []]
```

Note: Hilbert showed that if F is of the form $F = \{x_1, \ldots, x_n\}$, the s^{th} system of equations consists of $\binom{n}{s-1}$ rows and $\binom{n}{s}$ columns. Furthermore, all coefficients in the derived systems are linear polynomials.

One more example

This example demonstrates, that it can be useful to compute a Gröbner basis G of the basis of syzygies S and to take G instead of S as input for the next step. As one can see, the number of basis elements reduces from 11 to 3 in the first step and therefore the second system becomes much smaller.
$F = \{wy - x^2 + z, w^2z + xy - 1, xy^2 + y + z\}$

```
> f1:=[w*y-x^2+z]:
> f2:=[w^2*z+x*y-1]:
> f3:=[x*y^2+y+z]:
> F:=[f1,f2,f3];

                         2          2                    2
         F := [[y w - x  + z], [w  z + x y - 1], [x y  + y + z]]
```

```
> S:=msolveGB(F,[0],[w,x,y,z]):
> nops(S[2]);

                                      11

> G:=mgbasis(S[2],[w,x,y,z]);

          2                        2         2                        2
 G := [[x y  + y + z, 0, - w y + x  - z], [w  z + x y - 1, - w y + x  - z, 0],

           2             2
     [0, x y  + y + z, - w  z - x y + 1]]

> S:=msolveGB(G,[0,0,0],[w,x,y,z]):
> nops(S[2]);

                                      1

> G:=mgbasis(S[2],[w,x,y,z]);

                   2                   2                   2
           G := [[w  z + x y - 1, - x y  - y - z, - w y + x  - z]]

> S:=msolveGB(G,[0,0,0],[w,x,y,z]);
> nops(S[2]);

                                      0

> S;

                                   [[0], []]
```

5.5 How long does it take?

The following three examples should give a rough idea of how long it takes to compute such bases of syzygies. They also demonstrate, that the evaluation time (in seconds) and the number of basis elements depend strongly on the choosen ordering. The last two columns show the time which is needed to get a Gröbner basis from the basis of syzygies and the number of elements of the Gröbner basis.

The evaluation times were measured on:

- i486-50MHz (PC compatible)
- MAPLE V v1.1 under MS-DOS 5.0

Example I ([Wall](p. 71))

$$\begin{pmatrix} x^3-y & xy-1 & -x+y^2 \\ -x+y & x^3-1 & y-1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

function	ppt ordering	pp ordering	eval. time of `msolve??`	# of basis elements	eval. time of `mgbasis`	# of basis elements
`msolveGB`	index	plex	535.310	9	55.370	1
	index	tdeg	8.611	4	.449	1
	term	plex	18.950	6	4.340	1
	term	tdeg	4.180	1	.110	1
`msolveSP`	index	plex	919.230	4	**	
	index	tdeg	10.720	3	.330	1
	term	plex	27.800	4	92.110	1
	term	tdeg	5.440	1	.220	1

** the algorithm was aborted after 6000 sec

A basis of the submodule of syzygies:

$$\left\{ \begin{pmatrix} x^4 - x^3y^2 + xy^2 - xy - x + y^2 - y + 1 \\ -x^3y + x^3 + x^2 - xy^2 - xy + y^3 + y^2 - y \\ x^6 - x^3y - x^3 + x^2y - xy^2 - x + 2y \end{pmatrix} \right\}$$

Example II ([Wall](p. 71))

$$\begin{pmatrix} x^2+x & y+1 & x+y & xy+1 \\ x+y & y^2+y & x+y & x+y+1 \\ xy+x & y+1 & y & x^3+1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

function	ppt ordering	pp ordering	eval. time of `msolve??`	# of basis elements	eval. time of `mgbasis`	# of basis elements
`msolveGB`	index	plex	51.961	8	10.650	1
	index	tdeg	11.871	5	.441	1
	term	plex	12.510	3	3.350	1
	term	tdeg	6.700	1	.100	1
`msolveSP`	index	plex	124.140	6	**	
	index	tdeg	16.030	6	2.630	1
	term	plex	21.470	3	6.920	1
	term	tdeg	8.950	1	.060	1

** the algorithm was aborted after 6000 sec

A basis of the submodule of syzygies:

$$\left\{ \begin{pmatrix} x^3y^2 - x^3 + x^2y^3 - x^2y + xy^2 - x - y^4 + y^2 - y - 1 \\ x^5 + x^4y - x^3y - 2x^2y^2 - x^2y + 2x^2 - xy^3 + xy^2 + xy + 2y^3 - y \\ -x^4y^2 - x^4y - x^3y^2 + x^3 + x^2y^2 + 2x^2y + x^2 + xy^4 + 2xy^3 - xy^2 - xy + x - y^3 - 2y^2 - y \\ -x^2y - x^2 - xy^2 + x - y^4 + 2y^2 + y \end{pmatrix} \right\}$$

Example III

$$\begin{pmatrix} x^2yz + xy + 1 & y - 1 & z - z^2 \\ y^2 + z - 1 & yz^2 + z - 1 & x - 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

function	ppt ordering	pp ordering	eval. time of `msolve??`	# of basis elements	eval. time of `mgbasis`	# of basis elements
`msolveGB`	index	plex	26.750	12	2.361	1
	index	tdeg	15.270	5	1.160	1
	term	plex	11.711	3	.280	1
	term	tdeg	8.410	2	.220	1
`msolveSP`	index	plex	33.450	8	5.210	1
	index	tdeg	17.850	3	1.270	1
	term	plex	15.330	3	.330	1
	term	tdeg	10.170	2	.279	1

A basis of the submodule of syzygies:

$$\left\{ \begin{pmatrix} xy - x + yz^4 - yz^3 - y + z^3 - 2z^2 + z + 1 \\ -x^3yz + x^2yz - x^2y + xy - x - y^2z^2 + y^2z - z^3 + 2z^2 - z + 1 \\ x^2y^2z^3 + x^2yz^2 - x^2yz + xy^2z^2 + xyz - xy - y^3 + y^2 + yz^2 - yz + y + 2z - 2 \end{pmatrix} \right\}$$

References

[Buchberger] B. Buchberger: *An Algorithmic Method in Polynomial Ideal Theory*, Multidimensional Systems Theory (N. K. Bose, Ed.), D. Reidel Publishing Company, 1985

[Furukawa et al] A. Furukawa, T. Sasaki, H. Kobayashi: *Gröbner basis of a module over $K[x_1, \ldots, x_n]$ and polynomial solutions of a system of linear equations*, Proc. SYMSAC'86 (B. W. Char, Ed.), Waterloo, Canada, 1986

[Hilbert] D. Hilbert: *Über die Theorie der algebraischen Formen*, Math. Ann. 36, pp.473-534 (1890)

[Mora, Möller] F. Mora, H.M. Möller: *New Constructive Methods in Classical Ideal Theory*, Journal of Algebra 100, pp.138-178 (1986)

[Wall] B. Wall: *Computation of Syzygies Solution of Linear Systems over a Multivariate Polynomial Ring*, Publications/Reports RISC-LINZ Series no. 88-86.0, Univ. Linz, 1988

[Winkler] F. Winkler: *Solution of equations I: Polynomial ideals and Gröbner bases*, Conference on Computers and Mathematics, Stanford Univ., 1986

Appendix: Listing of Program