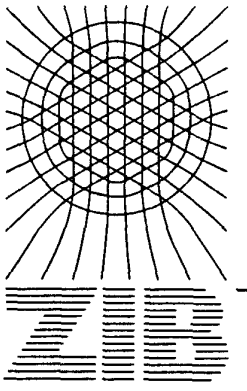
Konrad-Zuse-Zentrum für Informationstechnik Berlin



H. Melenk H.M. Möller* W. Neun

Symbolic Solution of Large Stationary Chemical Kinetics Problems

* Fernuniversität Hagen
FB Mathematik und Informatik
D-5800 Hagen
Federal Republic of Germany

Preprint SC 88-7 (November 1988)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Strasse 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Verwaltungsdruckerei Berlin

ISSN 0933-7911

Contents

Abstract

The paper presents a new application of computer algebra to the treatment of steady states of reaction systems. The method is based on the Buchberger algorithm. This algorithm was modified such that it can exploit the special structure of the equations derived from reaction systems, so even large systems can be handled. In contrast to numerical approximation techniques, the algebraic solution gives a complete and definite overview of the solution space and it is even applicable when parameter values are unknown or undetermined. The algorithm, its adaptation to the problem class and its application to selected examples are presented.

1. Introduction

Usually, chemical reaction systems for mass action kinetics are described by a set of ordinary explicit first order differential equations

$$\dot{c_m} = \sum_R k_{i\to j} \left(\prod_{l=1}^{n} c_l^{y_{il}} \right) (y_{jm} - y_{im}) \ , \ m = 1, \ldots , N \tag{1.1}$$

where c_m is the (time dependent) concentration of a species a_m, R is the set of all elementary reactions (reaction network), $k_{i\to j}$ is the rate constant for the elementary reaction $a_i \to a_j$ and has a nonnegative real constant value, and y_{il} is the stoichiometric coefficient of species a_l in the i^{th} complex [9]. The y_{il} are nonnegative integer constants (usually 0, 1, or 2) and so the right-hand side of (1.1) is a vector of multivariate polynomials with variables c_l.

A steady state of the reaction system is a composition $\{c_m\}$ such that all derivatives $\dot{c_m}$ vanish. So the steady state is described by :

$$0 = \sum_R k_{i\to j} \left(\prod_{l=1}^{n} c_l^{y_{il}} \right) (y_{jm} - y_{im}) \ , \ m = 1, \ldots , N \tag{1.2}$$

This is a system of homogeneous algebraic equations and we are looking for solutions of (1.2) with real values $c_l \geq 0$ or $c_l > 0$ (positive steady state).

Until now solutions of such systems are nearly exclusively sought by numerical methods. The possible advantages of an algebraic approach over the numerical treatment could be:

a) applicability even if some or all constants are unknown (formal parameters),

b) giving a complete and definite answer including the information about the dimension of the solution space (= number of solutions).

FEINBERG analyzes the steady state behaviour of a chemical reaction system in the deficiency zero and deficiency one theorems by analyzing the network structure with graph theoretic methods. The algebraic approach allows to do a similar analysis, even in cases which these theorems do not cover.

The following examples (taken from [9]) may serve as a first illustration. Their processing will be discussed in part 5.

$$\begin{aligned} \dot{c_1} &= \alpha\, c_1 - \beta\, c_1^2 - \gamma\, c_1 c_2 + \epsilon\, c_3 \\ \dot{c_2} &= -\gamma\, c_1 c_2 + (\epsilon + 2\theta) c_3 - 2\eta {c_2}^2 \\ \dot{c_3} &= \gamma c_1 c_2 + \eta c_2^2 - (\epsilon + \theta) c_3 \end{aligned} \tag{1.3}$$

Here the deficiency zero theorem states, that there is only one positive steady state. The algebraic treatment of the homogeneous version of the above system leads to three distinct solutions:

$$\begin{aligned} & c_1 = c_2 = c_3 = 0 \\ & c_1 = \frac{\alpha}{\beta}\,,\ c_2 = c_3 = 0 \\ & c_1 = \frac{\alpha}{\beta}\,,\ c_2 = \frac{\alpha\,\gamma\,\theta}{\beta\,\epsilon\,\eta}\,,\ c_3 = \frac{\alpha^2\,\gamma^2\,\theta}{\beta^2\,\epsilon^2\,\eta^2} \end{aligned} \tag{1.4}$$

So there is only one solution with positive (c_1, c_2, c_3) and the algebraic approach in this case confirms the deficiency zero theorem.

For the second system (EDELSTEIN) the deficiency zero theorem is not applicable:

$$\begin{aligned} \dot{c_1} &= \alpha\,c_1 - \beta c_1^2 - \gamma c_1 c_2 + \epsilon c_3 \\ \dot{c_2} &= -\gamma c_1 c_2 + (\epsilon + \theta) c_3 - \eta c_2 \\ \dot{c_3} &= \gamma c_1 c_2 + \eta c_2 - (\epsilon + \theta) c_3 \end{aligned} \tag{1.5}$$

This system has a one dimensional solution space; taking c_1 as parameter the complete solution is:

$$\begin{aligned} c_3 &= \frac{-\beta\gamma c_1^2 + (\alpha\beta - \beta\eta)c_1 + \alpha\eta}{\gamma\theta c_1 - \epsilon\eta}\,c_1 \\ c_2 &= \frac{(-\beta\epsilon + \beta\theta)c_1 + \alpha\epsilon + \alpha\theta}{\gamma\theta c_1 - \epsilon\eta}\,c_1 \end{aligned} \tag{1.6}$$

which describes c_2 and c_3 as functions of c_1 and shows, that for

$$c_1 = \frac{\epsilon\eta}{\gamma\theta}$$

there is a pole.

The kernel for the algebraic treatment of (1.2) is the Buchberger algorithm [4, 5]. This algorithm transfers a system of multivariate polynomials into a Groebner basis, which gives direct information about the solutions. Although this algorithm has a simple structure, it can achieve great complexity when executed: even for simple looking problems the intermediate results can grow dramatically in number and size, especially if the coefficients are parameters. This effect limits the applicability of the algorithm, even if it is performed by an efficient computer algebra system.

In the case of chemical reaction systems, however, we can find results for systems with a number of variables which usually is too large for the algorithm; additional structure helps to reduce the computational complexity. Particular items are:

a) the low degrees of the polynomials,

b) the sparse pattern of the coefficient matrix,

c) the decomposability of polynomials by factorization,

d) the restriction to nonnegative real solutions.

The purpose of the present paper is to introduce a variant of the Buchberger algorithm which takes advantage of the above properties.

In this paper we describe the Buchberger algorithm and its modifications in detail in the first part (sections 2, 3, and 4). The expository presentation is illustrated by bivariate examples. Its purpose is to enable the reader to use the algorithm for solving of simple examples by hand or, if polynomial manipulations turn out to be complicated, by means of a minicomputer. In the second part (section 5), we give some application examples. These were computed with the Groebner package, now implemented in the computer algebra system REDUCE [10, 12].

2. Multivariate Polynomials and Groebner Bases

Multivariate polynomials are linear combinations of power products, i.e. of terms

$$x_1^{i_1} * \cdots * x_n^{i_n} \text{ with nonnegative integers } i_1, \ldots, i_n .$$

In the bivariate case $n = 2$ we have for instance

$$f_1(x_1, x_2) = x_1^2 + 2x_1x_2^3 - x_2^2 - 7 ,$$

and

$$f_2(x_1, x_2) = x_1x_2 - 2 .$$

These two polynomials will be used in the following for illustrating the definitions and the Groebner basis construction. The coefficients of the multivariate polynomials (here 1, 2, -1, -7 for f_1 and 1, -2 for f_2) are integers, rationals or parameter expressions (multivariate rational functions over the integers), or more generally elements of a field or a unique factorization domain.

All power products occuring with nonzero coefficients in the power product representation of a polynomial f constitute $supp(f)$, *the support of* f. This is a nonvoid set if and only if $f \neq 0$. We have for instance

$$\begin{aligned} supp(f_1) &= \{x_1^2, x_1x_2^3, x_2^2, 1\} , \\ supp(f_2) &= \{x_1x_2, 1\} . \end{aligned}$$

The main tool for the Groebner basis method is that the set of power products is ordered by a relation $<$ satisfying in the bivariate case

$$1 < x_1^i x_2^j, \text{ if } i > 0 \text{ or } j > 0 ,$$

and for arbitrary nonnegative integers a, b, i, j, k, l

$$x_1^i x_2^j < x_1^k x_2^l \text{ implies } x_1^{i+a} x_2^{j+b} < x_1^{k+a} x_2^{l+b} .$$

Examples for such (bivariate) orderings are the lexicographical ordering

$$\begin{array}{ccccccccccc} 1 & < & x_1 & < & x_1^2 & < & \ldots & < & x_1^k & < & \ldots \\ < & x_2 & < & x_1x_2 & < & x_1^2x_2 & < & \ldots & < & x_1^kx_2 & < & \ldots \\ < & x_2^2 & < & x_1x_2^2 & < & x_1^2x_2^2 & < & \ldots & < & x_1^kx_2^2 & < & \ldots \end{array}$$

and the graduated lexicographical ordering

$$1 < x_1 < x_2 < x_1^2 < x_1x_2 < x_2^2 < x_1^3 < \ldots .$$

The multivariate analogy of the lexicographical and the graduated lexicographical ordering are ($<_{lex}$ for short):

$$x_1^{a_1} \cdots x_n^{a_n} <_{lex} x_1^{b_1} \cdots x_n^{b_n} :\Longleftrightarrow$$

$$\exists i \leq n : (a_i < b_i \wedge \forall j > i : a_j = b_j)$$

and ($<_{glex}$ for short):

$$x_1^{a_1} \cdots x_n^{a_n} <_{glex} x_1^{b_1} \cdots x_n^{b_n} :\Longleftrightarrow$$

$$\left(\sum_{i=1}^{n} a_i < \sum_{i=1}^{n} b_i\right) \vee \left(\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i \wedge x_1^{a_1} \cdots x_n^{a_n} <_{lex} x_1^{b_1} \cdots x_n^{b_n}\right) .$$

Using such an ordering, the elements in a support can be ordered; and two polynomials f and g can be compared by saying that f is *simpler than* g if $supp(f)$ and $supp(g)$ are different and $supp(g)$ contains a power product which is greater (w.r.t. $<$) than every power product in $supp(f) \setminus supp(g)$. Both in the lexicographical and the graduated lexicographical ordering, f_2 is simpler than f_1, because $x_1x_2^3$ is greater than x_1x_2, the maximal (and only) power product in $supp(f_2) \setminus supp(f_1)$. In this way any two polynomials with different supports can be compared, but no two ones with same support.

Polynomials can be simplified with respect (to the given ordering $<$ and) to a finite set G of nonzero polynomials. Denoting by $lt(f)$ the maximal power product in $supp(f)$ for $f \neq 0$, we say f *reduces modulo* G *to* f^*, briefly $f \longrightarrow_G f^*$, if a power product $x_1^{a_1} \cdots x_n^{a_n}$ of $supp(f)$ is a multiple of a $lt(g)$ with $g \in G$ and

$$f^* = c_1 f - c_2 \frac{x_1^{a_1} \cdots x_n^{a_n}}{lt(g)} g ,$$

where the constants c_1, c_2 are chosen such that $supp(f^*)$ does not contain $x_1^{a_1} \cdots x_n^{a_n}$. All power products in $supp(f^*)$ greater than $x_1^{a_1} \cdots x_n^{a_n}$ are identical with those in $supp(f)$, but $x_1^{a_1} \cdots x_n^{a_n}$ is contained only in $supp(f)$. Therefore, f^* is simpler than f. This reduction procedure modulo G can be applied iteratively until a polynomial is obtained, which is no more reducible modulo G. (It is easy to show, cf. the survey by BUCHBERGER [5] for the case of polynomials over a field, that every iterated application of a reduction modulo a finite polynomial set G terminates after a finite number of steps at an irreducible polynomial or at the zero polynomial.) We abbreviate by $f \longrightarrow_G^* f^*$ the sequence

$$f \longrightarrow_G g_1 \longrightarrow_G \ldots \longrightarrow_G g_s = f^*,$$

if f^* is 0 or no more reducible modulo G. If f is already irreducible modulo G or if $f = 0$, then we write for short also $f \longrightarrow_G^* f$.

Let for instance $G := \{f_2\}$ and $f := f_1$. Then

$$f = x_1^2 + 2x_1x_2^3 - x_2^2 - 7 \longrightarrow_G x_1^2 + 3x_2^2 - 7 = f_1 - 2x_2^2 f_2 =: f^*$$

with $supp(f^*) = \{x_2^2, x_1^2, 1\}$ and $supp(f) = \{x_1x_2^3, x_2^2, x_1^2, 1\}$ and hence f^* is simpler than f. f^* is irreducible modulo G, because $supp(f)$ contains no power product, which is a multiple of an $lt(g), g \in G$. Therefore, we may also write

$$f \longrightarrow_G^* f^*.$$

When the common zeros of a finite set of polynomials $H = \{h_1, \ldots, h_s\}$ have to be computed, every

$$h = \sum_{i=1}^{s} g_i h_i, \quad \text{with polynomials } g_1, \ldots, g_s$$

vanishes at these common zeros, and conversely the set of all such $h's$

$$I(H) := \left\{ \sum_{i=1}^{s} g_i h_i \mid g_1, \ldots, g_s \text{ polynomials} \right\}$$

has the same set of common zeros like H, because H is a subset of it. $I(H)$ is called *the ideal generated by* H. Therefore, for computing the common zeros of H, another finite set of polynomials generating $I(H)$, a so-called *basis of* $I(H)$, can be taken in place of H. We are dealing with Noetherian rings, where all ideals are generated by a finite subset. Therefore we may speak of an ideal instead of an ideal generated by a finite polynomial set H, if H is of no relevance for the forthcoming.

Definition. A finite set G of nonzero polynomials is called a *Groebner basis* of an ideal I, if for arbitrary $f \in I$

$$f \longrightarrow_G^* 0 .$$

Remark 1. A Groebner basis of an ideal generates in fact the ideal. The Groebner bases depend on the given ideal and also on the given ordering of the power products. But even if ideal and ordering are fixed, the Groebner basis is not unique. We obtain a uniquely determined Groebner basis, if we replace every polynomial g of a Groebner basis G by the (modulo $G \setminus \{g\}$) irreducible $g^*, g \longrightarrow_{G\setminus\{g\}}^* g^*$ and if we normalize the coefficient belonging to $lt(g^*)$ to 1 in case the coefficient domain is a field, cf. [5], or to another special element for other coefficient domains.

Remark 2. If G is a Groebner basis of an ideal I with respect to $<_{lex}$, then every polynomial $f \in I$ depending only on $x_1, \ldots, x_i$ is already reducible to 0 modulo the polynomials in G which depend also only on $x_1, \ldots, x_i$, because the other polynomials in G have lt-parts which are multiples of an $x_j, j > i$ and hence these lt do not divide a power product in $supp(f)$. Thus, the ideal of all polynomials in I depending only on $x_1, \ldots, x_i$ has the set of all polynomials in G depending only on $x_1, \ldots, x_i$ for Groebner basis, $i = 1, \ldots, n-1$.

This is useful for finding the common zeros of the polynomials in I. First the polynomials in G depending only on x_1 are considered. Their common zeros determine all possible first components for the required zeros. Then substituting the first component in all polynomials in G depending only on x_1, x_2 we get a univariate problem for determining all possible second components, etc.

Example. For finding the common zeros of the polynomials f_1, f_2 as introduced above, we consider the ideal I generated by $\{f_1, f_2\}$. As seen before,

$$f_3 := x_1^2 + 3x_2^2 - 7 = f_1 - 2x_2^2 f_2$$

belongs to I. But also

$$f_4 := x_1 f_3 - 3x_2 f_2 = x_1^3 - 7x_1 + 6x_2$$

and

$$f_5 := x_1 f_4 - 6f_2 = x_1^4 - 7x_1^2 + 12$$

are in I. We will see, that $\{f_4, f_5\}$ constitute a Groebner basis of I, when we consider all polynomials as polynomials with rational coefficients. The common zeros of f_1, f_2 can be easily detected by considering f_4 and f_5. The roots of f_5 , $\{2, -2, \sqrt{3}, -\sqrt{3}\}$ are the first components of all possible zeros in common to f_1 and f_2. Inserting these roots into $f_4 = 0$, we get for each choice of x_1 one choice of x_2:

$$(2,1),\ (-2,-1),\ \left(\sqrt{3},\ \frac{2}{3}\sqrt{3}\right), \left(-\sqrt{3},\ -\frac{2}{3}\sqrt{3}\right)$$

are the common zeros of f_1 and f_2.

3. The Computation of Groebner Bases

For the construction of Groebner bases, the notion of *S-polynomials* is essential. For every two nonzero polynomials f, g the S-polynomial is defined by

$$S(f,g) := \frac{c * lcm\{lt(f), lt(g)\}}{c_1 * lt(f)} * f - \frac{c * lcm\{lt(f), lt(g)\}}{c_2 * lt(g)} * g,$$

where c_1 and c_2 denote the coeffients of $lt(f)$ and $lt(g)$ in the respective power product representation of f and g and where $c := lcm\{c_1, c_2\}$. Then the following characterization of Groebner bases for polynomials over fields [5] and over principal ideal rings [19] allows a construction of such bases.

Theorem 1: *A finite set of nonzero polynomials $H = \{h_1, \ldots, h_r\}$ is a Groebner basis of $I(H)$, if and only if $S(h_i, h_j) \longrightarrow_H^* 0$ holds for every $h_i, h_j \in H$.*

Using this theorem, we can show, that the set $\{f_4, f_5\}$ of the example is in fact a Groebner basis. We show first, that $H := \{f_1, \cdots, f_5\}$ is a Groebner basis. This holds if $S(f_i, f_j) \longrightarrow_H^* 0$ for $i < j$ because the remaining S-polynomial reductions are trivial ($S(f_i, f_i) = 0$) or follow by symmetry ($S(f_j, f_i) = -S(f_i, f_j)$).

We have $f_3 = S(f_1, f_2)$. Hence

$$S(f_1, f_2) \longrightarrow_H S(f_1, f_2) - f_3 = 0\,.$$

Similarly

$$S(f_2, f_3) \longrightarrow_H S(f_2, f_3) + f_4 = 0$$

and

$$S(f_2, f_4) \longrightarrow_H S(f_2, f_4) - f_5 = 0\,.$$

More computation is required for the reduction of $S(f_3, f_4)$:

$$\begin{array}{rlcll}
S(f_3,f_4) = & 2f_3 - x_2 f_4 & = & -x_1^3 x_2 + 7x_1 x_2 + 2x_1^2 - 14 & =: p_1 \\
\longrightarrow_H & p_1 + \frac{1}{6}x_1^3 f_4 & = & 7x_1 x_2 + \frac{1}{6}x_1^6 - \frac{7}{6}x_1^4 + 2x_1^2 - 14 & =: p_2 \\
\longrightarrow_H & p_2 - \frac{7}{6}x_1 f_4 & = & \frac{1}{6}x_1^6 - \frac{7}{3}x_1^4 + \frac{61}{6}x_1^2 - 14 & =: p_3 \\
\longrightarrow_H & p_3 - \frac{1}{6}x_1^2 f_5 & = & -\frac{7}{6}x_1^4 + \frac{49}{6}x_1^2 - 14 & =: p_4 \\
\longrightarrow_H & p_4 + \frac{7}{6}f_5 & = & 0. &
\end{array}$$

The remaining tests $S(f_i, f_j) \longrightarrow_H^* 0$ for $i < j$ are as hard as the test of $S(f_3, f_4) \longrightarrow_H^* 0$. We omit these technicalities. (The criteria quoted in the following will show, that these remaining tests are in fact superfluous.)

Therefore, $H = \{f_1, \ldots, f_5\}$ is a Groebner basis. Using $f_3 = S(f_1, f_2)$ and the definition of $S(f_1, f_2)$, we get $f_1 = f_3 + 2x_2^2 f_2$. Hence each reduction step using f_1 for reduction can be replaced by a reduction step using f_3 followed by a reduction step

using f_2. Hence f_1 is *redundant in* H, i.e. $\{f_2, f_3, f_4, f_5\}$ is also a Groebner basis of the ideal $I(H)$. By the same argument, using $f_2 = \frac{1}{6}x_1 f_4 + \frac{1}{6}f_5$, f_2 is redundant in $\{f_2, f_3, f_4, f_5\}$. And using the reduction of $S(f_3, f_4)$ to 0, we find the redundancy of f_3 in $\{f_3, f_4, f_5\}$. Thus $\{f_4, f_5\}$ is a Groebner basis.

The test of a polynomial set for Groebner basis will be simpler, if not all $S(f_i, f_j)$ with $i < j$ have to be tested for being reducible to 0. BUCHBERGER [4] developed criteria for predicting that an S-polynomial reduces to 0. These criteria have been generalized by himself and by other authors.

Definition. Let H be a set of nonzero polynomials, $H := \{f_1, \ldots, f_r\}$. Using the notation $T(f_i, f_j)$ for $lcm\{lt(f_i), lt(f_j)\}$ and $T(f_i)$ for $lt(f_i)$, we say

a) *Criterion D holds for* (f_i, f_s), if f_i and f_s have a common (eventually constant) divisor g, such that $f_i = g * f_i^*, f_s = g * f_s^*$ and $lt(f_i^*), lt(f_s^*)$ without common divisor.

b) *Criterion M holds for* (f_i, f_s), if a $j < s$ exists, such that $T(f_j, f_s)$ divides $T(f_i, f_s)$ and $T(f_j, f_s) \neq T(f_i, f_s)$.

c) *Criterion F holds for* (f_i, f_s), if a $j < i$ exists, such that $T(f_j, f_s) = T(f_i, f_s)$.

d) *Criterion* B_s *holds for* (f_i, f_j), if $s > j$, $f_s \in H$ and $T(f_s)$ divides $T(f_i, f_j)$, but $T(f_i, f_s) \neq T(f_i, f_j) \neq T(f_j, f_s)$.

If one of the criteria M, F, and B_s holds for (f_i, f_j), then the S-polynomial $S(f_i, f_j)$ reduces to 0, as shown in [11]. Criterion D is an easy generalization of Buchberger's criterion 2 [5].

These criteria simplify the Groebner basis computation. In our example, only the S-polynomials $S(f_1, f_2) = f_3, S(f_2, f_3) = -f_4, S(f_2, f_4) = f_5$, and $S(f_3, f_4)$ have to be computed and reduced. For criterion M holds for $(f_1, f_3), (f_1, f_4), (f_1, f_5)$, and (f_3, f_5), F for (f_2, f_5) and D for (f_4, f_5). The example indicates the way, in which Groebner bases can be computed. The idea for this computation is due to BUCHBERGER [4]. Using the criteria and using, that a polynomial f_k is redundant in the final Groebner basis G, if an $f_i \in G$ exists such that $lt(f_i)$ divides $lt(f_k)$ and using that in this case every (f_k, f_j) for $j > k$ satisfies criterion M or a slightly modified criterion F allowing the cancellation of all such (f_k, f_j), provided the input polynomials are ordered by $lt(f_1) \geq lt(f_2) \geq \ldots$, the algorithm for computing Groebner bases can be formulated in the following way.

Buchberger Algorithm.
Input: nonzero polynomials $f_1, \ldots, f_r$ ordered ,
such that $lt(f_1) \geq \ldots \geq lt(f_r)$.
Output: A Groebner G basis for $I(f_1, \cdots, f_r)$.
Initialization:
$G := \emptyset; P := \emptyset;$

For $s = 1$ to r do
$P1$:= usecritMFD($\{(f, f_s) \mid$ for all $f \in G\}$);
P := join($P1$, usecritB(P, f_s));
G := update(f_s, G);
$s := r$;

Iteration:
Let $(f_i, f_j) \in P$;
$P := P \setminus \{(f_i, f_j)\}$;
Let $S(f_i, f_j) \longrightarrow_G^* h$;

If $h \neq 0$ then
$s := s + 1$;
$f_s := h$;
$P1$:= usecritMFD($\{(f, f_s) \mid$ for all $f \in G\}$);
P := join($P1$, usecritB(P, f_s));
G := update(f_s, G);
If $P \neq \emptyset$ then goto iteration
else return G

The subalgorithm usecritMFD applies the criteria M, F, and D to a set $P1$ of polynomial pairs (f_i, f_s) with fixed s and some $i < s$ by comparing the elements of $P1$ against each other. It returns the subset of $P1$ consisting of all elements for which neither M nor F nor D holds true. usecritB applied to a set of polynomial pairs P and a polynomial f_s cancels all pairs in P, for which criterion B_s holds true. update(f_s, G) cancels every polynomial $g \in G$, for which $lt(f)$ divides $lf(g)$, and returns this reduced set after enlarging it by f_s.

The correctness and the termination of the algorithm in the present form are shown in [11], when the polynomial coefficients are elements of a field like the field of rational numbers. For other coefficient domains see [19].

For simple examples, this algorithm can be applied successfully by hand calculation as our example showed. For more complicated cases the use of a Computer Algebra System is recommended. Apart from the bivariate and the trivariate case, bounds for the complexity of the algorithm are in general unknown. Examples suggest, that these bounds will be huge, if the input polynomials or the ordering of the power products are badly chosen. We developed some special tools for constructing Groebner bases in order to solve also problems, which cannot be solved by other Computer Algebra Systems [18]. For the problem of finding the common zeros of a given set of polynomials, we installed a variant of the algorithm, which will be presented in the following.

4. Solution of Systems of Algebraic Equations

We look for the solutions of a system of algebraic equations

$$f_1(x_1, \ldots, x_n) = 0, \ldots, f_r(x_1, \ldots, x_n) = 0. \tag{4.1}$$

Let us denote the solutions of this system by $Z(f_1, \ldots, f_r)$. Then $Z(f_1, \ldots, f_r)$ is the set of common zeros of $f_1, \ldots, f_r$, or as already mentioned in Section 2, the set of vcommon zeros of any basis of the Ideal I generated by $f_1, \ldots, f_r$. So we transform (4.1) into a Groebner basis with lexicographical ordering $g_1, \ldots, g_n$ and (4.1) is equivalent to $g_1 = 0, \ldots, g_n = 0$. On the other hand a lexicographical Groebner basis has a full or partial triangular dependency pattern, in the best case $n = m$ and

$$\begin{aligned} & g_1(x_1, x_2, \ldots, x_{n-1}, x_n) \\ & g_2(x_1, x_2, \ldots, x_{n-1}) \\ & \vdots \\ & g_{\{m-1\}}(x_1, x_2) \\ & g_m(x_1) \end{aligned}$$

This triagonal pattern can be exploited for the solution of (4.1), e.g. finding the roots of the last equation, substituting it into the others and so on.

But the calculation of the Groebner basis in lexicographical ordering can be extremely hard. So we have modified the Buchberger algorithm for the purpose of equation solving, such that it decomposes the problem wherever possible.

The algorithm (3) started with the set of polynomials $G_n^{(0)}\{f_1, \ldots, f_r\}$ generates in iteration step n a new set $G^{(n)}$. If in one of these $G^{(i)}$ there is a polynomial p with a decomposition

$$\begin{aligned} p(x_1, \ldots, x_n) = 0 \iff & (q_{11}(x_1, \ldots, x_n) = 0 \wedge q_{12}(x_1, \ldots, x_n) = 0 \wedge \ldots) \\ \vee \quad & q_{21}(x_1, \ldots, x_n) = 0 \wedge q_{22}(x_1, \ldots, x_n) = 0 \wedge \ldots) \\ & \vdots \\ \vee \quad & q_{k1}(x_1, \ldots, x_n) = 0 \wedge q_{k2}(x_1, \ldots, x_n) = 0 \wedge \ldots)) \end{aligned}$$

where the q_{ij} are "simpler" than p, then each of the righthand alternatives represents one (possible empty) part of the complete sets of solutions. By substituting them into $G^{(i)}$ one after the other, we get k calculation branches and so a decomposition of the complete calculation. The effect is twofold,

a) the algorithm becomes less complicated: the polynomials are often less in degree;

b) the final result is partitioned too, in the best case into individual solutions.

There are two sources for decomposition: *factorization* and *arithmetic restrictions.*

Factorization. If we find in the ideal I a polynomial h, which can be factorized into

$$h = h_1 * \cdots * h_m,$$

then $Z(f_1, \ldots, f_r)$ splits into the sets $Z(f_1, \ldots, f_r, h_1), \ldots, Z(f_1, \ldots, f_r, h_m)$.

This splitting method can be combined with the algorithm for Groebner basis computation. For the computation of $Z(f_1, \ldots, f_r)$ a Groebner basis of $I := I(f_1, \ldots, f_r)$ can be used. Each polynomial h, obtained by reducing an S-polynomial belongs to I. Therefore if such h is factorizable, $h = h_1 * \cdots * h_m$, then the Groebner basis computation for I can be stopped and the Groebner basis computations with input $f_1, \ldots, f_r, h_k$, $k = 1, \ldots, m$, can be started, because the union of the sets $Z(f_1, \ldots, f_r, h_k), k = 1 \ldots, m$ is the required set of solutions. A consequent application of this factorization gives a tree of Groebner basis computations. Whenever a factorizable polynomial h is found, the computation branches into several Groebner basis computations, and the old Groebner basis computation is discarded. However, the initial Groebner basis calculations for the ideals $I_k := I(f_1, \ldots, f_r, h_k)$, $k = 1, \ldots, m$, are identical with those for the ideal I, until the factored polynomial h appears. Then the algorithms usually differ. This is used in our installation of the procedure. We substitute in the algorithm for computing a Groebner basis of I the factorizable polynomial h by the respective factor h_k and obtain the identical beginnings of the Groebner basis calculations for the ideals I_k. (Some additional calculations may be in parallel too. Therefore before branching the algorithm, we perform also S-polynomial reductions which are known to be identical for all Groebner basis computations with respect to $I_1, \ldots, I_m$.)

For avoiding a multiple computation of the same zeros, we use in our installation a practical criterion. When a Groebner basis for finding $Z(f_1, \ldots, f_r)$ is computed and a factorization of a reduced S-polynomial h, $h = h_1 * \cdots * h_m$, is found, then the Groebner basis calculation for finding $Z(f_1, \ldots, f_r, h_k)$ can be discarded, if a previous $h_i, i < k$, is detected in the ideal generated by $f_1, \ldots, f_r, h_k$, because then the set $Z(f_1, \ldots, f_r, h_k) = Z(f_1, \ldots, f_r, h_k, h_i)$ is contained in $Z(f_1, \ldots, f_r, h_i)$. The same argument holds, if after forthcoming branchings a Groebner basis for a subset of $Z(f_1, \ldots, f_r, h_k)$ is computed and its corresponding ideal contains h_i, because then this subset of $Z(f_1, \ldots, f_r, h_k)$ is contained in $Z(f_1, \ldots, f_r, h_i)$.

The algorithm for computing Groebner bases is fast, when low degree polynomial are found early in the algorithm. This is forced by the method of factorizing. We decrease 'artificially' the degrees of the reduced S-polynomials by substituting the true reduced S-polynomial by each of its factors. The increased number of simpler Groebner basis computations does not destroy this effect. On the contrary, the more factorizations are found, the faster the branching method is, as our examples show.

Only if no or if very few factorizations of polynomials h are found, then by the overhead of testing polynomials for divisibility this branching method is slower than the algorithm presented in section 3.

Arithmetic restrictions. A second source for decomposition is the restriction, that only nonnegative real values make sense as components of solutions. So we can exploit a set of inequalities

$$x_i \geq 0 \quad i = 1, \ldots, N.$$

In generalizing Descartes rule of sign we can state for a multivariate polynomial

$$p = a_0 + a_1 x_1^{i_{11}} \cdots x_n^{i_{1n}} + \cdots + a_m x_1^{i_{m1}} \cdots x_n^{i_{mn}}$$

where all a_i are real and have the same sign:

a) if $a_0 \neq 0$ then $p(x_1, \ldots, x_n) = 0$ implies that at least one $x_i < 0$;

b) for $a_0 = 0$ then $p(x_1, \ldots, x_n) = 0$ and $x_i \geq 0$ implies, that each monomial in p is zero already.

In other words: If $a_0 \neq 0$ there cannot exist a solution which satisfies the above inequalities and the calculation branch can be cancelled. For $a_0 = 0$ p is decomposed into a list of monomials each of which has to be zero. As these monomials are factorizable immediately, we easily get a twofold decomposition. For example, the polynomial

$$17xy^2r \quad + 4x^2z$$

leads to the decomposition

$$x = 0 \ \vee \ (y = 0 \ \wedge \ z = 0) \ \vee \ (r = 0 \ \wedge \ z = 0)$$

which describes the nonnegative real solutions.

Experience has shown, that in the chemical application area the above criterion plays an important role as source of decomposition and as test for the nonexistence of a relevant solution. In the last case a calculation branch can be cancelled immediately.

The arithmetic decomposition uses the same hook in the algorithm as the factorization: when it hits, it generates sets of zero polynomials each of which is root of a separate calculation tree.

If the coefficients contain formal parameters, the same technique can be applied for determining their sign.

5. Computations for Chemical Reaction Systems

5.1 Polynomial equations

Chemical reaction systems for isothermal reactors are described by reaction networks or by systems of chemical reaction equations:

$$k_{i\to j} : y_{in}c_n + \ldots + y_{im}c_m \Longrightarrow y_{jk}c_k + \ldots + y_{jl}c_l \tag{5.1}$$

where c_s denotes the overall molecular concentration of the species with number s, y_{sj} is the stoichiometric coefficient of species s in the complex j and $k_{i\to j}$ is the rate constant for the reaction from complex i to j. For the numerical simulation of their kinetics they are transformed into systems of ordinary differential equations (ODEs), which in the case of mass action have the general form

$$\dot{c_s} = \sum_{i\to j} k_{i\to j} \prod_{l=1,n} c_l^{y_{il}}(y_{sj} - y_{si}) \quad , \quad s = 1, \ldots, n \ . \tag{5.2}$$

The transformation from (5.1) (or other equivalent input forms) to (5.2) can be done automatically with REDUCE: the right-hand sides of the ODEs are initialized with zeros and the contributions of each reaction read into the species are added into the corresponding ODE:

LET $p = k_{i\to j} * c_n * *y_{in} * \ldots * c_{mi} * *y_{im}$; (5.3)

FOR EACH c_p in left side of (5.1) subtract $(k_i * y_{ip})$ from ODE (c_p) ; (5.4)

FOR EACH c_p in right side of (5.1) add $(k_i * y_{ip})$ to ODE (c_p) ; (5.5)

If we now are interested in the steady states of sytem (5.2), we have to set $\dot{c_s}$ to zero. (5.2) is transformed into a system of homogeneous algebraic equations

$$0 = \sum_{i\to j} k_{i\to j} \prod_{l=1,n} c_l^{y_{il}}(y_{sj} - y_{si}) \quad , \quad s = 1, \ldots, n \tag{5.6}$$

which can be investigated with algebraic methods. The "variables" in (5.6) are the c_i. The y_i are small nonnegative integer constants (typically 0,1 or 2). The $k_{i\to j}$ may be numerical constants (nonnegative numbers, typically floating point fractions) or they may be formal parameters (all or some of them). In all cases we are interested to solve (5.6) or to look for criteria for the solvability.

5.2 Computations with algebraic coefficients

Example EDELSTEIN: As a first detailed example we look at the Edelstein system with the reaction graph:

$$A_1 \underset{\beta}{\overset{\alpha}{\rightleftarrows}} 2A_1$$

$$A_1 + A_2 \underset{\epsilon}{\overset{\gamma}{\rightleftarrows}} A_3 \underset{\eta}{\overset{\theta}{\rightleftarrows}} A_2$$

The corresponding algebraic equations are given in (1.5). For the Groebner basis calculation we use the variable sequence $\{c_3,\ c_2,\ c_1\}$, which is the optimal one because it represents "growing" degrees in the equations [3]. Rewritten in this variable order the system is

$$\begin{aligned} p_1 &= -c_3\epsilon - c_3\theta + c_2c_1\gamma + c_2\eta \\ p_2 &= c_3\epsilon + c_3\theta - c_2c_1\gamma - c_2\eta \\ p_3 &= c_3\epsilon - c_2c_1\gamma - c_1\beta + c_1\alpha \end{aligned}$$

Here the Groebner basis computation is extremely simple: p_2 is redundant and can be omitted and the only h-polynomial is the sum of p_1 and p_3, eliminating the c_3 term. The resulting Groebner basis is

$$\begin{aligned} b_1 &= -c_3\theta + c_2\eta - c_1\beta + c_1\alpha \\ b_2 &= -c_2c_1\gamma\theta + c_2\epsilon\eta - c_1^2\beta\epsilon \\ &\quad -c_1^2\beta\theta + c_1\alpha\epsilon + c_1\alpha\theta \end{aligned}$$

This basis has an incomplete triangular dependency: $\{b_1\,(c_3,\ c_2,\ c_1);\ b_2\,(c_2,\ c_1)\}$. We now can determine the complete solution by solving b_2 either *wrt* c_1 (using c_2 as formal parameter) or *wrt* c_2 (c_1 as parameter), substitute the result into $b1$ and solve b_1 *wrt* c_3. If we select c_1 as parameter, the final result is

$$\begin{aligned} c_3 &= \frac{c_1(-c_1^2\beta\gamma + c_1\alpha\gamma - c_1\beta\eta + \alpha\eta)}{c_1\gamma\theta - \epsilon\eta} \\ c_2 &= \frac{c_1(-c_1\beta\epsilon - c_1\beta\theta + \alpha\epsilon + \alpha\theta)}{c_1\gamma\theta - \epsilon\eta} \end{aligned}$$

representing c_2 and c_3 as formal functions of c_1.

For the interpretation of this result in the style of the deficiency theorems, we have to do further algebraic investigation. The question here is "are there multiple steady states within one stoichiometric compatibility class"? Two compositions c and

c' are members of one stoichiometric compatibility class, if their difference is a linear combination of reaction vectors. The reaction vectors for the Edelstein network are

$$\begin{array}{ll} e_1 - 2e_1 & 2e_1 - e_1 \\ e_1 + e_2 - e_3 & e_3 - e_1 - e_2 \\ e_3 - e_2 & e_2 - e_3 \end{array}$$

These vectors generate a 2 dimensional subspace, e.g. with the basis $\{e_1,\ e_3 - e_2\}$. We now want to evaluate, which steady states are in the same stoichiometric compatibility class as a given concentration vector $\{c_1',\ c_2',\ c_3'\}$. The corresponding equation system is

$$0 = -\begin{bmatrix} c_1 \\ c_2(c_1) \\ c_3(c_1) \end{bmatrix} - \begin{bmatrix} c_1' \\ c_2' \\ c_3' \end{bmatrix} + l_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + l_2 \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

where c_1, l_1 and l_2 are variables, while the c_1', c_2', c_3' and the rates are formal parameters (constants). The linear system is solvable if and only if the determinant of the matrix

$$\begin{bmatrix} c_1 - c_1' & , & 1 & , & 0 \\ c_2(c_1) - c_2' & , & 0 & , & -1 \\ c_3(c_1) - c_3' & , & 0 & , & 1 \end{bmatrix}$$

vanishes. This leads to the polynomial

$$-\alpha\gamma c_1^2 - \alpha\epsilon c_1 - \alpha\eta c_1 - \alpha\,\theta\,c_1 + \beta\,\gamma\,c_1^3 + \beta\,\epsilon\,c_1^2$$

$$+\beta\,\eta\,c_1^2 + \beta\,\theta\,c_1^2 + \gamma\,\theta\,c_1\,c_2' + \gamma\,\theta\,c_1\,c_3' - \epsilon\,\eta\,c_2' - \epsilon\,\eta\,c_3'$$

which does not depend from c_1' because of the degenerate structure of the matrix. This cubic polynomial can be solved with respect to c_1 by the Cardano formula in full generality with α, β, γ, η, θ, c_1, c_2' and c_3' as formal parameters: c_1 is expressed as function of the rate constants and the input concentration vector; this computation can be done with REDUCE. We receive three formal solutions, which, however, are bit lengthy in their full generality (several hundreds of lines of REDUCE output). The solutions contain formal imaginary parts and radicals with formally undetermined sign; so we are not able to determine the number of positive real solutions in full generality. By assigning numeric values to the rate constants and to the c_i however, they can be evaluated.

Example MODIFIED EDELSTEIN [9]: In the previous example, two of the equations are linearly dependent and in fact the algorithm eliminates the redundant equation

immediately. We now investigate a modification of the Edelstein system, where this dependence is no longer present:

$$A_1 \underset{\beta}{\overset{\alpha}{\rightleftharpoons}} 2A_1$$

$$A_1 + A_2 \underset{\epsilon}{\overset{\gamma}{\rightleftharpoons}} A_3 \underset{\eta}{\overset{\theta}{\rightleftharpoons}} 2A_2$$

The corresponding algebraic equations are:

$$\begin{aligned} 0 &= -c_1^2\,\beta - c_1\,c_2\,\gamma + c_1\,\alpha + c_3\,\epsilon \\ 0 &= -c_1\,c_2\,\gamma - 2\,c_2^2\,\eta + c_3\,\epsilon + 2\,c_3\,\theta \\ 0 &= c_1\,c_2\,\gamma + c_2^2\,\eta - c_3\,\epsilon - c_3\,\theta \end{aligned}$$

Here the algorithm soon detects a H-polynomial

$$-c_1^2\,\beta + c_1\,\alpha$$

which reflects a typical behaviour of the whole application class: it can be factorized into a product *variable* $*$ *polynomial*, here

$$c_1\,(-c_1\,\beta + \alpha)$$

The algorithm detects more factorizations of this type and generates separate calculation branches, which result in three Groebner basis:

$$\{c_3, c_2, c_1\}\ ,$$

$$\{c_3, c_2, c_1\beta - \alpha\}\ ,$$

$$\{-c_3\beta^2\epsilon^2\eta + \alpha^2\gamma^2\theta, c_2\beta\epsilon\eta - \alpha\gamma\theta, c_1\beta - \alpha\}$$

Here the first basis represents the trivial solution $c_3 = c_2 = c_1 = 0$; the second basis has one nonzero value $c_1 = \alpha/\beta$. The third basis offers the unique positive solution

$$c_3 = \frac{\alpha^2\gamma^2\theta}{\beta^2\epsilon^2\eta}\ ,\quad c_2 = \frac{\alpha\gamma\theta}{\beta\epsilon\eta}\ ,\quad c_1 = \frac{\alpha}{\beta}$$

Here no futher processing is necessary. If we execute the algorithm with the restriction to positive real solutions, only the third basis and the last result is presented.

Example HEINMETS [2]:

$$\{\ -x_1\,p_2 - x_1\,p_3 + x_3\,p_1\ ,$$

$$-x_2\,p_4 + x_3\,p_{15}\ ,\ x_1\,p_2 + x_1\,p_3 + x_2\,p_4 - x_3\,p_1 - x_3\,p_{15}\ ,$$

$$-x_4\,p_7 + x_6\,x_7\,p_{12}\ ,$$

$$-x_5\,p_8 + x_7\,p_6\ ,\ x_1\,p_3 - x_6\,x_7\,p_{12} - x_6\,p_5\ ,$$

$$x_4\,p_7 + x_5\,p_8 - x_6\,x_7\,p_{12} - x_7\,p_6\ ,$$

$$-x_8\,x_9\,p_{13} + x_{12}\,p_{14}\ ,\ x_5\,p_8 - x_8\,x_9\,p_{13} - x_9\,p_9\ ,\ x_{11}\,p_{11}\ ,$$

$$-x_{11}\,p_{11} + x_{12}\,p_{10}\ ,\ x_8\,x_9\,p_{13} + x_{11}\,p_{11} - x_{12}\,p_{10} - x_{12}\,p_{14}\ \}$$

Although still moderate in size, this system demonstrates a behaviour which we find as well with the "big" systems derived from reaction systems. The system cannot have a positive solution: e.g. the 10th equation forces x_{11} to zero already in the original system. During the calculation of reduction steps the polynomial $x_8\,x_7$ is found, which is of purely monomial type and so is easy factorizable; it causes the algorithm to branch. So the final set for Groebner bases is

$$\{\ x_2\,p_1\,p_3\,p_4 - x_6\,p_2\,p_5\,p_{15} - x_6\,p_3\,p_5\,p_{15}\ ,\ x_4\ ,\ -x_1\ p_3 + x_6\,p_5\ ,$$

$$-x_3\,p_1\,p_3 + x_6\,p_2\,p_5 + x_6\,p_3\,p_5\ ,\ x_5\ ,\ x_{11}\ ,\ x_{12}\ ,\ x_9\ ,\ x_7\}\ ,$$

$$\{\ -x_2\,p_1\,p_3\,p_4 + x_6\,x_7\,p_2\,p_{12}\,p_{15} + x_6\ x_7\,p_3\,p_{12}\,p_{15}$$

$$+x_6\,p_2\,p_5\,p_{15} + x_6\,p_3\,p_5\,p_{15}\ ,\ x_4\,p_7 - x_6\,x_7\,p_{12}\ ,$$

$$x_1\,p_3 - x_6\,x_7\,p_{12} - x_6\,p_5\ ,$$

$$x_3\,p_1\,p_3 - x_6\,x_7\,p_2\,p_{12} - x_6\,x_7\,p_3\,p_{12} - x_6\,p_2\,p_5$$

$$-x_6\,p_3\,p_5\ ,\ x_5\,p_8 - x^7\,p_6\ ,\ x_8\ ,\ x_{11}\ ,\ x_{12}\ ,\ x_7\,p_6 - x_9\,p_9\}$$

and from that we can calculate (by an automatic postprocessor) the sets of solutions:

$$
\begin{aligned}
\{\{x_2 &= \frac{x_6\, p_2\, p_5\, p_{15} + x_6\, p_3\, p_5\, p_{15}}{p_1\, p_3\, p_4}\,, \\
x_4 &= 0\,, \\
x_1 &= \frac{x_6\, p_5}{p_3}\,, \\
x_3 &= \frac{x_6\, p_2\, p_5 + x_6\, p_3\, p_5}{p_1\, p_3}\,, \\
x_4 &= 0,\ x_5 = 0,\ x_{11} = 0,\ x_{12} = 0,\ x_9 = 0,\ x_7 = 0\}\,, \\
\{x_2 &= \frac{x_6\, x_7\, p_2\, p_{12}\, p_{15} + x_6\, x_7\, p_3\, p_{12}\, p_{15} + x_6\, p_2\, p_5\, p_{15} + x_6\, p_3\, p_5\, p_{15}}{p_1\, p_3\, p_4}\,, \\
x_4 &= \frac{x_6\, x_7\, p_{12}}{p_7}\,, \\
x_1 &= \frac{x_6\, x_7\, p_{12} + x_6\, p_5}{p_3}\,, \\
x_3 &= \frac{x_6\, x_7\, p_2\, p_{12} + x_6\, x_7\, p_3\, p_{12} + x_6\, p_2\, p_5 + x_6\, p_3\, p_5}{p_1\, p_3}\,, \\
x_5 &= \frac{x_7\, p_6}{p_8}\,, \\
& x_8 = 0,\ x_{11} = 0,\ x_{12} = 0\,, \\
x_9 &= \frac{x_7\, p_6}{p_9}\}\}
\end{aligned}
$$

Note that some of the variables are not mentioned as left-hand sides of the solutions at all. So the value of x_6 is arbitrary for both solutions and x_7 is arbitrary in the second solution. This is typical for larger systems.

Example SCHLOSSER [20]**:** Of course, computations with formal coefficients are not applicable in the general case for two reasons:

a) during the internal computations the Buchberger algorithm has to combine coefficients by the basic arithmetic operations +,-,*,/ very often, which produces a growth of coefficients in an exploding manner so that soon all resources of the machine are exhausted,

b) the algebraic dependence in the case of many parameters is often so complicated, that general solutions, even if computable, are difficult to be interpreted.

As an example for the second effect we present the following system of algebraic equations derived from the ODEs for a continuous flow stirred reactor [20]:

$$\begin{aligned} 0 &= -\ k1\ ca\ cb + 2k3\ cc + (caf - ca)/\theta; \\ 0 &= -\ k1\ ca\ cb - k2\ cb\ cc + (cbf - cb)/\theta; \\ 0 &= -\ k2\ cb\ cc - k3\ cc + (ccf - cc)/\theta; \\ 0 &= \quad k1\ ca\ cb + (cdf - cd)/\theta; \\ 0 &= \quad k2\ cb\ cc + (cef - ce)/\theta; \end{aligned} \tag{5.7}$$

The Buchberger algorithm almost immediately offers the base (LEX over $\mathbf{Q}[\ k1,\ k2,\ k3,\ caf,\ cbf,\ ccf,\ cdf,\ cef,\ \theta\ ]$). For technical reasons we here present only the last member of the base, which is a univariate polynomial of degree 3 in cb:

$$\begin{aligned} &(-k1\ \theta^2\ k2)\ cb^3 + (-k1\ k3\ \theta^2 - k1\ caf\ \theta^2 k2 \\ &+k1\ \theta^2\ k2\ cbf - k1\ \theta^2\ k2\ ccf - k1\ \theta - \theta\ k2)\ cb^2 \\ &+(-k1\ k3\ caf\ \theta^2 + k1\ k3\ \theta^2 cbf - 2k1\ k3\ \theta^2\ ccf - k1\ caf\ \theta \\ &+k1\ \theta\ cbf - k3\ \theta + \theta\ k2\ cbf - \theta\ k2\ ccf - 1\ )\ cb + (\ k3\ \theta\ cbf + cbf\ ) \end{aligned} \tag{5.8}$$

This polynomial can be solved with respect to the variable cb by the Cardano formulas, but this computation is time consuming and the interpretation of the result will be difficult. In this case and even in more complicated cases, it makes sense to substitute some of the parameters with reasonable numerical values and to calulate the base with very few formal parameters. For example, if we subsitute in (5.8) the values proposed by SCHLOSSER [20]

$k1 := 0.5;\ k2 := 0.9;\ k3 := 0.8;\ caf := 16;\ cbf := 1;\ ccf := 4;\ cdf := 80;\ cef := 36\ ;$

leaving θ as formal parameter, we compute instead of (5.8) the much better readable polynomial

$$(9\theta^2)\ cb^3 + (179\ \theta^2 + 28\ \theta)^2\ cb + (184\ \theta^2 + 220\ \theta + 20)\ cb + (-16\ \theta - 20)\ . \tag{5.9}$$

5.3 Computations with numerical coefficients

The following examples have their origin in "real" chemical reaction systems. The problems are formulated as chemical reaction equations including numerical values for the rate constants. The Buchberger algorithm is used with factorization and nonnegative restriction. The systems have low degrees and a sparse pattern. As a consequence of this specific structure even a Groebner basis with graduated order leads to the set of solutions; its main task is the decomposition. The solutions that contain a more or less big set of variables with zero values, and often values of other variables are arbitrary.

Example PYRIDINE [13]: This example is the smallest member of the series. Reaction system:

a	$\Rightarrow$	b	(1.8)	b	$\Rightarrow$	c	(0.9)
b + c	$\Rightarrow$	d + f	(29.)	c + c	$\Rightarrow$	e + f	(9.)
d	$\Rightarrow$	e	(0.053)	c	$\Rightarrow$	f + g	(2.)
d	$\Rightarrow$	b + g	(0.06)	e	$\Rightarrow$	c + g	(5.)
b	$\Rightarrow$	a	(0.02)	d + f	$\Rightarrow$	b + c	(0.5)
e + f	$\Rightarrow$	c + c	(2.)				

Algebraic equation system:

$$
\begin{aligned}
0 &= -1.8\, ca + 0.02 cb, \\
0 &= 1.8\, ca - 29.0\, cb\, cc - 0.92\, cb + 0.5\, cd\, cg + 0.06\, cd, \\
0 &= -29.0\, cb\, cc + 0.9\, cb - 18.0\, cc\, 2 \\
&= -2.0\, cc + 0.5\, cd\, cf + 4.0\, ce\, cf + 5.0\, ce, \\
0 &= 29.0\, cb\, cc - 0.5\, cd\, cf - 0.113\, cd, \\
0 &= 9.0\, cc\ 2 + 0.053\, cd - 2.0\, ce\, cf - 5.0\, ce, \\
0 &= 29.0\, cb\, cc + 9.0\, cc\ 2 + 2.0\, cc - 0.5\, cd\, cf - 2.0\, ce\, cf, \\
0 &= 2.0\, cc + 0.06\, cd + 5.0\, ce
\end{aligned}
$$

A Groebner basis is $\{ca,\ cb,\ cc,\ cd,\ ce\}$. So all solutions of the system have the form

$$ca = 0\ ,\ cb = 0\ ,\ cc = 0\ ,\ cd = 0\ ,\ ce = 0$$

with cf, cg arbitrary (free variable).

Example SMOG [8]: The reaction system SMOG is characterized by a high deficiency (=31). The set of algebraic equations has 50 polynomials in 37 variables (some of the species appear on the right -hand side of the reaction arrow only). The longest polynomial has 31 terms; there are some polynomials having only one term.

```
6.0*c2*c25 + 6.0*c2*c27 + 6.0*c2*c30 + 4.0*c2*c33 + 6.0*c2*c35 + 6.0*c2*c38 + 8.0*c2*c43
+ 0.32*c2*c45 + 0.2*c2*c46 + 0.2*c2*c47 + 280000.0*c2*c6 + 3.0*c2*c20 + 3.0*c2*c23 + 0.1*c2*c44
+ 22.5*c2*c5 - 0.001*c2*c1 + 30.0*c2*c13 - 1470.0*c10*c1 + 2.76*c7 - 0.001*c25*c1 - 0.001*c27*c1
- 2.0*c43*c1 - 858.0*c6*c1 - 0.1*c20*c1 - 0.1*c23*c1 - 0.049*c5*c1 - 0.001*c1*c13 - 16.2*c1*c24
- 16.2*c1*c28 - 5150.0*c1*c3 - 0.37*c1,

- 10000.0*c2*c10 - 6.0*c2*c25 - 6.0*c2*c27 - 6.0*c2*c30 - 4.0*c2*c33 - 6.0*c2*c35 - 6.0*c2*c38
- 8.0*c2*c43 -0.16*c2*c45 - 0.1*c2*c46 - 0.1*c2*c47 - 140000.0*c2*c6 - 3.0*c2*c20 - 3.0*c2*c23
- 0.1*c2*c44 - 22.5*c2*c5 - 0.001*c2*c1 - 30.0*c2*c13 + 0.01*c9 + 0.49*c6*c1 + 5150.0*c1*c3
+ 0.37*c1,
```

- 220.0*c21*c3 - 294.0*c22*c3 - 0.00125*c15*c4*c3 - 3425.0*c15*c3 - 5150.0*c1*c3 + 0.37*c1 - 22.0*c4*c3,

22.5*c2*c5 - 0.00098*c10*c11*c4 - 0.125*c10*c15*c4 - 15.0*c16*c4 - 15.0*c18*c4 - 15.0*c19*c4 - 0.015*c32*c4- 0.005*c39*c4 + 0.49*c6*c1 - 2.6*c17*c4 + 2.45e-05*c21*c5 + 0.0005*c22*c5 + 0.049*c5*c1 - 0.00125*c15*c4*c3 + 0.001*c1*c13 + 5150.0*c1*c3 - 0.26*c4*c24 - 0.26*c4*c28 - 0.015*c4*c37 -22.0*c4*c3 + 70.0*$c13^2$,

- 22.5*c2*c5 - 2.45e-05*c21*c5 - 0.0005*c22*c5 - 0.014*c5*c15 - 0.049*c5*c1 + 22.0*c4*c3,

- 140000.0*c2*c6 + 2.76*c7 + 0.01*c45 + 0.05*c46 + 0.05*c47 - 858.49*c6*c1 + 0.1*c20*c1 + 0.1*c23*c1 + 0.049*c5*c1,

- 3.51*c7 + 858.0*c6*c1,

1470.0*c10*c1 + 1.5*c7,

10000.0*c2*c10 + 0.002*c2*c1 - 0.01*c9 + 0.001*c1*c13,

- 10000.0*c2*c10 + 30.0*c2*c13 - 0.00098*c10*c11*c4 - 36000.0*c10*c17 - 25000.0*c10*c21 - 25000.0*c10*c22 - 0.125*c10*c15*c4 - 5000.0*c10*c15 - 1470.0*c10*c1 + 0.01*c9 + 3000.0*c34 + 2000.0*c39 + 0.3*c23 + 2.45e-05*c21*c5 + 220.0*c21*c3 + 0.0005*c22*c5 + 294.0*c22*c3,

- 0.00098*c10*c11*c4 + 36000.0*c10*c17 + 2.6*c17*c4,

0.00098*c10*c11*c4 + 0.0001*c44,

- 30.0*c2*c13 + 0.00098*c10*c11*c4 + 0.001*c30 + 0.001*c33 + 0.001*c35 + 0.001*c38 + 0.005*c39*c4 + 2.6*c17*c4 - 0.001*c1*c13 + 0.26*c4*c24 + 0.26*c4*c28 - 140.0*$c13^2$,

70.0*$c13^2$,

- 0.125*c10*c15*c4 - 5000.0*c10*c15 - 0.014*c5*c15 - 0.00125*c15*c4*c3 - 3425.0*c15*c3,

- 15.0*c16*c4 + 2925.0*c15*c3,

- 36000.0*c10*c17 + 25000.0*c10*c21 - 2.6*c17*c4 + 0.3*c20 + 0.3*c23 + 2.45e-05*c21*c5 + 220.0*c21*c3 + 3.3e-05*c21 + 0.0036*c22 + 2925.0*c15*c3,

- 15.0*c18*c4 + 0.0001*c44 + 0.0036*c22 + 500.0*c15*c3,

0.1*c2*c44 + 25000.0*c10*c22 - 15.0*c19*c4 + 0.0005*c22* c5 + 294.0*c22*c3 + 500.0*c15*c3,

- 3.0*c2*c20 - 0.1*c20*c1 - 0.3*c20 + 0.007*c5*c15 + 0.00125*c15*c4*c3,

3.0*c2*c23 - 25000.0*c10*c21 + 3000.0*c34 + 0.001*c33 + 0.1* c23*c1 + 360000.0*c44*c24 - 2.45e-05*c21*c5 - 220.0*c21*c3 - 3.3e-05*c21 + 0.007*c5*c15 + 0.00125*c15*c4*c3 + 0.26*c4*c24 + 0.015*c4*c37 + 360000.0*$c24^2$,

3.0*c2*c20 - 25000.0*c10*c22 + 0.015*c32*c4 + 0.001*c38 + 2000.0*c39 + 0.1*c20*c1 + 360000.0*c44*c28 - 0.0005*c22*c5 - 294.0*c22*c3 - 0.0036*c22 + 0.007*c5*c15 + 0.26*c4*c28 + 360000.0*$c28^2$,

- 3.0*c2*c23 - 0.1*c23*c1 - 0.3*c23 + 0.007*c5*c15,

6.0*c2*c25 + 0.1*c2*c46 + 0.05*c46 + 0.3*c20 - 360000.0*c44 *c24 - 16.2*c1*c24 - 0.26*c4*c24 - 720000.0*$c24^2$,

- 6.0*c2*c25 + 15.0*c18*c4 - 0.001*c25*c1,

16.2*c1*c24,

- 6.0*c2*c27 + 15.0*c16*c4 - 0.001*c27*c1,

6.0*c2*c27 + 0.1*c2*c47 + 0.05*c47 - 360000.0*c44*c28 - 16.2*c1*c28 - 0.26*c4*c28 - 720000.0*$c28^2$,

16.2*c1*c28,

- 6.0*c2*c30 + 0.075*c10*c15*c4 - 0.001*c30,

0.001*c30,

6.0*c2*c30 - 0.015*c32*c4,

- 4.0*c2*c33 + 0.015*c32*c4 - 0.001*c33,

4.0*c2*c33 - 3000.0*c34,

- 6.0*c2*c35 + 0.05*c10*c15*c4 - 0.001*c35,

0.001*c35,

6.0*c2*c35 - 0.015*c4*c37,

- 6.0*c2*c38 - 0.001*c38 + 0.015*c4*c37,

6.0*c2*c38 - 0.005*c39*c4 - 2000.0*c39,

0.005*c39*c4 + 360000.0*c44*c24 + 360000.0*c44*c28,

5000.0*c10*c15,

36000.0*c10*c17 + 25000.0*c10*c21 + 25000.0*c10*c22 + 5000.0*c10*c15,

- 8.0*c2*c43 + 15.0*c19*c4 - 2.0*c43*c1,

8.0*c2*c43 + 0.16*c2*c45 - 0.1*c2*c44 + 0.01*c45 - 360000.0*c44*c24 - 360000.0*c44*c28 - 0.0001*c44,

- 0.16*c2*c45 + 2.0*c43*c1 - 0.01*c45,

- 0.1*c2*c46 + 0.001*c25*c1 - 0.05*c46,

- 0.1*c2*c47 + 0.001*c27*c1 - 0.05*c47,

3.3e-05*c21,

$360000.0*c24^2$,

$360000.0*c28^2$

The decomposition leads to 330 partial result bases which are shipped here. A postprocessing analysis detects, that most of these solutions are redundant in the sense, that they describe a subspace of other solutions. The final number of completely independent solution spaces is 11; most of them force some variables to be zero and leave other variables completely free. There is an important intersection between all solutions.

S O L U T I O N S :

---

common part of all solutions:
c28 = 0 , c24 = 0 , c13 = 0 , c1 = 0 , c22 = 0 , c21 = 0 , c44 = 0 ,
c23 = 0 , c20 = 0 , c47 = 0 , c46 = 0 , c45 = 0 , c39 = 0 , c38 = 0 ,
c35 = 0 , c33 = 0 , c30 = 0 , c7 = 0 , c34 = 0

---

solution 1: free variables: c6 c11 c3 c5 c18 c16 c32 c37 c10 c19 c43 c25 c27
c2 = 0 , c4 = 0 , c15 = 0 , c17 = 0 , c9 = 0

---

solution 2: free variables: c6 c11 c5 c4 c43 c25 c27
c2 = 0 , c10 = 0 , c15 = 0 , c3 = 0 , c17 = 0 , c37 = 0 , c32 = 0 ,
c19 = 0 , c18 = 0 , c16 = 0 , c9 = 0

---

solution 3: free variables: c6 c11 c3 c5 c18 c16 c32 c37 c17 c19 c43 c25 c27
c2 = 0 , c4 = 0 , c10 = 0 , c15 = 0 , c9 = 0

---

solution 4: free variables: c6 c5 c10 c4 c43 c25 c27
c2 = 0 , c15 = 0 , c3 = 0 , c17 = 0 , c11 = 0 , c37 = 0 , c32 = 0 ,
c19 = 0 , c18 = 0 , c16 = 0 , c9 = 0

---

solution 5: free variables: c6 c11 c18 c16 c32 c37 c15 c17 c19 c43 c25 c27
c2 = 0 , c4 = 0 , c10 = 0 , c3 = 0 , c5 = 0 , c9 = 0

---

solution 6: free variables: c6 c11 c15 c4 c43 c25 c27
c2 = 0 , c10 = 0 , c3 = 0 , c5 = 0 , c17 = 0 , c37 = 0 , c32 = 0 ,
c19 = 0 , c18 = 0 , c16 = 0 , c9 = 0

---

solution 7: free variables: c11 c3 c18 c16 c32 c37 c19
c4 = 0 , c15 = 0 , c5 = 0 , c17 = 0 , c6 = 0 , c43 = 0 , c27 = 0 ,
c25 = 0 ,
c2*c10 - 1.0e-06*c9 = 0

---

solution 8: free variables: c11 c3 c18 c16 c32 c37 c17 c19 c2
c4 = 0 , c10 = 0 , c15 = 0 , c5 = 0 , c6 = 0 , c43 = 0 , c27 = 0 ,
c25 = 0 , c9 = 0

---

solution 9: free variables: c11 c18 c16 c32 c37 c15 c17 c19 c2
c4 = 0 , c10 = 0 , c3 = 0 , c5 = 0 , c6 = 0 , c43 = 0 , c27 = 0 ,
c25 = 0 , c9 = 0

---

solution 10: free variables: c11 c15 c4 c2
c10 = 0 , c3 = 0 , c5 = 0 , c17 = 0 , c6 = 0 , c43 = 0 , c27 = 0 ,
c25 = 0 , c37 = 0 , c32 = 0 , c19 = 0 , c18 = 0 , c16 = 0 , c9 = 0

---

solution 11: free variables: c4
c15 = 0 , c3 = 0 , c5 = 0 , c17 = 0 , c6 = 0 , c43 = 0 , c27 = 0 ,
c25 = 0 , c11 = 0 , c37 = 0 , c32 = 0 , c19 = 0 , c18 = 0 , c16 = 0 ,
c2*c10 - 1.0e-06*c9 = 0

6. Examples from Related Application Areas

Example CYCLOHEXANE [6]: The molecular geometry of a cyclic carbon hydrogen molecule with six nodes is described by four equations; these are calculated from two determinants

$$f_1 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 8/3 & y_1 & 8/3 \\ 1 & 1 & 0 & 1 & 8/3 & y_2 \\ 1 & 8/3 & 1 & 0 & 1 & 8/3 \\ 1 & y_1 & 8/3 & 1 & 0 & 1 \\ 1 & 8/3 & y_2 & 8/3 & 1 & 0 \end{vmatrix} \quad f_4 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 8/3 & y_1 & 8/3 & 1 \\ 1 & 1 & 0 & 1 & 8/3 & y_2 & 8/3 \\ 1 & 8/3 & 1 & 0 & 1 & 8/3 & y_3 \\ 1 & y_1 & 8/3 & 1 & 0 & 1 & 8/3 \\ 1 & 8/3 & y_2 & 8/3 & 1 & 0 & 1 \\ 1 & 1 & 8/3 & y_3 & 8/3 & 1 & 0 \end{vmatrix}$$

The first equation is $0 = f_1$, the equations two and three are calculated from f_1 by interchanging y_1, y_2 and y_3 in a cyclic manner and the fourth equation is $0 = f_4$. The complete system is

$$\begin{aligned} 0 &= (-81\,y_1^2\,y_2^2 + 594\,y_1^2\,y_2 - 225\,y_1^2 + 594\,y_1\,y_2^2 - 3492\,y_1\,y_2 \\ &\quad -750\,y_1 - 225\,y_2^2 - 750\,y_2 + 14575)/81 \\ 0 &= (-81\,y_2^2\,y_3^2 + 594\,y_2^2\,y_3 - 225\,y_2^2 + 594\,y_2\,y_3^2 - 3492\,y_2\,y_3 \\ &\quad -750\,y_2 - 225\,y_3^2 - 750\,y_3 + 14575)/81 \\ 0 &= (-81\,y_1^2\,y_3^2 + 594\,y_1^2\,y_3 - 225\,y_1^2 + 594\,y_1\,y_3^2 - 3492\,y_1\,y_3 \\ &\quad -750\,y_1 - 225\,y_3^2 - 750\,y_3 + 14575)/81 \\ 0 &= (162\,y_1^2\,y_2^2\,y_3 + 162\,y_1^2\,y_2\,y_3^2 - 1188\,y_1^2\,y_2\,y_3 - 450\,y_1^2\,y_2 \\ &\quad -450\,y_1^2\,y_3 + 3300\,y_1^2 + 162\,y_1\,y_2^2\,y_3^2 - 1188\,y_1\,y_2^2\,y_3 \\ &\quad -450\,y_1\,y_2^2 - 1188\,y_1\,y_2\,y_3^2 + 5184\,y_1\,y_2\,y_3 + 5100\,y_1\,y_2 \\ &\quad -450\,y_1\,y_3^2 + 5100\,y_1\,y_3 - 7150\,y_1 - 450\,y_2^2\,y_3 + 3300\,y_2^2 \\ &\quad -450\,y_2\,y_3^2 + 5100\,y_2\,y_3 - 7150\,y_2 + 3300\,y_3^2 - 7150\,y_3 - 60500)/81 \end{aligned}$$

Here the Algorithm detects nonhomogeneous factorizations and splits the problem into seven separate Groebner basis:

$$\{\{3\,y_1-11,9\,y_2-25,3\,y_3-11\},\{9\,y_1-25,3\,y_2-11,3\,y_3-11\}\,,\{3\,y_1-11,3\,y_2-11,3\,y_3-11\}\,,$$

$$\{3\,y_1\,y_2+3\,y_1\,y_3-22\,y_1+3\,y_2\,y_3-22\,y_2-22\,y_3+121\,,$$

$$27\,y_1\,y_3^2-198\,y_1\,y_3+75\,y_1+27\,y_2\,y_3^2-198\,y_2\,y_3+75\,y_2-198\,y_3^2+1164\,y_3+250\,,$$

$$81\,y_2^2\,y_3^2-594\,y_2^2\,y_3+225\,y_2^2-594\,y_2\,y_3^2+3492\,y_2\,y_3+750\,y_2+225\,y_3+750\,y_3-14575\}\,,$$

$$\{3\,y_1+5,3\,y_2+5,3\,y_3+5\},\{3\,y_1-19,3\,y_2+5,3\,y_3+5\}\,,$$

$$\{3\,y_1+5,3\,y_2-19,3\,y_3+5\}\}$$

Some of these (e.g. the first three) represent zero dimensional ideals, here with one unique solution each, others represent parameterized solutions. The variables y_i here represent distances between molecules and so only nonnegative real values are interesting. So if we invoke the Buchberger with the additional restriction to nonnegativity, it detects the polynomial $3\,y_3+5$ and cancels all calculation branches with this element; the result is the same as above without bases 5 to 8. The complete set of solutions calculated from this restricted set is

$$\left\{y_1=\frac{11}{3},\,y_2=\frac{25}{9},\,y_3=\frac{11}{3}\right\},\,\left\{y_1=\frac{25}{9},\,y_2=\frac{11}{3},\,y_3=\frac{11}{3}\right\},\,\left\{y_1=\frac{11}{3},\,y_2=\frac{11}{3},\,y_3=\frac{11}{3}\right\},$$

$$\left\{y_1=\frac{4\sqrt{486\,y_3^4-6696\,y_3^3+30564\,y_3^2-52200\,y_3+23750}+99\,y_3^2-582\,y_3-125}{27\,y_3^3-198\,y_3+75}\right.,$$

$$\left.y_2=\frac{4\sqrt{486\,y_3^4-6696\,y_3^3+30564\,y_3^2-52200\,y_3+23750}+99\,y_3^2-582\,y_3-125}{27\,y_3^3-198\,y_3+75}\right\},$$

$$\left\{y_1=\frac{4\sqrt{486\,y_3^4-6696\,y_3^3+30564\,y_3^2-52200\,y_3+23750}+99\,y_3^2-582\,y_3-125}{27\,y_3^3-198\,y_3+75}\right.,$$

$$\left.y_2=\frac{4\sqrt{486\,y_3^4-6696\,y_3^3+30564\,y_3^2-52200\,y_3+23750}+99\,y_3^2-582\,y_3-125}{27\,y_3^3-198\,y_3+75}\right\}$$

Note that some of the isolated solutions are also contained in the parameterized ones (as double solutions).

Example WALTER**:** Systems of polynomial equations arise in problems of parameter identification too. The following example is taken from [22]:

$$
\begin{aligned}
0 &= \theta_4 + \hat{\theta}_2 - \hat{\theta}_4 \\
0 &= \theta_1\hat{\theta}_2 - \theta_1\hat{\theta}_4 + \theta_3 - \theta_3\hat{\theta}_1 - \theta_3\hat{\theta}_3 - \theta_3\hat{\theta}_4 + \hat{\theta}_1\hat{\theta}_3 + \hat{\theta}_1\hat{\theta}_4 + \hat{\theta}_3\hat{\theta}_4 \\
0 &= \theta_2\hat{\theta}_2 - \theta_2\hat{\theta}_4 - \theta_3 + \theta_3\hat{\theta}_1 + \theta_3\hat{\theta}_2 + \theta_3\hat{\theta}_3 - \hat{\theta}_1\hat{\theta}_2 \\
0 &= -\hat{\theta}_1\hat{\theta}_3 - \hat{\theta}_2^2 - \hat{\theta}_2\hat{\theta}_3 + \hat{\theta}_2\hat{\theta}_4 \\
0 &= \theta_3^3 - \theta_3^2\hat{\theta}_1 - \theta_3^2\hat{\theta}_3 - \theta_3\hat{\theta}_4 + \theta_3\hat{\theta}_1\hat{\theta}_3 \\
&\quad +\theta_3^2\hat{\theta}_1\hat{\theta}_4 + \theta_3\hat{\theta}_3\hat{\theta}_4 - \hat{\theta}_1\hat{\theta}_3\hat{\theta}_4
\end{aligned}
$$

The $(\theta_1, \theta_2, \theta_3, \theta_4)$ are the variables, while the $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4)$ are formal parameters. WALTHER, LECOURTIER and RAKSANYI describe a solution procedure which is very close to the Buchberger algorithm. With the factorizing variant however, the corresponding set of Groebner bases gives a better decomposition of the problem:

$$
\begin{aligned}
&\{\{-\theta_1\hat{\theta}_2 + \theta_1\hat{\theta}_4 - \hat{\theta}_3\hat{\theta}_4, -\theta_2\hat{\theta}_2 + \theta_2\hat{\theta}_4 + \hat{\theta}_2^2 \\
&\quad +\hat{\theta}_2\hat{\theta}_3 - \hat{\theta}_2\hat{\theta}_4, \theta_4 + \hat{\theta}_2 - \hat{\theta}_4, \theta_3 - \hat{\theta}_1\}, \\
&\{-\theta_1\hat{\theta}_2 + \theta_1\hat{\theta}_4 - \hat{\theta}_1\hat{\theta}_4, -\theta_2\hat{\theta}_2 + \theta_2\hat{\theta}_4 + \hat{\theta}_1\hat{\theta}_2 + \hat{\theta}_2 - \hat{\theta}_2\hat{\theta}_4, \\
&\quad \theta_4 + \hat{\theta}_2 - \hat{\theta}_4, \theta_3 - \hat{\theta}_3\}, \\
&\{-\theta_1\hat{\theta}_2 + \theta_1\hat{\theta}_4 - \hat{\theta}_1\hat{\theta}_3, -\theta_2\hat{\theta}_2 + \theta_2\hat{\theta}_4 + \hat{\theta}_1\hat{\theta}_2 \\
&\quad +\hat{\theta}_1\hat{\theta}_3 - \hat{\theta}_1\hat{\theta}_4 + \hat{\theta}_2^2 + \hat{\theta}_2\hat{\theta}_3 - 2\hat{\theta}_2\hat{\theta}_4 - \hat{\theta}_3\hat{\theta}_4 + \hat{\theta}_4^2, \\
&\quad \theta_4 + \hat{\theta}_2 - \hat{\theta}_4, \theta_3 - \hat{\theta}_4\}\}
\end{aligned}
$$

Each of the solutions contains four linear polynomials. A further processing is not necessary.

The same happens with the following system taken from [22]:

$$
\begin{aligned}
0 &= x_1^2 - x_1x_2 - x_1x_3 + x_2x_3 \\
0 &= -x_2x_3 + 82x_2 \\
0 &= x_1x_3 - x_1 - 2x_3 + 2
\end{aligned}
$$

Here the second polynomial is already factorizable; the factorizing Buchberger algorithm detects more factorizable polynomials in the ideal generated by this system, e.g. $(x_2 - x_1)(x_3 - x_1)$ and the final system is completely decomposed to degree 1:

$$\{\{x_2, x_3 - 2, x_1 - 2\}, \{x_2, x_3 - 1, x_1 - 1\}, \{x_2, x_3 - 1, x_1\}, \{x_2 - 2, x_3 - 82, x_1 - 2\}\}$$

More calculations are cited in [18].

Conclusion

We have demonstrated, that Buchberger's algorithm is a useful tool for the analysis of steady states of some reaction systems. This algorithm is one of the "modern" approaches to classical algebraic problems, which gain an increasing importance when supported by modern computer algebra systems. In the past the application of symbolic methods was limited to small problems. It has been shown, that the scope of solvable problems could be enlarged by a careful adaptation of the algorithm to the algebraic properties of the problem class and that algebraic computation can give useful results for problems in chemical engineering. However, the authors regard this paper as a first step only; the expansion of the application area of computer algebra towards the applied sciences and engineering remains a permanent challenge.

References

[1] J. W. Anderson, W. F. Galway, R. R. Kessler, H. Melenk, W. Neun: *Implementing and Optimizing Lisp for the Cray.* IEEE Software, July 1987.

[2] H. G. Bock: *Numerical Treatment of Inverse Problems in Chemical Reaction Kinetics.* In [7], pp. 102–125 (1981).

[3] W. Böge, R. Gebauer, H. Kredel: *Some Examples for Solving Systems of Algebraic Equations by Calculating Groebner Bases.* J. Symb. Comp. 1 (1986).

[4] B. Buchberger: *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal.* Ph.D. thesis, Innsbruck (1965).

[5] B. Buchberger: *Groebner bases: An algorithmic method in polynomial ideal theory.* (In: Multidimensional Systems Theory, ed.: N.K. Bose) D. Reidel Publ. Comp., pp. 184–232 (1981).

[6] A. Dress: *Private communication* 1987.

[7] K. H. Ebert, P. Deuflhard, W. Jaeger (ed.): *Modelling of Chemical Reaction Systems.* Springer Ser. Chem. Phys. 18 (1981).

[8] L. A. Farrow, D. Edelson: *The steady-state approximation: fact or fiction?* Int. J. Chem. Kin. 6, pp. 787–800 (1974).

[9] M. Feinberg: *Chemical Reaction Network Structure and the stability of Complex Isothermal Reactors: I. The Deficiency Zero and Deficiency One Theorems.* To appear in Chemical Engineering Science.

[10] J. Fitch: *Solving algebraic problems with REDUCE.* J.Symb.Comp. (1986).

[11] R. Gebauer, H. M. Möller: *On an installation of Buchberger's algorithm.* To appear in J. Symb. Comp. (1988).

[12] A. C. Hearn: *REDUCE User's Manual, Version 3.3.* The Rand Corporation (1987).

[13] F. Heinmets: *Analog computer analysis of inducible enzyme synthesis.* Math. Biosci. 6, 57 (1969).

[14] R. Jenks et al.: *Scratchpad II, an experimental computer algebra system, abbreviated primer and examples.* IBM Thomas Watson Research Center, Yorktown Heights, N.Y. 10598.

[15] D. E. Knuth: *The Art of Computer Programming: Seminumerical Algorithms.* Vol. 2, 2nd edition (1981).

[16] H. Melenk, W. Neun: *REDUCE User's Guide for CRAY 1/ X-MP Series Running COS.* Konrad-Zuse-Zentrum für Informationstechnik Berlin, Technical Report TR 87-4 (1987).

[17] H. Melenk, W. Neun: *Usage of Vector Hardware for LISP Processing.* Konrad-Zuse-Zentrum für Informationstechnik Berlin (1986).

[18] H. Melenk, H. M. Möller, W. Neun: *On Groebner Bases Computation on a Supercomputer Using REDUCE* Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 88-2 (1988).

[19] H. M. Möller: *On the construction of Groebner bases using syzygies.* To appear in J. Symb. Comp. (1988).

[20] P. Schlosser, M. Feinberg: *A Graphical Determination of the Possibility of Multiple Steady States in Complex Isothermal CFSTRs.* Presented at the AIChE Annual Meeting, Miami, Florida (1986).

[21] Eric Walter, Yves Lecourtier: *Global Approaches to Identifiability for Linear and Nonlinear State Space Models.* Mathematics and Computers in Simulation, Vol. 24, North-Holland (1982).

[22] Eric Walter, Yves Lecourtier, Attilla Raksanyi, John Happel: *On the Distinguishability of Parametric Models with Different Structures.* Mathematics and Computers in Biomedical Applications, North-Holland (1985).

[23] P. S. Wang: *A p-adic algorithm for univariate partial fractions.* Proc. SYMSAC, ACM (1981).

[24] P. S. Wang, M. J. T. Guy, J. H. Davenport: *P-adic Reconstruction of Rational Numbers.* ACM Sigsam Bull., Vol 16, 2 (1982).

Veröffentlichungen des Konrad-Zuse-Zentrums für Informationstechnik Berlin
Preprints November 1988

SC 86-1. P. Deuflhard; U. Nowak. *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems.*

SC 86-2. H. Melenk; W. Neun. *Portable Standard LISP for CRAY X-MP Computers.*

SC 87-1. J. Anderson; W. Galway; R. Kessler; H. Melenk; W. Neun. *The Implementation and Optimization of Portable Standard LISP for the CRAY.*

SC 87-2. Randolph E. Bank; Todd F. Dupont; Harry Yserentant. *The Hierarchical Basis Multigrid Method.* (vergriffen)

SC 87-3. Peter Deuflhard. *Uniqueness Theorems for Stiff ODE Initial Value Problems.*

SC 87-4. Rainer Buhtz. *CGM-Concepts and their Realization.*

SC 87-5. P. Deuflhard. *A Note on Extrapolation Methods for Second Order ODE Systems.*

SC 87-6. Harry Yserentant. *Preconditioning Indefinite Discretization Matrices.*

SC 88-1. Winfried Neun; Herbert Melenk. *Implementation of the LISP-Arbitrary Precision Arithmetic for a Vector Processor.*

SC 88-2. H. Melenk; H.M. Möller; W. Neun. *On Gröbner Bases Computation on a Supercomputer Using REDUCE.*

SC 88-3. J. C. Alexander; B. Fiedler. *Global Decoupling of Coupled Symmetric Oscillators.*

SC 88-4. Herbert Melenk; Winfried Neun. *Parallel Polynomial Operations in the Buchberger Algorithm.*

SC 88-5. P. Deuflhard; P. Leinen; H. Yserentant. *Concepts of an Adaptive Hierarchical Finite Element Code.*

SC 88-6. P. Deuflhard; M. Wulkow. *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable*

Veröffentlichungen des Konrad-Zuse-Zentrums für Informationstechnik Berlin
Technical Reports November 1988

TR 86-1. H.J. Schuster. *Tätigkeitsbericht 1985.* (vergriffen)

TR 87-1. Hubert Busch; Uwe Pöhle; Wolfgang Stech. *CRAY-Handbuch. - Einführung in die Benutzung der CRAY..*

TR 87-2. Herbert Melenk; Winfried Neun. *Portable Standard LISP Implementation for CRAY X-MP Computers. Release of PSL 3.4 for COS.*

TR 87-3. Herbert Melenk; Winfried Neun. *Portable Common LISP Subset Implementation for CRAY X-MP Computers.*

TR 87-4. Herbert Melenk; Winfried Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running COS Version 3.2.*

TR 87-5. Herbert Melenk; Winfried Neun. *REDUCE Users Guide for the CRAY 1 / X-MP Series Running COS. Version 3.2.*

TR 87-6. Rainer Buhtz; Jens Langendorf; Olaf Paetsch; Danuta Anna Buhtz. *ZUGRIFF - Eine vereinheitlichte Datenspezifikation für graphische Darstellungen und ihre graphische Aufbereitung.*

TR 87-7. J. Langendorf; O. Paetsch. *GRAZIL (Graphical ZIB Language).*

TR 88-1. Rainer Buhtz; Danuta Anna Buhtz. *TDLG 3.1 - Ein interaktives Programm zur Darstellung dreidimensionaler Modelle auf Rastergraphikgeräten.*

TR 88-2. Herbert Melenk; Winfried Neun. *REDUCE User's Guide for the CRAY 1 / CRAY X-MP Series Running UNICOS. Version 3.3.*

TR 88-3. Herbert Melenk; Winfried Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running UNICOS. Version 3.3.*

TR 88-4. Danuta Anna Buhtz; Jens Langendorf; OLaf Paetsch. *GRAZIL-3D. Ein graphisches Anwendungsprogramm zur Darstellung von Kurven- und Funktionsverläufen im räumlichen Koordinatensystem.*