

# APPROXIMATE COMPUTATION OF ZERO-DIMENSIONAL POLYNOMIAL IDEALS

DANIEL HELDT, MARTIN KREUZER, SEBASTIAN POKUTTA, AND HENNIE POULISSE

ABSTRACT. The Buchberger-Möller algorithm is a well-known efficient tool for computing the vanishing ideal of a finite set of points. If the coordinates of the points are (imprecise) measured data, the resulting Gröbner basis is numerically unstable. In this paper we introduce a numerically stable *Approximate Vanishing Ideal (AVI) Algorithm* which computes a set of polynomials that almost vanish at the given points and almost form a border basis. Moreover, we provide a modification of this algorithm which produces a Macaulay basis of an approximate vanishing ideal. We also generalize the Border Basis Algorithm ([11]) to the approximate setting and study the approximate membership problem for zero-dimensional polynomial ideals. The algorithms are then applied to actual industrial problems.

## 1. INTRODUCTION

Let us consider the following common situation in industrial applications. A finite set of data points  $\mathbb{X} = \{p_1, \dots, p_s\} \subset \mathbb{R}^n$ , representing noisy measurements collected in a field experiment, is available. Their coordinates will be suggestively called here *inputs*. Furthermore, there exist one or more measured values at each point that we give the suggestive name *outputs*. Our goal is to construct polynomial functions of the coordinates of the points  $p_i$  fitting the measured inputs to the measured outputs. The polynomial model is then checked in a validation experiment: the inputs from another set of measured data points, which have not been used in the fitting process, are substituted as values for the corresponding indeterminates in the constructed polynomials. Then the evaluations obtained in this way are compared to the actual measured outputs.

This setting is markedly different from the settings for which the concept of empirical polynomials was introduced by Stetter in his ground-breaking book [20]. Not anything like a specified polynomial is given here, nor is there structural information in the form of a fixed support available. For the same reason, also the theory of pseudozero sets, as for instance laid out in [9] or [20] does not apply to the present problem.

Since the polynomial ring  $\mathbb{R}[x_1, \dots, x_n]$  is an infinite dimensional  $\mathbb{R}$ -vector space, this precludes many traditional methods of numerical linear algebra. Furthermore, there exist strong incentives to use algebraic structures such as polynomial ideals for modelling real-life applications. For instance, the output of such symbolic-numerical computations may be used as input for further symbolic treatment. In

---

*Date:* November 17, 2008.

*2000 Mathematics Subject Classification.* Primary 13P10; Secondary 41A10, 65D05, 14Q99.

*Key words and phrases.* Buchberger-Moeller algorithm, ideal membership, border basis.

The third author was supported by GIF project I-706-54.6/2001.

this way one can retrieve algebraic structures and invariants (e.g. syzygies) hidden in the input data which are not accessible via purely numerical methods.

For computing the vanishing ideal of a finite set of points, the Buchberger-Möller Algorithm (BM-Algorithm) was introduced in [2] and studied further in a number of papers (see for instance [15] and [1]). In the empirical setting described above, the exact vanishing ideal it computes does not yield polynomials to which a physical meaning can be attached because they ignore the inexact nature of the data. For instance, it computes polynomials vanishing *exactly* at the given points, but there could be polynomials passing *almost* through the points as in the following picture.

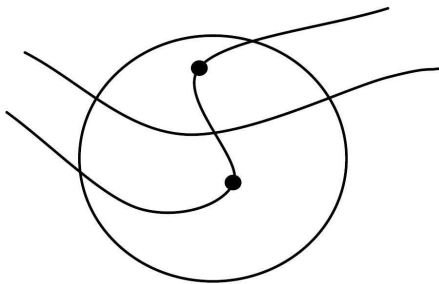


FIGURE 1. passing close vs. going through

Thus we should require merely that the polynomials vanish approximately at the points, i.e. that  $|f(p)| < \varepsilon$  for some number  $\varepsilon > 0$  which we shall call the *threshold number*. Furthermore, since the property of having small evaluations at the points of  $\mathbb{X}$  is not preserved under multiplication by scalars, we require this property only for *unitary* polynomials, i.e. for polynomials whose coefficient vector has Euclidean norm 1.

Altogether, combining the wish to generalize the notion of vanishing ideal of a set of points to the empirical setting and the need to prevent the problems we just discussed, we arrive at the following definition.

**Definition 1.1.** Let  $\mathbb{X} = \{p_1, \dots, p_s\}$  be a finite set of (empirical) points in  $\mathbb{R}^n$ , and let  $P = \mathbb{R}[x_1, \dots, x_n]$ .

- (1) The  $\mathbb{R}$ -linear map  $\text{eval} : P \rightarrow \mathbb{R}^s$  defined by  $\text{eval}(f) = (f(p_1), \dots, f(p_s))$  is called the *evaluation map* associated to  $\mathbb{X}$ .
- (2) The ideal  $I_{\mathbb{X}} = \ker(\text{eval}) \subseteq P$  is called the *vanishing ideal* of  $\mathbb{X}$ .
- (3) Given  $\varepsilon > 0$ , an ideal  $J \subseteq P$  is called an  $\varepsilon$ -*approximate vanishing ideal* of  $\mathbb{X}$  if there exists a system of generators  $G$  of  $J$  such that  $\|\text{eval}(g)\| < \varepsilon$  and  $\|g\| = 1$  for all  $g \in G$ . Here  $\|g\|$  denotes the Euclidean norm of the coefficient vector of  $g$ .

Algebraically, an approximate vanishing ideal is almost always the unit ideal. What we are really looking for are systems of generators having a particular structure which ensures that *close by* there exists a system of generators of an actual vanishing ideal. For reasons which will become apparent later, this structure is described by the notion of *approximate border bases* (cf. Definition 3.1).

In Section 2, we start by recalling some of the main properties of the singular value decomposition (SVD) of a matrix of real numbers that are used later on.

Moreover, we include a definition of the  $\varepsilon$ -*approximate kernel* of a real matrix and an interpretation in terms of a *total least squares problem*.

Then, in Section 3, the first task addressed in this paper is how to compute an approximate vanishing ideal of a set of points. We use the SVD to adapt the BM-Algorithm to the approximate setting. The resulting algorithm will be called the *Approximate Vanishing Ideal Algorithm* (AVI-Algorithm). It yields an order ideal  $\mathcal{O}$  of terms whose evaluation vectors have no small relation vectors, as well as an approximate  $\mathcal{O}$ -border basis.

A small modification of the AVI-Algorithm 3.3 computes polynomials which are most of the time close to being a Gröbner basis of an approximate vanishing ideal. However, as explained in Remark 3.8, due to the numerical instability of the concept of Gröbner basis, this cannot be certified in all situations. Another version of the AVI-Algorithm computes Macaulay bases of approximate vanishing ideals (see Corollary 3.12). A particular feature of the AVI-Algorithm is that it frequently produces the vanishing ideal of a *smaller* set of points. This is partly due to the fact that close-by points should be regarded as approximately equal and will be identified by the algorithm (see Examples 3.13 and 3.14).

In Section 4 we consider a related problem: given polynomials  $f_1, \dots, f_s \in P = \mathbb{R}[x_1, \dots, x_n]$  which “almost” define a zero-dimensional polynomial ideal (i.e. there exist close-by polynomials which define an ideal  $I \subset P$  such that  $\dim_{\mathbb{R}}(P/I) > 0$ ), find a border basis of the smallest nearby ideal  $I$ . The idea to solve this task is similar to the idea underlying the AVI-Algorithm: use the SVD to make the usual border basis algorithm (see [11], Props. 18 and 21) numerically more stable. The precise formulation of the approximate border basis algorithm is given in Theorem 4.10 and some examples and timings are provided in Section 6.

Next we study the approximate ideal membership problem in Section 5. For approximate vanishing ideals, the *decision problem* “Is  $f$  approximately in  $I$ ?” can be easily solved using the evaluation vector of  $f$ . For general zero-dimensional ideals, we use a completely reduced, orthogonal Macaulay basis to decide approximate membership by checking the length of the orthogonal projection to the ideal. Moreover, the division by this completely reduced, orthogonal Macaulay basis yields representations which enable us to solve the *explicit membership problem* for approximate ideals.

In Section 6 we provide some timings and study the behavior of the implementations of our algorithms for some real-world data sets. We also show the importance of appropriate data scaling. Finally, in Section 7, we explain how one can apply the results to some concrete industrial problems.

Before we begin with the main part of the paper, we would like to state a few restrictions we have made. We hasten to add that these restrictions do not obstruct in any way the real-life applicability of our results. One assumption has been tacitly made already: We assume that the relation between inputs and outputs mentioned above is an algebraic rather than a differential equation. A second assumption is that, again with reference to the setting sketched above, we restrict ourselves to the situation where we consider one output depending on several inputs. We plan to address multi-input, multi-output situations in a future paper.

Furthermore, we would like to point out that the present paper reflects our ongoing research in this area. While in the process of finishing our preprint, the paper [5] by Fassino was brought to our attention, in which the mathematical

problem examined in our Section 3 is addressed, albeit using methods that are clearly different from ours.

Unless explicitly stated otherwise, we use the definitions and notations introduced in [12] and [13]. The base field will be the field of real numbers throughout this paper. We leave it to the interested readers to write down the appropriate versions of our results over the field of complex numbers.

## 2. THE SINGULAR VALUE DECOMPOSITION

The Singular Value Decomposition (SVD) of a matrix of real numbers is a ubiquitous tool in numerical linear algebra. Since we are going to use it heavily (as well as certain variants of it), we recall it here. For further details, see [6]. Unless specified explicitly, we shall always equip  $\mathbb{R}^m$  with the standard scalar product and the Euclidean norm. With a slight abuse of notation, by the kernel of a matrix we shall mean the kernel of its associated linear map.

**Theorem 2.1** (The Singular Value Decomposition).

Let  $\mathcal{A} \in \text{Mat}_{m,n}(\mathbb{R})$ .

- (1) There are orthogonal matrices  $\mathcal{U} \in \text{Mat}_{m,m}(\mathbb{R})$  and  $\mathcal{V} \in \text{Mat}_{n,n}(\mathbb{R})$  and a matrix  $\mathcal{S} \in \text{Mat}_{m,n}(\mathbb{R})$  of the form  $\mathcal{S} = \begin{pmatrix} \mathcal{D} & 0 \\ 0 & 0 \end{pmatrix}$  such that

$$\mathcal{A} = \mathcal{U} \cdot \mathcal{S} \cdot \mathcal{V}^{\text{tr}} = \mathcal{U} \cdot \begin{pmatrix} \mathcal{D} & 0 \\ 0 & 0 \end{pmatrix} \cdot \mathcal{V}^{\text{tr}}$$

where  $\mathcal{D} = \text{diag}(s_1, \dots, s_r)$  is a diagonal matrix.

- (2) In this decomposition, it is possible to achieve  $s_1 \geq s_2 \geq \dots \geq s_r > 0$ . The numbers  $s_1, \dots, s_r$  depend only on  $\mathcal{A}$  and are called the **singular values** of  $\mathcal{A}$ .
- (3) The number  $r$  is the rank of  $\mathcal{A}$ .
- (4) The matrices  $\mathcal{U}$  and  $\mathcal{V}$  have the following interpretation:

$$\begin{aligned} \text{first } r \text{ columns of } \mathcal{U} &\equiv \text{ONB of the column space of } \mathcal{A} \\ \text{last } m - r \text{ columns of } \mathcal{U} &\equiv \text{ONB of the kernel of } \mathcal{A}^{\text{tr}} \\ \text{first } r \text{ columns of } \mathcal{V} &\equiv \text{ONB of the row space of } \mathcal{A} \\ &\equiv \text{ONB of the column space of } \mathcal{A}^{\text{tr}} \\ \text{last } n - r \text{ columns of } \mathcal{V} &\equiv \text{ONB of the kernel of } \mathcal{A} \end{aligned}$$

*Proof.* See for instance [6], Sections 2.5.3 and 2.6.1. □

The SVD of a real matrix allows us to define and compute its *approximate kernel*.

**Corollary 2.2.** Let  $\mathcal{A} \in \text{Mat}_{m,n}(\mathbb{R})$ , and let  $\varepsilon > 0$  be given. Let  $k \in \{1, \dots, r\}$  be chosen such that  $s_k > \varepsilon \geq s_{k+1}$ . Form the matrix  $\tilde{\mathcal{A}} = \mathcal{U} \tilde{\mathcal{S}} \mathcal{V}^{\text{tr}}$  by setting  $s_{k+1} = \dots = s_r = 0$  in  $\mathcal{S}$ .

- (1) We have  $\min\{\|\mathcal{A} - \mathcal{B}\| : \text{rank}(\mathcal{B}) \leq k\} = \|\mathcal{A} - \tilde{\mathcal{A}}\| = s_{k+1}$ . (Here  $\|\dots\|$  denotes the 2-operator norm of a matrix.)
- (2) The vector subspace  $\text{apker}(\mathcal{A}, \varepsilon) = \ker(\tilde{\mathcal{A}})$  is the largest dimensional kernel of a matrix whose Euclidean distance from  $\mathcal{A}$  is at most  $\varepsilon$ . It is called the  $\varepsilon$ -approximate kernel of  $\mathcal{A}$ .

- (3) The last  $n - k$  columns  $v_{k+1}, \dots, v_n$  of  $\mathcal{V}$  are an ONB of  $\text{apker}(\mathcal{A}, \varepsilon)$ . They satisfy  $\|\mathcal{A}v_i\| < \varepsilon$ .

*Proof.* See [6], Section 2.5.4 and the theorem. For (3), observe that  $\|\mathcal{A}v_i\| = \|(\mathcal{A} - \tilde{\mathcal{A}})v_i\| \leq \|\mathcal{A} - \tilde{\mathcal{A}}\| < \varepsilon$ .  $\square$

The number  $\varepsilon > 0$  in this corollary will be called the *threshold number*. For matrices arising from measured data, there is sometimes a large gap in the sequence of singular values  $(s_1, \dots, s_r)$ , so that there exists a natural choice for the threshold number. The matrix  $\tilde{\mathcal{A}}$  is also called the  $\varepsilon$ -truncation of the SVD of  $\mathcal{A}$  and  $k$  is sometimes referred to as the *numerical rank* of  $\mathcal{A}$ . The approximate kernel of a matrix can be reinterpreted as follows.

**TLS Interpretation of the Approximate Kernel.** The following explanations follow those in [7] and reinterpret the results in our context. Rather than using the classical least squares methods, our setting leads us to consider *total least squares* (TLS) problems. For instance, suppose we are given a finite set of *input* data points  $\mathbb{X} = \{p_1, \dots, p_s\} \subset \mathbb{R}^n$  and measured *output* values  $q_1, \dots, q_s \in \mathbb{R}$ . If we want to interpolate these data linearly, we are looking for an  $n$ -dimensional affine subspace  $\hat{\mathbf{c}}^\perp = (c_1, \dots, c_n, 1)^\perp$  that is nearest to the data points, i.e. that minimizes  $J(\mathbf{c}) = \sum_{i=1}^s (\mathbf{z}_i^{\text{tr}} \cdot \hat{\mathbf{c}})^2$  where  $\mathbf{z}_i = (p_i, -q_i)$ . Since  $\mathbf{z}_i^{\text{tr}} \cdot \hat{\mathbf{c}} / \|\hat{\mathbf{c}}\|$  is the distance from  $\mathbf{z}_i$  to the subspace, we want to minimize  $\mathbf{r} \mapsto \|(\mathbf{z}_1, \dots, \mathbf{z}_s)^{\text{tr}} \cdot \mathbf{r}\|^2$  subject to  $\|\mathbf{r}\|^2 = 1$ . In other words, the total least-squares solution minimizes the Euclidean distance of an affine hyperspace to a given set of input/output points, and not only the output components of the distances. In our applications this means that we allow errors in *all* components rather than only the right hand side of the fitting problem.

Now let us connect these TLS approximations to the interpolation problem in higher degrees and to the approximate kernel of a matrix. Let  $\mathcal{A} \in \text{Mat}_{m,n}(\mathbb{R})$  and  $i \in \{1, \dots, n\}$ . We use the choice of this component index  $i$  to dehomogenize the linear system of equations  $\mathcal{A} \cdot \mathbf{x} = 0$ . Let  $\mathcal{A}'$  be the matrix obtained by deleting the  $i^{\text{th}}$  column of  $\mathcal{A}$ , and let  $a_i$  be the  $i^{\text{th}}$  column of  $\mathcal{A}$ . The TLS solution of the (usually over-determined) linear system  $\mathcal{A}' \cdot \mathbf{x} = -a_i$  minimizes the sum of the Euclidean distances of the column vectors of  $\mathcal{A}'$  to an affine subspace  $(c_1, \dots, \hat{c}_i, \dots, c_n)^\perp$ . If it exists, it corresponds to the kernel of the minimizer of the Frobenius norm  $\|\mathcal{A} - \mathcal{B}\|^2$  subject to  $\text{rank}(\mathcal{B}) < n$  (see [7], Sec. 5). This minimization problem is solved by the SVD, and the right singular vector corresponding to the smallest singular value of  $\mathcal{A}$  is the required solution of the  $i^{\text{th}}$  TLS problem provided its  $i^{\text{th}}$  component is not zero.

If we use a threshold number  $\varepsilon > 0$  and compute the  $\varepsilon$ -truncation  $\tilde{\mathcal{A}}$  of the SVD of  $\mathcal{A}$ , we are looking for as many solutions to the TLS-problems  $\mathcal{A}' \cdot \mathbf{x} = a_i$  as possible for which there exists a solvable linear system  $\mathcal{B}' \cdot \mathbf{x} = b_i$  which is “nearest” to the system  $\mathcal{A}' \cdot \mathbf{x} = a_i$  and not farther away than specified by the threshold number. This is exactly the way we will use the SVD in Sections 3 and 4. Note that, compared to the classical least squares solutions, our SVD approach allows implicitly that *all* columns of  $\mathcal{A}$  (which will be the evaluation vectors of terms at the input data points) contain “noise”, not just the columns corresponding to the right hand sides of the dehomogenizations (which will correspond to the evaluation vectors of the leading terms of the normalized border basis polynomials). We will come back to this point later on.

## 3. APPROXIMATE VANISHING IDEALS OF POINTS

In this paper we will use the polynomial ring  $P = \mathbb{R}[x_1, \dots, x_n]$  over the field of real numbers  $\mathbb{R}$ . Moreover, we always assume that a positive real number  $\varepsilon > 0$  is given. We will call it the *threshold number*. In the approximate world it is unlikely that a polynomial vanishes exactly at a given point. We shall say that a polynomial  $f \in P$  *vanishes  $\varepsilon$ -approximately* at a point  $p \in \mathbb{R}^n$  if  $|f(p)| < \varepsilon$ .

Of course, if the coefficients of  $f \in P$  are very small, it is always true that  $f$  vanishes  $\varepsilon$ -approximately at  $p$ . Hence we need to measure the size of a polynomial by the Euclidean norm of its coefficient vector. If this norm is one, the polynomial will be said to be *unitary*. Dividing a polynomial or a vector by its norm will be called *normalizing* it. Thus we are most interested in unitary polynomials vanishing at a given point.

Now let  $\mathbb{X} = \{p_1, \dots, p_s\} \subseteq \mathbb{R}^n$  be a finite set of points. We can evaluate the polynomials at the points of  $\mathbb{X}$ . This defines the *evaluation homomorphism*  $\text{eval}_{\mathbb{X}} : \mathbb{X} \rightarrow \mathbb{R}^s$  given by  $f \mapsto (f(p_1), \dots, f(p_s))$ . We will say that  $f \in P$  vanishes  $\varepsilon$ -approximately on  $\mathbb{X}$  if  $\|\text{eval}_{\mathbb{X}}(f)\| < \varepsilon$ .

Since it is central to this section, let us recall the following notion (see Definition 1.1): An ideal  $I \subseteq P$  is called an  $\varepsilon$ -approximate vanishing ideal of  $\mathbb{X}$  if there exist unitary polynomials  $f_1, \dots, f_m \in P$  such that  $\|\text{eval}_{\mathbb{X}}(f_i)\| < \varepsilon$  for  $i = 1, \dots, m$  and such that  $I = \langle f_1, \dots, f_m \rangle$ . In other words, we are requiring that  $I$  is generated by unitary polynomials which vanish  $\varepsilon$ -approximately on  $\mathbb{X}$ .

Our goal is to present an algorithm which uses the SVD to compute an  $\varepsilon$ -approximate vanishing ideal of a finite set of points  $\mathbb{X} = \{p_1, \dots, p_s\} \subset \mathbb{R}^n$  given by their (approximate) coordinates. To this end, we modify the usual Buchberger-Möller algorithm (see [2]) in several ways: we process all terms of some degree simultaneously, we remove “small” polynomials in the vanishing ideal, and we compute an approximate border bases (rather than a Gröbner basis).

For a definition and the basic properties of border bases, see [13], Section 6.4. The following discussion centers around the concept of an  $\varepsilon$ -approximate border basis which is defined as follows.

**Definition 3.1.** Let  $\mathcal{O} = \{t_1, \dots, t_\mu\} \subseteq \mathbb{T}^n$  be an order ideal of terms, let  $\partial\mathcal{O} = \{b_1, \dots, b_\nu\}$  be its border, and let  $G = \{g_1, \dots, g_\nu\}$  be an  $\mathcal{O}$ -border prebasis of the ideal  $I = \langle g_1, \dots, g_\nu \rangle$  in  $P$ . Recall that this means that  $g_j$  is of the form  $g_j = b_j - \sum_{i=1}^{\mu} c_{ij}t_i$  with  $c_{ij} \in \mathbb{R}$ .

For every pair  $(i, j)$  such that  $b_i, b_j$  are neighbors in  $\partial\mathcal{O}$ , we compute the normal remainder  $S'_{ij} = \text{NR}_{\mathcal{O}, G}(S_{ij})$  of the S-polynomial of  $g_i$  and  $g_j$  with respect to  $G$ . We say that  $G$  is an  $\varepsilon$ -approximate border basis of the ideal  $I = \langle G \rangle$  if we have  $\|S'_{ij}\| < \varepsilon$  for all such pairs  $(i, j)$ .

One further ingredient of our theorem is a stabilized version of Gaussian reduction. The computation of reduced row echelon forms is necessary, since we need to know all possible leading terms contained in a certain vector space of polynomials, and we have to make sure that we only accept leading terms whose corresponding leading coefficient is large enough. The following method for finding the most suitable pivot elements is fashioned after the computation of the QR decomposition. Its numerical adequacy follows from the work of Shirayanagi and Sweedler (cf. [19]).

**Lemma 3.2** (Stabilized Reduced Row Echelon Form).

Let  $A \in \text{Mat}_{m,n}(\mathbb{R})$  and  $\tau > 0$  be given. Let  $a_1, \dots, a_n$  be the columns of  $A$ . Consider the following instructions.

- (1) Let  $\lambda_1 = \|a_1\|$ . If  $\lambda_1 < \tau$ , let  $R = (0, \dots, 0) \in \text{Mat}_{m,1}(\mathbb{R})$ . Otherwise, let  $Q = ((1/\lambda_1) a_1) \in \text{Mat}_{m,1}(\mathbb{R})$  and  $R = (\lambda_1, 0, \dots, 0) \in \text{Mat}_{m,1}(\mathbb{R})$ .
- (2) For  $i = 2, \dots, n$ , compute  $q_i = a_i - \sum_{j=1}^{i-1} \langle a_i, q_j \rangle q_j$  and  $\lambda_i = \|q_i\|$ . If  $\lambda_i < \tau$ , append a zero column to  $R$ . Otherwise, append the column  $(1/\lambda_i) q_i$  to  $Q$  and the column  $(\lambda_i \langle a_1, q_1 \rangle, \dots, \lambda_i \langle a_{i-1}, q_{i-1} \rangle, \lambda_i, 0, \dots, 0)$  to  $R$ .
- (3) Starting with the last row and working upwards, use the first non-zero entry of each row of  $R$  to clean out the non-zero entries above it.
- (4) For  $i = 1, \dots, m$ , compute the norm  $\rho_i$  of the  $i$ -th row of  $R$ . If  $\rho_i < \tau$ , set this row to zero. Otherwise, divide this row by  $\rho_i$ . Then return the matrix  $R$ .

This is an algorithm which computes a matrix  $R$  in reduced row echelon form. The row space of  $R$  is contained in the row space of the matrix  $\bar{A}$  which is obtained from  $A$  by setting the columns whose norm is less than  $\tau$  to zero. Here the pivot elements of  $R$  are not 1, but its rows are unitary vectors.

Furthermore, if the rows of  $A$  are unitary and mutually orthogonal, the row vectors of  $R$  differ by less than  $\tau m \sqrt{n}$  from unitary vectors in the row space of  $A$ .

*Proof.* The explanation of this algorithm is straightforward: the matrix  $Q$  contains an orthonormal basis of the column space of the matrix  $\bar{A}$  which is obtained from  $A$  by removing the columns whose norm is less than  $\tau$ . The columns of the matrix  $\bar{R}$ , defined to be the matrix  $R$  after step (2), contain the coordinates of the columns of  $\bar{A}$  in this orthonormal basis. Since we have  $\bar{R} = Q^{\text{tr}} \bar{A}$ , the row spaces of  $\bar{R}$  and  $\bar{A}$  agree. The row space of  $R$  is contained in that of  $\bar{R}$ .

It remains to prove the last claim. A row  $v$  of  $R$  can be written as a linear combination  $v = c_1 w_1 + \dots + c_m w_m$  of the rows  $w_i$  of  $\bar{R}$ , where  $c_i \in \mathbb{R}$ . Furthermore, for  $i = 1, \dots, m$ , we use  $\bar{R} = Q^{\text{tr}} \bar{A}$  to write  $w_i = q_{i1} u_1 + \dots + q_{im} u_m$  where  $u_1, \dots, u_m$  are the rows of  $\bar{A}$  and  $q_{ij} \in K$ . Now each row  $u_j$  differs from the corresponding row  $r_j$  of  $A$  by at most  $n$  entries which have been set to zero, each of which has an absolute value smaller than  $\tau$ . Hence we have  $\|u_j - r_j\| < \tau \sqrt{n}$ .

If we let  $\tilde{w}_i = q_{i1} r_1 + \dots + q_{im} r_m$  and  $\tilde{v} = c_1 \tilde{w}_1 + \dots + c_m \tilde{w}_m$ , then  $\|v - \tilde{v}\| \leq \|\sum_{i,j=1}^m c_i q_{ij} (u_j - r_j)\| \leq \tau \sqrt{n} \sum_{j=1}^m \|\sum_{i=1}^m c_i q_{ij}\| \leq \tau m \sqrt{n}$  since  $Q$  is orthogonal and  $\|v\| = 1$ . From this the claim follows.  $\square$

It is clear that the procedure in this lemma can be stabilized even further by using the modified Gram-Schmidt method, Givens or Householder transformations. Moreover, the given error estimates are quite crude and could be refined. Nevertheless, the stated simple version of the lemma suffices for the following main result of this section and does not deviate the reader's attention from the main algorithm.

Before plunging into it, let us insert one more explanation. By scaling the coordinates of the points of  $\mathbb{X}$  appropriately, we may assume that they are in the interval  $[-1, 1]$ . This assumption not only makes it easier to formulate error estimates for our algorithm, it also enhances the numerical stability of the algorithm. This will become evident in Section 7.

**Theorem 3.3** (The Approximate Vanishing Ideal Algorithm (AVI-Algorithm)).  
 Let  $\mathbb{X} = \{p_1, \dots, p_s\} \subset [-1, 1]^n \subset \mathbb{R}^n$ , let  $P = \mathbb{R}[x_1, \dots, x_n]$ , let  $\text{eval} : P \rightarrow \mathbb{R}^s$  be the associated evaluation map  $\text{eval}(f) = (f(p_1), \dots, f(p_s))$ , and let  $\varepsilon > \tau > 0$  be small positive numbers. Moreover, we choose a degree compatible term ordering  $\sigma$ . Consider the following sequence of instructions.

- A1** Start with lists  $G = \emptyset$ ,  $\mathcal{O} = [1]$ , a matrix  $\mathcal{M} = (1, \dots, 1)^{\text{tr}} \in \text{Mat}_{s,1}(\mathbb{R})$ , and  $d = 0$ .
- A2** Increase  $d$  by one and let  $L$  be the list of all terms of degree  $d$  in  $\partial\mathcal{O}$ , ordered decreasingly w.r.t.  $\sigma$ . If  $L = \emptyset$ , return the pair  $(G, \mathcal{O})$  and stop. Otherwise, let  $L = (t_1, \dots, t_\ell)$ .
- A3** Let  $m$  be the number of columns of  $\mathcal{M}$ . Form the matrix

$$\mathcal{A} = (\text{eval}(t_1), \dots, \text{eval}(t_\ell), \mathcal{M}) \in \text{Mat}_{s, \ell+m}(\mathbb{R}).$$

Using its SVD, calculate a matrix  $\mathcal{B}$  whose column vectors are an ONB of the approximate kernel  $\text{apker}(\mathcal{A}, \varepsilon)$ .

- A4** Using the lemma, compute the stabilized reduced row echelon form of  $\mathcal{B}^{\text{tr}}$  with respect to the given  $\tau$ . The result is a matrix  $\mathcal{C} = (c_{ij}) \in \text{Mat}_{k, \ell+m}(\mathbb{R})$  such that  $c_{ij} = 0$  for  $j < \nu(i)$ . Here  $\nu(i)$  denotes the column index of the pivot element in the  $i^{\text{th}}$  row of  $\mathcal{C}$ .
- A5** For all  $j \in \{1, \dots, \ell\}$  such that there exists an  $i \in \{1, \dots, k\}$  with  $\nu(i) = j$  (i.e. for the column indices of the pivot elements), append the polynomial

$$c_{ij}t_j + \sum_{j'=j+1}^{\ell} c_{ij'}t_{j'} + \sum_{j'=\ell+1}^{\ell+m} c_{ij'}u_{j'}$$

to the list  $G$ , where  $u_{j'}$  is the  $(j' - \ell)^{\text{th}}$  element of  $\mathcal{O}$ .

- A6** For all  $j = \ell, \ell - 1, \dots, 1$  such that the  $j^{\text{th}}$  column of  $\mathcal{C}$  contains no pivot element, append the term  $t_j$  as a new first element to  $\mathcal{O}$  and append the column  $\text{eval}(t_j)$  as a new first column to  $\mathcal{M}$ .
- A7** Using the SVD of  $\mathcal{M}$ , calculate a matrix  $\mathcal{B}$  whose column vectors are an ONB of  $\text{apker}(\mathcal{M}, \varepsilon)$ .
- A8** Repeat steps **A4** – **A7** until  $\mathcal{B}$  is empty. Then continue with step **A2**.

This is an algorithm which computes a pair  $(G, \mathcal{O})$  of sets  $G = \{g_1, \dots, g_\nu\}$  and  $\mathcal{O} = \{t_1, \dots, t_\mu\}$  with the following properties:

- (a) The set  $G$  consists of unitary polynomials which generate a  $\delta$ -approximate vanishing ideal of  $\mathbb{X}$ , where  $\delta = \varepsilon\sqrt{\nu} + \tau\nu(\mu + \nu)$ .
- (b) The set  $\mathcal{O} = \{t_1, \dots, t_\mu\}$  contains an order ideal of terms such that there is no unitary polynomial in  $\langle \mathcal{O} \rangle_K$  which vanishes  $\varepsilon$ -approximately on  $\mathbb{X}$ .
- (c) The set  $\tilde{G} = \{(1/\text{LC}_\sigma(g))g \mid g \in G\}$  is an  $\mathcal{O}$ -border prebasis.
- (d) The set  $\tilde{G}$  is an  $\eta$ -approximate border basis for  $\eta = 2\delta + 2\nu\delta^2/\gamma\varepsilon + 2\nu\delta\sqrt{s}/\varepsilon$ . Here  $\gamma$  denotes the smallest absolute value of the border term coefficient of one the polynomials  $g_i$ .

*Proof.* First we prove finiteness. When a new degree is started in step **A2**, the matrix  $\mathcal{M}$  has  $m = \#\mathcal{O}$  columns where  $\mathcal{O}$  is the *current* list of terms. In step **A6** we enlarge  $\mathcal{M}$  by new first columns which are linearly independent of the other columns. Clearly, this can happen only finitely many times, and only finitely many terms are appended to  $\mathcal{O}$ . In step **A5** we construct new elements of  $G$  which have leading terms in  $\partial\mathcal{O}$ . Therefore also this can happen only finitely many times.



Eventually we arrive at a situation where all new columns  $\text{eval}(t_i)$  of  $\mathcal{A}$  in step **A3** would lead to a contradiction by yielding either a new element of  $G$  or a new column of  $\mathcal{M}$ . Thus we must eventually get  $L = \emptyset$  and the algorithm stops.

Next we show (a). Let  $g \in G$ . By the construction of the stabilized reduced row echelon form, the coefficient vector  $c$  of  $g$  is unitary. The vector  $c$  differs by less than  $\tau m \sqrt{n}$  from a unitary vector  $\tilde{c}$  in  $\text{apker}(A)$ . Moreover, the columns of  $\mathcal{A}$  are the evaluation vectors of terms which are ordered decreasingly w.r.t.  $\sigma$ . Note that  $\tilde{c}$  is a unitary vector in the column space of  $\mathcal{B}$ . Let  $\tilde{g} = \sum_{i=1}^{\ell+m} \tilde{c}_i t_i$  be its associated polynomial. We write  $\tilde{c} = a_1 b_1 + \dots + a_k b_k$  where  $a_j \in \mathbb{R}$  and  $b_j$  is the  $j^{\text{th}}$  column of  $\mathcal{B}$ . Then we have  $\|\text{eval}(\tilde{g})\| = \|\sum_{j=1}^k a_j \text{eval}(\sum_{i=1}^{\ell+m} b_i t_i)\| \leq \sum_{j=1}^k |a_j| \varepsilon \leq \varepsilon \sqrt{k} \sum_{j=1}^k |a_j|^2 = \varepsilon \sqrt{k}$ . To get a bound for  $k$ , we can use  $k \leq \#\partial\mathcal{O} = \nu$ . By the lemma, we have  $\|g - \tilde{g}\| < \tau \nu \sqrt{\mu + \nu}$  if we use  $\nu$  as a (crude) upper bound for the number of rows of  $\mathcal{B}^{\text{tr}}$  and  $\mu + \nu$  for the number of its columns. Since  $\mathbb{X} \subset [-1, 1]^n$ , the norm of the evaluation vector of a term is  $\leq 1$ . The number of terms in the support of  $g - \tilde{g}$  is bounded by  $\mu + \nu$ . Hence we obtain  $\|\text{eval}(g - \tilde{g})\| \leq \|g - \tilde{g}\| \sqrt{\mu + \nu} < \tau \mu (\mu + \nu)$ .

To prove (b), we observe that the columns of the final matrix  $\mathcal{M}$  are precisely the evaluation vectors of the terms in  $\mathcal{O}$ . After the loop in steps **A4** – **A8**, we have  $\text{apker}(\mathcal{M}) = \{0\}$ . Hence no unitary polynomial in  $\langle \mathcal{O} \rangle_K$  has an evaluation vector which is smaller than  $\varepsilon$ . Now we show that  $\mathcal{O}$  is an order ideal of terms. Suppose that  $t \in \mathcal{O}$  and  $x_i t \in \partial\mathcal{O}$  is put into  $\mathcal{O}$  in step **A6**. We have to show that all divisors of  $x_i t$  are in  $\mathcal{O}$ . If not, there exists an indeterminate  $x_j$  such that  $t = x_j t'$  and  $x_i t' = \text{LT}_\sigma(g)$  for some  $g \in G$ . Then  $x_i t = \text{LT}_\sigma(x_j g)$  is the leading term of a unitary polynomial of degree  $d$  whose evaluation vector is less than  $\varepsilon$ . But the loop in **A4** – **A8** makes sure that all such leading terms are detected, in contradiction to  $x_i t \in \mathcal{O}$ . Hence all divisors  $x_i t'$  of  $x_i t$  are in  $\mathcal{O}$ .

To prove (c), we use the fact that  $\mathcal{C}$  is a reduced row echelon form. By the way  $G$  and  $\mathcal{O}$  are updated in steps **A5** and **A6**, the polynomials  $g \in G$  satisfy  $\text{Supp}(g - \text{LM}_\sigma(g)) \subseteq \mathcal{O}$ . By the way the list  $L$  is updated in **A2**, we have  $\text{LT}_\sigma(g) \in \partial\mathcal{O}$ .

Finally we show (d). We write  $g_i = \gamma_i b_i + h_i$  where  $\gamma_i \in \mathbb{R}$  is the border term coefficient,  $b_i \in \partial\mathcal{O}$  and  $\text{Supp}(h_i) \subseteq \mathcal{O}$ . Then  $\tilde{g}_i = (1/\gamma_i) g_i = b_i + \tilde{h}_i$  is the corresponding border basis element. Letting  $\omega$  be the smallest singular value of the evaluation matrix  $\mathcal{M}$  of  $\mathcal{O}$ , we have  $\|\text{eval}(\tilde{h}_i)\| = \|\mathcal{M} \cdot v_i\| \geq \omega \|v_i\| = \omega \|\tilde{h}_i\|$  where  $v$  denotes the vector of coefficients of  $\tilde{h}_i$ . This yields  $\omega \|\tilde{h}_i\| \leq \|\text{eval}(g_i)\| + \|\text{eval}(b_i)\| \leq \delta/|\gamma_i| + \sqrt{s}$ . Hence every coefficient  $c_{ij}$  of  $\tilde{h}_i$  satisfies  $c_{ij} \leq \delta/(\omega\gamma) + \sqrt{s}/\omega < \delta/(\varepsilon\gamma) + \sqrt{s}/\varepsilon$ .

Given a neighbor pair  $(i, j)$ , the S-polynomial  $S_{ij} = x_k \tilde{g}_i - x_\ell \tilde{g}_j$  resp.  $S_{ij} = \tilde{g}_i - x_k \tilde{g}_j$  has a normal remainder of the form  $S'_{ij} = \text{NR}_{\mathcal{O}, \tilde{G}}(S_{ij}) = x_k \tilde{g}_i - x_\ell \tilde{g}_j - \sum_\nu c_\nu \tilde{g}_\nu$  resp.  $S'_{ij} = \tilde{g}_i - x_k \tilde{g}_j - \sum_\nu c_\nu \tilde{g}_\nu$  where the  $c_\nu \in \mathbb{R}$  are some of the coefficients of the polynomials  $\tilde{h}_i$ . Thus we get  $\|\text{eval}(S'_{ij})\| < 2\delta + \sum_\nu (\delta/(\varepsilon\gamma) + \sqrt{s}/\varepsilon) \|\text{eval}(\tilde{g}_\nu)\| \leq \eta$ , as claimed.  $\square$

Let us collect some remarks about this theorem and its proof.

**Remark 3.4.** The assumption that the coordinates of the points of  $\mathbb{X}$  are in the interval  $[-1, 1]$  is not necessary for the correctness of the AVI-Algorithm. It was only used to prove the stated bounds for  $\delta$  and  $\eta$ . However, a suitable amount

of data scaling is essential for the performance and the numerical behavior of this algorithm, as we shall see in the last part of Section 6.

**Remark 3.5.** In the theorem, the stated bounds for  $\delta$  and  $\eta$  are rather crude. Using a more refined analysis, they could be improved significantly. In practical examples, the behavior of the computed approximate border bases is much better than predicted by these bounds. In all cases we explored, the evaluations of the polynomials  $g_i$  were smaller than  $\varepsilon$  and the evaluations of the normal remainders of the S-polynomials were of the same order of magnitude as  $\varepsilon$ . Clearly, if we choose  $\varepsilon = 0$ , the algorithm computes an exact border basis.

**Remark 3.6.** The loop started in step **A7** of the theorem is most of the time unnecessary. It has been included to deal with an unusual behavior of the approximate kernel which occurs sometimes: if the matrix  $\mathcal{A}$  satisfies  $\dim_{\mathbb{R}}(\text{apker}(\mathcal{A}, \varepsilon)) = k$  and one removes the  $k$  columns at the pivot positions of an ONB of the approximate kernel, the remaining matrix  $\tilde{\mathcal{A}}$  may still have  $\text{apker}(\tilde{\mathcal{A}}, \varepsilon) \neq \{0\}$ .

In a practical implementation of the algorithm, it is more efficient to deal with this anomaly in the following way: ignore step **A7**. Instead, if in the next degree more than  $\ell$  non-zero rows of  $\mathcal{C}$  are found, some of whose pivot positions correspond to elements of  $\mathcal{O}$ , remove those elements of  $\mathcal{O}$  and repeat the computation for the preceding degree. In this way, quite a few unnecessary SVD computations can be saved, but the formulation and the analysis of the algorithm become more cumbersome.

**Remark 3.7.** The AVI-Algorithm can be optimized in a variety of ways. For instance, it is not necessary that the “blocks” of terms used in the loop **A2** – **A8** are all the terms of degree  $d$  in  $\partial\mathcal{O}$ . Especially in higher degrees it may be useful to process “blocks” of consecutive terms for which the SVD can be computed efficiently.

**Remark 3.8.** By changing the construction of the list  $L$  in step **A2** appropriately, the AVI-Algorithm can be used to compute an “approximate Gröbner basis” of an approximate vanishing ideal of  $\mathbb{X}$ . More precisely, the list  $L$  should be defined as all terms in  $\mathbb{T}_d^n$  which are not contained in  $\langle \text{LT}_\sigma(G) \rangle$ . Unfortunately, there is no guarantee that the computed polynomials are close to an actual Gröbner basis. The computation of the normal remainders of the S-polynomials requires a number of reductions steps which can be very large. Therefore no bound for the size of the evaluation vectors of these normal remainders can be given. In many practical examples, however, the Gröbner basis version works fine.

**Remark 3.9.** The approximate border basis computed by the AVI-Algorithm is “close” to an actual border basis in the sense that its coefficients define a point close to the border basis scheme (see [14]). The task of finding an actual exact border basis close to it could be solved by computing the eigenvectors of the generic multiplication matrix corresponding to  $G$ , reading the approximate points from them, and computing their (exact) vanishing ideal. This method is rather inefficient. Another possibility is to use the Approximate Border Basis Algorithm 4.10. However, for many of the tasks studied below, the knowledge of an approximate border basis is sufficient.

**Remark 3.10.** In the theorem, the choice of a term ordering  $\sigma$  could be replaced by a suitable *choice function*. This function would have to have the property

that it guarantees that the computed set of terms  $\mathcal{O}$  is an order ideal, or else we would have to make sure that the order ideal property is retained by the choice of the pivot elements in the computation of the stable reduced row echelon form in Lemma 3.2. We leave it to the interested reader to work out explicit versions of this generalization of the AVI-Algorithm.

If the bound  $\delta$  of the theorem yields insufficient results, we can use the following optimization.

**Threshold control.** In Algorithm 3.3 we use a fixed  $\varepsilon > 0$  for the singular value truncation. In many cases this suffices to get good results, but in some cases it makes sense to vary  $\varepsilon$ . The main idea behind this is that, after using a specific  $\varepsilon$  for the singular value truncation, one checks the quality of the evaluation vectors of the relations in  $G$ . If the quality is below a given bound, the constant  $\varepsilon$  is iteratively adapted until the resulting relations meet the quality requirements. This principle will be called *threshold control*. Using threshold control for Algorithm 3.3 may involve additional iterations and hence slow the algorithm down a bit. On the other hand, it will result in a smoother evaluation behavior of the calculated relations. Threshold control is achieved by modifying the theorem as follows.

**Corollary 3.11** (AVI Algorithm with Threshold Control).

*In the setting of the theorem, let  $\xi > 0$  the maximal tolerable size of the evaluation vectors of the calculated Gröbner basis polynomials, and let  $N > 1$  (for instance  $N = 10$ ). Replace steps **A5** and **A8** by the following steps **A5'** and **A8'**.*

**A5'** *For all  $j \in \{1, \dots, \ell\}$  such that there exists a  $i \in \{1, \dots, k\}$  with  $\nu(i) = j$  (i.e. for the column indices of the pivot elements), form the polynomials*

$$g_j = c_{ij}t_j + \sum_{j'=j+1}^{\ell} c_{ij'}t_{j'} + \sum_{j'=\ell+1}^{\ell+m} c_{ij'}u_{j'}$$

*where  $u_{j'}$  is the  $(j' - \ell)^{\text{th}}$  element of  $\mathcal{O}$ . Calculate  $\max_j \{\|\text{eval}(g_j)\|\}$  and check whether it is  $< \xi$ . If this is the case, append the polynomials  $g_j$  to  $G$ . Otherwise, replace  $\varepsilon$  by  $\varepsilon/N$ , replace  $\tau$  by  $\tau/N$ , and continue with step **A3**.*

**A8'** *Repeat steps **A4** – **A7** until  $\mathcal{B}$  is empty. Then reset  $\varepsilon$  and  $\tau$  to their original values and continue with step **A2**.*

*The resulting algorithm computes a pair  $(G, \mathcal{O})$  having the properties stated in the theorem. Moreover, the polynomials  $g_j \in G$  satisfy  $\|\text{eval}(g_j)\| < \xi$ .*

*Proof.* The fact that this procedure terminates follows from the observation that the size of the computed order ideal  $\mathcal{O}$  is bounded by its size in the case  $\varepsilon = \tau = 0$ , in which it is precisely the number of points in  $\mathbb{X}$ .  $\square$

In Section 6 we present some computational results for the calculation of approximate border bases of  $\varepsilon$ -approximate vanishing ideals. In particular, it turns out that approximate border bases are indeed numerically more stable than Gröbner bases in the examples considered there.

**Macaulay Bases.** The AVI-Algorithm can be adapted to calculate a *Macaulay basis* (also called an *H-basis*) of an approximate vanishing ideal of a set of points. The definition and fundamental properties of Macaulay bases are explained in [13], Sections 4.2 and 4.3. In Section 5 we shall use these bases to study the approximate

membership problem for zero-dimensional polynomial ideals. Macaulay bases were also used by H.M. Möller and T. Sauer to address numerical problems associated with multivariate interpolation in [16] and [17].

In some sense, since the AVI-Algorithm 3.3 calculates an approximate  $\mathcal{O}_\sigma(I)$ -border basis and  $\sigma$  is a degree compatible term ordering, we have already found an approximate Macaulay basis for an approximate vanishing ideal of  $\mathbb{X}$  by [13], Cor. 4.2.16. However, this is not the kind of Macaulay basis we are really interested in. We would prefer an *almost orthogonal* Macaulay basis with good numerical properties. Here “almost orthogonal” refers to a Macaulay basis for which the orthogonality conditions of [17], Thm. 6.4 are satisfied up to an error whose size is not larger than the given threshold number (see also Definition 5.1). If one completely reduces such a set of polynomials, the result does not depend on the choice of the term ordering  $\sigma$  (cf. [17], 6.5), so that  $\sigma$  is merely a method to perform certain choices during the course of the algorithm. In order to find an almost orthogonal Macaulay basis, we modify Algorithm 3.3 as follows.

**Corollary 3.12** (Macaulay Bases for Approximate Vanishing Ideals).

*In the setting of the theorem, consider the following sequence of instructions.*

**M1** Start with lists  $G = \emptyset$ ,  $H = \emptyset$ ,  $\mathcal{O} = (1)$ ,  $\mathcal{Q} = (1)$ , a matrix  $\mathcal{M} = (1, \dots, 1)^{\text{tr}} \in \text{Mat}_{s,1}(\mathbb{R})$ , and  $d = 0$ .

**M2** Increase  $d$  by one and let  $L = (t_1, \dots, t_\ell)$  be the list of all terms of degree  $d$  which are not contained in  $\partial\mathcal{O}$ , ordered decreasingly w.r.t.  $\sigma$ . If  $L = \emptyset$ , continue with step **M10**.

**M3** Let  $m$  be the number of columns of  $\mathcal{M}$ . Form the matrix

$$\mathcal{A} = (\text{eval}(t_1), \dots, \text{eval}(t_\ell), \mathcal{M}) \in \text{Mat}_{s, \ell+m}(\mathbb{R}).$$

Using its SVD, compute a matrix  $\mathcal{B}$  whose column vectors are an ONB of the approximate kernel  $\text{apker}(\mathcal{A}, \varepsilon)$ .

**M4** Using Lemma 3.2, compute the stabilized reduced row echelon form of  $\mathcal{B}^{\text{tr}}$  with respect to the given  $\tau$ . The result is a matrix  $\mathcal{C} = (c_{ij}) \in \text{Mat}_{k, \ell+m}(\mathbb{R})$  such that  $c_{ij} = 0$  for  $j < \nu(i)$ . Here  $\nu(i)$  denotes the column index of the pivot element in the  $i^{\text{th}}$  row of  $\mathcal{C}$ . Moreover, let  $\bar{\mathcal{B}} = (\bar{b}_{ij})$  be the matrix obtained from  $\mathcal{B}^{\text{tr}}$  by setting all columns to zero whose norm is less than  $\tau$ .

**M5** For every  $i \in \{1, \dots, k\}$ , append the polynomial

$$h_i = \sum_{j'=1}^{\ell} \bar{b}_{ij'} t_{j'} + \sum_{j'=\ell+1}^{\ell+m} \bar{b}_{ij'} u_{j'}$$

to the list  $H$ , where  $u_{j'}$  is the  $(j' - \ell)^{\text{th}}$  element of  $\mathcal{O}$ .

**M6** For all  $j \in \{1, \dots, \ell\}$  such that there exists an  $i \in \{1, \dots, k\}$  with  $\nu(i) = j$  (i.e. for the column indices of the pivot elements), append the polynomial

$$g_j = c_{ij} t_j + \sum_{j'=j+1}^{\ell} c_{ij'} t_{j'} + \sum_{j'=\ell+1}^{\ell+m} c_{ij'} u_{j'}$$

to the list  $G$ , where  $u_{j'}$  is the  $(j' - \ell)^{\text{th}}$  element of  $\mathcal{O}$ .

**M7** For all  $j = \ell, \ell - 1, \dots, 1$  such that the  $j^{\text{th}}$  column of  $\mathcal{C}$  contains no pivot element, append the term  $t_j$  as a new first element to  $\mathcal{O}$  and append the column  $\text{eval}(t_j)$  as a new first column to  $\mathcal{M}$ .

- M8** Using the SVD of  $\mathcal{M}$ , calculate a matrix  $\mathcal{B}$  whose column vectors are an ONB of  $\text{apker}(\mathcal{M}, \varepsilon)$ .
- M9** Repeat steps **M4** – **M8** until  $\mathcal{B}$  is empty. Then continue with step **M2**.
- M10** Let  $\mathcal{O} = (t_1, \dots, t_\mu)$ . Compute the SVD  $\mathcal{M} = \mathcal{U}\mathcal{S}\mathcal{V}^{\text{tr}}$  of  $\mathcal{M}$ . Put the polynomials  $(q_1, \dots, q_\mu) = (t_1, \dots, t_\mu) \cdot \mathcal{V}\mathcal{S}^{-1}$  into  $\mathcal{Q}$  and return the pair  $(H, \mathcal{Q})$ .

This is an algorithm which computes a pair  $(H, \mathcal{Q})$ . Here  $H$  is an almost orthogonal Macaulay basis of a  $\delta$ -approximate vanishing ideal  $I$  of  $\mathbb{X}$  where  $\delta = \varepsilon\sqrt{\nu} + \tau\nu(\mu + \nu)$ . Moreover, the set  $\mathcal{Q}$  is an ONB of a complement of  $I$  in  $P$ .

*Proof.* Since the computation of  $G$  and  $\mathcal{O}$  proceeds as in the algorithm of the theorem, it suffices to prove the claimed properties of  $H$  and  $\mathcal{Q}$ . In each degree  $d$ , the new elements of  $H$  are obtained from the new elements of  $G$  by an invertible linear transformation, namely the transformation corresponding to the inversion of the reduction steps in Lemma 3.2. In particular, the degree forms of the new elements of  $H$  generate the same vector space as the degree forms of the new elements of  $G$ . Since  $G$  is a Macaulay basis, the set  $H$  is therefore a Macaulay basis, too. In each degree  $d$ , the new elements of  $H$  are almost orthogonal to each other, because the columns of  $\mathcal{B}$  form an ONB of  $\text{apker}(\mathcal{A})$  and  $\bar{\mathcal{B}}$  is obtained from  $\mathcal{B}$  by setting some very small columns equal to zero. More precisely, the scalar product of the coefficient vectors of any two elements of  $H$  is less than  $\tau$ .

After the last degree  $d$  has been treated by the algorithm, the list  $\mathcal{O}$  contains a vector space basis of  $P/I$  where  $I$  is a  $\delta$ -approximate vanishing ideal of  $\mathbb{X}$ . Hence  $\mathcal{O}$  is a vector space basis of  $P/\langle H \rangle$ .

Considering the SVD of  $\tilde{\mathcal{M}}$  we see that  $\tilde{\mathcal{M}}\tilde{\mathcal{V}}\tilde{\mathcal{S}}^{-1} = (u_1, \dots, u_\mu, 0, \dots, 0)$  where the  $u_i$  are the first  $\mu$  columns of  $\tilde{\mathcal{U}}$ . Consequently, the evaluation vectors of the elements of  $\mathcal{Q}$  are an ONB of the column space of  $\tilde{\mathcal{M}}$ . Since the columns of  $\tilde{\mathcal{M}}$  are exactly the evaluation vectors of the elements of  $\mathcal{O}$ , the claim follows.  $\square$

We would like to point out that the algorithm of this corollary can be optimized substantially: if no column has to be set equal to zero in the computation of the stabilized reduced row echelon form (as is normally the case), we can take the matrix  $\mathcal{B}$  instead of  $\bar{\mathcal{B}}$  in step **M5**. Similarly, if the columns we set equal to zero are few and much smaller than  $\tau$ , the polynomials derived from the columns of  $\mathcal{B}$  are a very good approximation to the polynomials constructed in step **M5**. Notice also that this algorithm produces a particularly nice Macaulay basis of a  $\delta$ -approximate vanishing ideal  $I$  of  $\mathbb{X}$  and an ONB of a complement of  $I$  in  $P$ . These facts will be used to solve the approximate ideal membership problem for  $I$  in Section 5.

**The Case of the Vanishing Points.** In Theorem 3.3 the computed order ideal  $\mathcal{O}$  satisfies the inequality  $\#\mathcal{O} \leq s$  but not necessarily the equality.

Apparently the zero-set of an ideal  $I$  generated by a border basis close to  $G$  and satisfying  $P/I = \langle \mathcal{O} \rangle_K$  may consist of *less* than  $s$  points. Is this really possible? Yes, it is! Below we exhibit examples for this phenomenon. In fact, when working with real-world data sets, it is quite likely to occur: if two measurements yield approximately the same values, the corresponding data points will be very close to each other. A polynomial of low degree will vanish approximately on one point if it vanishes approximately on the other. Thus, from the point of view of constructing

an approximate vanishing ideal, the points should be considered as one. And luckily enough, this is exactly what the SVD achieves for us, without the need for any data preprocessing. Let us have a look a simple example.

**Example 3.13.** (Two Points Become One)

Consider the set of two points  $\mathbb{X} = \{(0.25, 1), (0.3, 1)\}$  which may be considered close to each other. We apply the AVI-Algorithm 3.3. The evaluation matrix in degree one is

$$\mathcal{A} = (\text{eval}(x), \text{eval}(y), \text{eval}(1)) = \begin{pmatrix} 0.25 & 1 & 1 \\ 0.3 & 1 & 1 \end{pmatrix}$$

The singular values of this matrix are  $s_1 = 1.9699$  and  $s_2 = 0.5214$ . Given  $\varepsilon = 0.6$ , the approximate kernel of  $\mathcal{A}$  is 2-dimensional. Hence we find two linearly independent linear forms passing through  $\mathbb{X}$  (instead of a linear and a quadratic polynomial). In other words, the corresponding approximate vanishing ideal  $I$  of  $\mathbb{X}$  satisfies  $\dim_{\mathbb{R}}(P/I) = 1$ , i.e. it defines *one point*. The two points have become one!

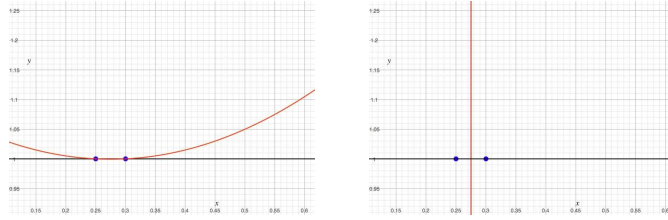


FIGURE 2. Two close-by points become one “approximate” point

Here is a more complicated example of this phenomenon.

**Example 3.14.** (Nine Points Become Five)

Consider the set of nine points  $\mathbb{X} = \{(0.264, 0.001), (0.099, 0.302), (0.103, 0.298), (0.203, -0.211), (0.198, -0.213), (-0.200, 0.209), (-0.198, 0.212), (-0.201, 0.214), (-0.266, -0.002)\}$ , some of which may be considered close to each other. We apply the AVI-Algorithm 3.3 with  $\sigma = \text{DegRevLex}$ . The evaluation matrix in degree one is

$$\begin{pmatrix} 0.264 & 0.099 & 0.103 & 0.203 & 0.198 & -0.2 & -0.198 & -0.201 & -0.266 \\ 0.001 & 0.302 & 0.298 & -0.211 & -0.213 & 0.209 & 0.212 & 0.214 & -0.002 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^{\text{tr}}$$

The singular values of this matrix are  $s_1 = 3.012$ ,  $s_2 = 0.703$ , and  $s_3 = 0.44$ .

- (1) First, let us use  $\varepsilon = 0.05$ . Then no singular value truncation is necessary in degree 1 and we set  $\mathcal{O} = \{1, x, y\}$ . In degree 2 the evaluation matrix  $\mathcal{A}$  has singular values  $s_1 = 3.018$ ,  $s_2 = 0.704$ ,  $s_3 = 0.450$ ,  $s_4 = 0.082$ ,  $s_5 = 0.061$ , and  $s_6 = 0.001$ . Therefore the singular value truncation sets  $s_6 = 0$  and yields a new matrix  $\tilde{\mathcal{A}}$  of rank 5. Then  $\text{apker}(\mathcal{A}, \varepsilon) = \ker(\tilde{\mathcal{A}})$  corresponds to the polynomial  $g_1 = 0.833x^2 - 0.01xy + 0.549y^2 + 0.001x + 0.002y - 0.058$ . The order ideal becomes  $\mathcal{O} = \{1, x, y, xy, y^2\}$ .

This shows that the points of  $\mathbb{X}$  lie  $\varepsilon$ -approximately on the ellipse  $\mathcal{Z}(g_1) \approx \mathcal{Z}(301x^2 + 198y^2 - 21)$ . When we proceed with degree 3, we find that three polynomials  $g_2, g_3, g_4$  have to be added to the approximate

border basis, but no element is appended to  $\mathcal{O}$ , and the algorithm stops. Altogether, we have computed an approximate border basis of an ideal of five points which lie approximately on an ellipse.

- (2) Now we redo the computation with  $\varepsilon = 0.0001$ . No singular value truncation has to be performed in degree 1, 2, or 3. After finishing degree 2 we have  $\mathcal{O} = \{1, x, y, x^2, xy, y^2\}$ . Then we find one polynomial in degree 3 of the  $\varepsilon$ -approximate vanishing ideal, namely  $g_1 = -0.748x^3 - 0.597x^2y - 0.225xy^2 + 0.008y^3 - 0.043x^2 - 0.099xy - 0.122y^2 + 0.052x + 0.035y + 0.002$ , and four further polynomials in degree 4. The algorithm stops after degree 4 and returns  $\mathcal{O} = \{1, x, y, x^2, xy, y^2, x^2y, xy^2, y^3\}$ . Thus the 0.0001-approximate vanishing ideal corresponds nine points which are close to the original points of  $\mathbb{X}$ .

#### 4. APPROXIMATE BORDER BASIS COMPUTATION

Next we want to address the following problem: Suppose we are given “empirical” polynomials  $f_1, \dots, f_r \in P = \mathbb{R}[x_1, \dots, x_n]$  with  $r \geq n$ . If  $r > n$ , the ideal  $\langle f_1, \dots, f_r \rangle$  will probably be the unit ideal. However, we assume that “close by” there exists a zero-dimensional ideal  $I \subset P$  with  $\dim_{\mathbb{R}}(P/I) \gg 0$ . Our task is to find  $I$  and to compute a border basis of  $I$ .

**From Polynomials to Matrices.** To begin with, let us suppose for a moment that we have found a finite dimensional vector space of polynomials which “is close to” a part of  $I_{\leq d}$ , the polynomials of degree  $\leq d$  in  $I$ . As discussed above, in our applications it makes sense to measure the size of a polynomial by the Euclidean norm of its coefficient vector. Given a threshold number  $\varepsilon > 0$ , a polynomial is called  $\varepsilon$ -small if the norm of its coefficient vector is less than  $\varepsilon$ . The  $\varepsilon$ -small polynomials form an open neighborhood of the zero polynomial. Since a polynomial ideal  $I \neq P$  contains only a small part of this neighborhood, we have to remove from  $V$  as many  $\varepsilon$ -small polynomials as possible.

In order to do this, we represent a finite dimensional vector space of polynomials by a matrix. Let  $V \subset P$  be a vector space, let  $f_1, \dots, f_r$  be a basis of  $V$ , and let  $\text{Supp}(f_1) \cup \dots \cup \text{Supp}(f_r) = \{t_1, \dots, t_s\}$ . For  $i = 1, \dots, r$ , we write  $f_i = a_{i1}t_1 + \dots + a_{is}t_s$  with  $a_{ij} \in \mathbb{R}$ . Then  $V$  can be represented by the matrix

$$\mathcal{A} = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & & \vdots \\ a_{r1} & \cdots & a_{rs} \end{pmatrix}$$

Now we may use the SVD of  $\mathcal{A}$  to filter out some  $\varepsilon$ -small polynomials in  $V$ . In the following sense, this does not change  $V$  very much.

**Definition 4.1.** Let  $V, W$  be two vector subspaces of  $P$ , and let  $\delta > 0$ .

- (1) A polynomial  $f \in P$  is called  $\delta$ -close to  $V$  if there exists a polynomial  $v \in V$  such that  $\|f - v\| < \delta$ .
- (2) We say that  $V$  is  $\delta$ -close to  $W$  if every unitary polynomial in  $V$  is  $\delta$ -close to  $W$ .

Note that the relation expressing the fact that  $V$  is  $\delta$ -close to  $W$  is not symmetric.

**Remark 4.2.** [The  $\varepsilon$ -truncation of a vector space]

Let  $V$  be a finite dimensional vector space of polynomials with basis  $\{f_1, \dots, f_r\}$ , let  $A \in \text{Mat}_{r,s}(\mathbb{R})$  be a matrix representing  $V$  as above, and let  $\varepsilon > 0$  be a given threshold number.

First we compute the SVD of  $\mathcal{A}$  and get  $\mathcal{A} = \mathcal{U}\mathcal{S}\mathcal{V}^{\text{tr}}$  as in Theorem 2.1. If we now replace all singular values  $s_i < \varepsilon$  by zeroes, Corollary 2.2 says that the resulting matrix  $\tilde{\mathcal{A}} = \mathcal{U}\tilde{\mathcal{S}}\mathcal{V}^{\text{tr}}$  represents the polynomial vector space  $V_{\text{ap}}$  of smallest rank which is  $\varepsilon$ -close to  $V$ . The vector space  $V_{\text{ap}}$  is called the  $\varepsilon$ -truncation of  $V$ .

**Approximate Leading Terms.** In the Border Basis Algorithm we need a vector space basis  $\mathcal{V}$  of  $V$  with pairwise different leading terms and a vector space basis extension  $\mathcal{V} \cup \mathcal{W}'$  with pairwise different leading terms. We want to find an approximate version of this special kind of basis extension. Given a finite dimensional vector space of empirical polynomials  $V \subset P$  and threshold numbers  $\varepsilon, \tau > 0$ , we first replace  $V$  by  $V_{\text{ap}}$  (using the threshold number  $\varepsilon$ ). Let  $\mathcal{A} = (a_{ij}) \in \text{Mat}_{r,s}(\mathbb{R})$  be a matrix representing  $V_{\text{ap}}$ . By choosing the ONB  $\{f_1, \dots, f_r\}$  of  $V_{\text{ap}}$  provided by the rows of the matrix  $\mathcal{V}$  in the SVD, we may assume that  $r \leq s$  and  $\mathcal{A} = \mathcal{V}_1$  where  $\mathcal{V}_1$  consists of the first  $r$  rows of the orthogonal matrix  $\mathcal{V}$  of size  $s \times s$ .

**Definition 4.3.** Let  $\tau > 0$  be a given threshold number, and let  $\sigma$  be a degree compatible term ordering.

- (1) For a unitary polynomial  $f \in P$ , the maximal term  $t \in \text{Supp}(f)$  whose coefficient has an absolute value  $> \tau$  is called the  $\tau$ -approximate leading term of  $f$  with respect to  $\sigma$  and is denoted by  $\text{LT}_{\sigma,\tau}^{\text{ap}}(f)$ . The coefficient of  $\text{LT}_{\sigma,\tau}^{\text{ap}}(f)$  in  $f$  is called the  $\tau$ -approximate leading coefficient of  $f$  and is denoted by  $\text{LC}_{\sigma,\tau}^{\text{ap}}(f)$ .
- (2) Let  $V \subseteq P$  be a finite dimensional vector subspace. The set

$$\text{LT}_{\sigma,\tau}^{\text{ap}}(V) = \{\text{LT}_{\sigma,\tau}^{\text{ap}}(f) \mid f \in V, \|f\| = 1\}$$

is called the  $\tau$ -approximate leading term set of  $V$ .

If the threshold number  $\tau$  is small enough, every unitary polynomial in  $V$  has an approximate leading term. However, it is easy to see that the set of approximate leading terms of a unitary vector space basis of  $V$  may be strictly smaller than  $\text{LT}_{\sigma,\tau}^{\text{ap}}(V)$ . We would like to find a unitary vector space basis of  $V$  which has a numerically well-behaved approximate leading term set in the sense that its leading terms are exactly the approximate leading terms of the polynomials in  $V$ . For this purpose we may have to modify  $V$  slightly, and we have to choose  $\tau$  small enough. The following example explains why.

**Example 4.4.** Let  $P = \mathbb{R}[x, y]$ , let  $\mathcal{O} = \{1, x, y, xy\}$ , and let the vector space  $V = \langle 0.6xy + 0.8, 0.6x + 0.8, 0.6y + 0.8 \rangle$  be represented by the matrix  $\mathcal{A} =$

$$\begin{pmatrix} 0.6 & 0 & 0 & 0.8 \\ 0 & 0.6 & 0 & 0.8 \\ 0 & 0 & 0.6 & 0.8 \end{pmatrix}$$

with respect to  $\sigma = \text{DegRevLex}$ . Then the polynomial  $f = \tilde{f}/\|\tilde{f}\|$ , where  $\tilde{f} = 0.6xy + 0.6x + 0.6y + 1.8$ , is a unitary polynomial in  $V$ . Since  $\tilde{f} \approx 0.229xy + 0.229x + 0.229y + 0.306$ , the choice of  $\tau = 0.25$  would yield  $\text{LT}_{\sigma,\tau}^{\text{ap}}(f) = 1$  and the given polynomials would not be a well-behaved approximate leading term basis of  $V$ .

The following proposition specifies a measure for the maximal  $\tau$  we can use.



**Proposition 4.5.** *Let  $\mathcal{A} = (a_{ij}) \in \text{Mat}_{m,n}(\mathbb{R})$  be a matrix representing a list of unitary polynomials  $(f_1, \dots, f_r)$  with respect to the list of terms  $\mathcal{O} = (t_1, \dots, t_s)$ . Suppose that  $\mathcal{A}$  is in reduced row echelon form and that its pivot entries are  $\ell_1, \dots, \ell_s$ . If we let  $c = \max_{i,j} \{|a_{ij}|/|\ell_i|\}$  and choose  $\tau < (r + (s-r)r^2c^2)^{-1/2}$ , we have  $\text{LT}_{\sigma,\tau}^{\text{ap}}(V) = \{\text{LT}_{\sigma,\tau}^{\text{ap}}(f_1), \dots, \text{LT}_{\sigma,\tau}^{\text{ap}}(f_r)\}$ .*

*If this condition holds, we shall say that  $\{f_1, \dots, f_r\}$  is a  $\tau$ -approximate leading term basis of  $V$ .*

*Proof.* Let  $f \in V$  be a unitary polynomial. We denote the set of column indices of the pivot elements by  $J$  and choose  $\tilde{f} = d_1t_1 + \dots + d_st_s \in \mathbb{R} \cdot f$  with  $d_j \in \mathbb{R}$  such that  $\max\{|d_j| \mid j \in J\} = 1$ . We write  $\tilde{f} = c_1f_1 + \dots + c_rf_r$  with  $c_i \in \mathbb{R}$ . Then we obtain

$$\tau < (r + (s-r)c^2r^2)^{-1/2} \leq (r + (s-r)c^2(\sum_{i=1}^r |c_i\ell_i|)^2)^{-1/2}$$

since  $|d_j| \leq 1$  for  $j \in J$  and  $c_i\ell_i$  is the coefficient of  $\tilde{f}$  at the term corresponding to the  $i$ -th pivot position. Moreover, for  $i = 1, \dots, r$ , we have  $|c_i\ell_i| \geq |a_{ij}|$  by the definition of  $c$ , and hence

$$\begin{aligned} \tau &< (r + (s-r)(\sum_{i=1}^r |c_i\ell_i|)^2)^{-1/2} \\ &\leq (r + \sum_{j \notin J} (\sum_{i=1}^r |c_ia_{ij}|)^2)^{-1/2} \\ &\leq (\sum_{j=1}^s |d_j^2|)^{-1/2} \leq 1/\|\tilde{f}\| \end{aligned}$$

Consequently, the largest coefficient  $|d_j|/\|\tilde{f}\|$  of  $f$  at one of the approximate leading terms is  $> \tau$ , and  $\text{LT}_{\sigma,\tau}^{\text{ap}}(f)$  is one of the approximate leading terms of the basis.  $\square$

The requirements for the size of  $\tau$  imposed by this proposition are usually quite modest, as the following remark shows.

**Remark 4.6.** In the setting of the preceding proposition, some values of the bound for  $\tau$  are given by the following table:

r	s	c	maximal $\tau$
3	10	100	$1.26 \cdot 10^{-3}$
10	50	100	$1.58 \cdot 10^{-4}$
20	100	100	$5.59 \cdot 10^{-5}$
30	1000	100	$1.07 \cdot 10^{-5}$
30	10000	100	$3.34 \cdot 10^{-6}$

Given an arbitrary matrix representing a vector space of polynomials, we can compute an approximate leading term basis using Lemma 3.2.

**Corollary 4.7** (Approximate Leading Term Bases).

*Let  $\mathcal{A} = (a_{ij}) \in \text{Mat}_{r,s}(\mathbb{R})$  be a matrix representing a vector space of polynomials  $V = V_{\text{ap}}$  with basis  $\{f_1, \dots, f_r\}$ . Recall that the columns of  $\mathcal{A}$  are indexed by the terms in  $\text{Supp}(V)$ . Suppose there exists a degree compatible term ordering  $\sigma$  such that larger terms w.r.t.  $\sigma$  correspond to columns having smaller column indices. We bring  $\mathcal{A}$  into reduced row echelon form using Lemma 3.2. The resulting matrix  $\mathcal{A}'$  represents a vector space of polynomials  $V' \subset P$ . Let  $f'_1, \dots, f'_{r'} \in V'$  be the unitary vector space basis of  $V'$  corresponding to the rows of  $\mathcal{A}'$ . If  $\tau < (r' + (s' - r')(r')^2c^2)^{-1/2}$ , where  $c$  is defined as in the proposition, then every unitary polynomial  $f \in V'$  satisfies*

$$\text{LT}_{\sigma,\tau}^{\text{ap}}(f) \in \text{LT}_{\sigma,\tau}^{\text{ap}}(V') = \{\text{LT}_{\sigma,\tau}^{\text{ap}}(f'_1), \dots, \text{LT}_{\sigma,\tau}^{\text{ap}}(f'_{r'})\}$$

Thus  $\{f'_1, \dots, f'_{r'}\}$  is a  $\tau$ -approximate leading term basis of  $V'$ . Moreover, the vector space  $V'$  is  $(s-r)\tau$ -close to  $V$ .

*Proof.* The first claim follows immediately from the proposition. The second claim follows from the observation that setting a coefficient of absolute value  $< \tau$  to zero in the algorithm of Lemma 3.2 changes the norm of a polynomial by less than  $\tau$ .  $\square$

Notice that the bound for  $\tau$  required in this corollary holds in most cases. The rows of  $\mathcal{A}'$  are unitary polynomials and its pivot entries are larger than  $\tau$  by construction. This gives already a crude bound  $\tau < 1/c$  and shows that the slightly stronger bound of the corollary can only fail if we have a leading coefficient  $\text{LC}_\sigma^{\text{ap}}(f'_j)$  which is just barely larger than the given threshold number. To simplify the following presentation, we shall assume that the bound for  $\tau$  is always satisfied. Of course, in an actual implementation a suitable check should be inserted.

**The Approximate  $V^+$ .** Next we extend the procedure of passing from  $V$  to  $V_{\text{ap}}$  to the following construction. In the Border Basis Algorithm (cf. [11], Prop. 18) the vector space  $V$  is repeatedly replaced by

$$V^+ = V + x_1 V + \dots + x_n V.$$

The approximate version of this construction works as follows.

**Remark 4.8.** [Approximate Leading Term Basis Extension]

Let  $V \subset P$  be a vector space of polynomials, let  $\varepsilon, \tau > 0$  be threshold numbers, and let  $\sigma$  be a degree compatible term ordering.

- (1) Compute the matrix  $\mathcal{A}'$  which represents an approximate leading term basis  $\{f'_1, \dots, f'_{r'}\}$  of  $V'_{\text{ap}}$  as above. (Recall that columns of  $\mathcal{A}'$  having lower column indices correspond to larger terms w.r.t.  $\sigma$ .)
- (2) The representing matrix of  $(V'_{\text{ap}})^+ = V'_{\text{ap}} + x_1 V'_{\text{ap}} + \dots + x_n V'_{\text{ap}}$  is of the form  $\begin{pmatrix} \hat{\mathcal{A}} \\ \mathcal{B} \end{pmatrix}$  where  $\hat{\mathcal{A}}$  is obtained by enlarging  $\mathcal{A}'$  by some zero columns corresponding to new terms in the support of  $(V'_{\text{ap}})^+$ . If necessary, resort the columns of the matrix  $\begin{pmatrix} \hat{\mathcal{A}} \\ \mathcal{B} \end{pmatrix}$  such that columns having lower column indices correspond to larger terms. Using the pivot entries in  $\hat{\mathcal{A}}$  corresponding to the approximate leading terms of the polynomials  $f'_i$ , clean out the corresponding columns of  $\mathcal{B}$  and get a matrix  $\mathcal{B}'$ .
- (3) Delete the rows of  $\mathcal{B}'$  which are  $\tau$ -approximately zero. Compute the  $\varepsilon$ -truncation of the SVD of the resulting matrix and get  $\tilde{\mathcal{B}} = \mathcal{U}' \cdot \mathcal{S}' \cdot (\mathcal{V}')^{\text{tr}}$ .
- (4) Let  $\mathcal{V}'_1$  be the first rows of  $\mathcal{V}'$  which form an ONB of the row space of  $\tilde{\mathcal{B}}$  (see Theorem 2.1.4). Using Corollary 4.7, compute an approximate leading term basis  $\{f'_{r'+1}, \dots, f'_{r'}\}$  of a vector space  $W'$  which is  $\delta$ -close to the vector space  $W$  represented by  $\tilde{\mathcal{B}}$ . (Here  $\delta \leq ((n+1)s - nr)\tau$  can be used as a crude estimate.)

Then the set  $\{f'_1, \dots, f'_{r'}\}$  is an approximate leading term basis of the vector space  $V'_{\text{ap}} \oplus W' \subset P$  which is  $\delta'$ -close to the vector space  $(V'_{\text{ap}})^+$  for  $\delta' = \delta + \varepsilon$ .

**Approximate Stable Spans.** The last ingredient we need is an approximate version of the computation of a stable span explained in [11], Prop. 13. We continue to use two threshold numbers:  $\varepsilon$  for singular value truncations to remove small polynomials and  $\tau$  to prevent impossibly small leading coefficients.

**Proposition 4.9.** *Let  $f_1, \dots, f_r \in P$  be linearly independent unitary polynomials, let  $V = \langle f_1, \dots, f_r \rangle_{\mathbb{R}}$ , let  $s = \# \text{Supp}(V)$ , let  $U = \langle \mathbb{T}_{\leq d}^n \rangle_{\mathbb{R}}$  for some  $d \geq \max\{\deg(f_1), \dots, \deg(f_r)\}$ , let  $\sigma$  be a degree compatible term ordering, and let  $\varepsilon, \tau > 0$  be threshold numbers. We perform the following steps.*

- (1) *Using Corollary 4.7, compute a unitary basis  $\mathcal{V} = \{f'_1, \dots, f'_{r'}\}$  of a vector space  $V'_{\text{ap}}$  which is an approximate leading term basis of  $V'_{\text{ap}}$ .*
- (2) *Using Remark 4.8, compute a unitary basis extension  $\mathcal{W}'$  for  $V'_{\text{ap}} \subseteq (V'_{\text{ap}})^+$  so that the elements of  $\mathcal{V} \cup \mathcal{W}'$  are an approximate leading term basis of a vector space  $\delta + \varepsilon$ -close to  $(V'_{\text{ap}})^+$ .*
- (3) *Let  $\mathcal{W} = \{f'_{r'+1}, \dots, f'_{r'+\varrho}\} = \{p \in \mathcal{W}' \mid \deg(p) \leq d\}$ .*
- (4) *If  $\varrho > 0$  then replace  $\mathcal{V}$  with  $\mathcal{V} \cup \mathcal{W}$ , increase  $r'$  by  $\varrho$ , and go to step 2.*
- (5) *Return  $\mathcal{V}$ .*

*This is an algorithm which returns a set of unitary polynomials  $\mathcal{V} = \{f'_1, \dots, f'_{r'}\}$  which is an approximate leading term basis of  $\tilde{V} := \langle f'_1, \dots, f'_{r'} \rangle_{\mathbb{R}}$ , such that the original polynomials  $f_1, \dots, f_r$  are  $\varepsilon + (s - r)\tau$ -close to  $\tilde{V}$ , and such that  $\tilde{V}$  is approximately  $U$ -stable, i.e. such that we have  $(\tilde{V}'_{\text{ap}})^+ \cap U = \tilde{V}$ .*

*Proof.* The method of the proof of [11], Prop. 13 shows that the result is approximately  $U$ -stable. Let us check that the procedure is finite. This is due to the fact that the basis extension performed in step 2 does not decrease the dimension of the vector space  $\tilde{V}$  generated by  $\mathcal{V}$ . Thus the dimensions of the vector spaces  $\langle \mathcal{V} \rangle_{\mathbb{R}}$  form a non-decreasing sequence and the bound  $r < \dim_{\mathbb{R}}(U)$  implies that the loop terminates.

The claim that the original polynomials are  $\varepsilon + (s - r)\tau$ -close to the computed vector space  $\tilde{V}$  follows from the facts that  $V_{\text{ap}}$  is  $\varepsilon$ -close to  $V$  (see Corollary 2.2) and  $V'_{\text{ap}}$  is  $(s - r)\tau$ -close to  $V_{\text{ap}}$  (see Corollary 4.7). Clearly, extensions of this vector space cannot increase the distances under consideration.  $\square$

**ABBA and IABBA.** Combining the preceding steps, we can formulate an approximate version of the Border Basis Algorithm [11], Prop. 18. Recall that, for a polynomial ideal  $I$ , the order ideal of all terms not in  $\text{LT}_{\sigma}(I)$  is denoted by  $\mathcal{O}_{\sigma}(I)$ .

**Theorem 4.10** (The Approximate Border Basis Algorithm (ABBA)).

*Let  $\{f_1, \dots, f_r\} \subset P = \mathbb{R}[x_1, \dots, x_n]$  be a linearly independent set of  $r \geq n$  unitary polynomials, let  $V = \langle f_1, \dots, f_r \rangle_{\mathbb{R}}$ , let  $s = \# \text{Supp}(V)$ , let  $\sigma$  be a degree-compatible term ordering, and let  $\varepsilon, \tau > 0$ . The following algorithm computes the  $\mathcal{O}_{\sigma}(I)$ -border basis  $\{g_1, \dots, g_{\nu}\}$  of an ideal  $I = \langle g_1, \dots, g_{\nu} \rangle$  such that  $f_1, \dots, f_r$  are  $\delta$ -close to  $I$  for  $\delta = \varepsilon + (s - r)\tau$ .*

- B1** *Let  $d = \max\{\deg(f_i) \mid 1 \leq i \leq r\}$  and  $U = \langle \mathbb{T}_{\leq d}^n \rangle_{\mathbb{R}}$ .*
- B2** *Using Corollary 4.7, compute a unitary basis  $\tilde{\mathcal{V}} = \{f'_1, \dots, f'_{r'}\}$  of a vector space  $V'_{\text{ap}}$  which is an approximate leading term basis of  $V'_{\text{ap}}$ .*
- B3** *Using Remark 4.8, compute a unitary basis extension  $\mathcal{W}'$  for  $V'_{\text{ap}} \subseteq (V'_{\text{ap}})^+$  so that the elements of  $\tilde{\mathcal{V}} \cup \mathcal{W}'$  are an approximate leading term basis of a vector space close to  $(V'_{\text{ap}})^+$ .*
- B4** *Let  $\mathcal{W} = \{f'_{r'+1}, \dots, f'_{r'+\varrho}\} = \{p \in \mathcal{W}' \mid \deg(p) \leq d\}$ .*
- B5** *If  $\varrho > 0$  then replace  $\tilde{\mathcal{V}}$  by  $\tilde{\mathcal{V}} \cup \mathcal{W}$ , increase  $r'$  by  $\varrho$ , and go to **B3**.*
- B6** *Let  $\mathcal{O} = \mathbb{T}_{\leq d}^n \setminus \{\text{LT}_{\sigma}^{\text{ap}}(f'_1) \dots \text{LT}_{\sigma}^{\text{ap}}(f'_{r'})\}$ .*
- B7** *If  $\partial \mathcal{O} \not\subseteq U$  then increase  $d$  by one, update  $U := \langle \mathbb{T}_{\leq d}^n \rangle_{\mathbb{R}}$ , and go to **B3**.*

**B8** Apply the Final Reduction Algorithm (cf. [11], Prop. 17) and return its result  $(g_1, \dots, g_\nu)$ .

*Proof.* *Mutatis mutandis*, it suffices to follow the proof of Prop. 18 in [11] and to add the following observation: The set  $\mathcal{O}$  of terms computed in step **B6** is indeed an order ideal.

Suppose a term  $t$  occurs as a leading term in the basis  $\{f'_1, \dots, f'_{r'}\}$  but only with small coefficients, i.e. not as an approximate leading term. Then this term will be put into  $\mathcal{O}$  by step **B6**. Suppose that a divisor  $t'$  of this term is of the form  $t' = \text{LT}_{\sigma, \tau}^{\text{ap}}(f'_i)$  for some  $i$ . There exists a polynomial  $f'_i$  whose coefficient at  $t'$  is larger than  $\tau$ . If we multiply  $f'_i$  by the appropriate product of indeterminates, the coefficient of  $t$  in the resulting polynomial is larger than  $\tau$ . Thus, after several extensions of  $V$ , the term  $t$  has to be the  $\tau$ -approximate leading term of some polynomial  $f'_j$ , in contradiction to our assumption.

The claim that the original polynomials are  $\delta$ -close to the computed ideal  $I$  was shown in Proposition 4.9.  $\square$

Notice that steps **B2**–**B5** are in essence the algorithm of the preceding proposition. As mentioned in [11], this algorithm can be optimized substantially. In particular, the proof of [11], Prop. 21 shows that we can reduce the size of the space  $U$  (the “computational universe”) dramatically. We get the following improved algorithm.

**Corollary 4.11** (Improved Approximate Border Basis Algorithm (IABBA)).

*In the setting of the theorem, the following algorithm computes the  $\mathcal{O}_\sigma(I)$ -border basis  $\{g_1, \dots, g_\nu\}$  of an ideal  $I = \langle g_1, \dots, g_\nu \rangle$  such that  $f_1, \dots, f_r$  are  $\delta$ -close to  $I$ .*

- C1** Let  $\mathcal{L}$  be the order ideal spanned by  $\bigcup_{i=1}^r \text{Supp}(f_i)$ .
- C2** Using Corollary 4.7, compute a unitary basis  $\mathcal{V} = \{f'_1, \dots, f'_{r'}\}$  of a vector space  $V'_{\text{ap}}$  which is an approximate leading term basis of  $V'_{\text{ap}}$ .
- C3** Using Remark 4.8, compute a unitary basis extension  $\mathcal{W}'$  for  $V'_{\text{ap}} \subseteq (V'_{\text{ap}})^+$  so that the elements of  $\mathcal{V} \cup \mathcal{W}'$  are an approximate leading term basis of a vector space close to  $(V'_{\text{ap}})^+$ .
- C4** Let  $\mathcal{W} = \{f \in \mathcal{W}' \mid \text{LT}_\sigma(f) \in \mathcal{L}\}$ .
- C5** If  $\bigcup_{f \in \mathcal{W}} \text{Supp}(f) \not\subseteq \mathcal{L}$  then replace  $\mathcal{L}$  by the order ideal spanned by  $\mathcal{L}$  and  $\bigcup_{f \in \mathcal{W}} \text{Supp}(f)$ . Continue with step **C4**.
- C6** If  $\mathcal{W} \neq \emptyset$  then replace  $\mathcal{V}$  by  $\mathcal{V} \cup \mathcal{W}$ . Continue with step **C3**.
- C7** Let  $\mathcal{O} = \mathcal{L} \setminus \{\text{LT}_\sigma(v) \mid v \in \mathcal{V}\}$ .
- C8** If  $\partial \mathcal{O} \not\subseteq \mathcal{L}$  then replace  $\mathcal{L}$  by the order ideal  $\mathcal{L}^+ = \mathcal{L} \cup \bigcup_{i=1}^n x_i \mathcal{L}$  and continue with step **C3**.
- C9** Apply the Final Reduction Algorithm and return the polynomials  $g_1, \dots, g_\nu$  computed by it.

To end this section, we apply ABBA to a concrete example.

**Example 4.12.** Consider the (approximately) unitary polynomials

$$\begin{aligned}
 f_1 &= 0.13z^2 + 0.39y - 0.911z \\
 f_2 &= 0.242yz - 0.97y \\
 f_3 &= 0.243xz - 0.97y \\
 f_4 &= 0.242y^2 - 0.97y \\
 f_5 &= 0.243xy - 0.97y \\
 f_6 &= 0.035x^5 - 0.284x^4 + 0.497x^3 + 0.284x^2 - 0.532x + 0.533y
 \end{aligned}$$

We apply ABBA with  $\varepsilon = \tau = 0.001$  and follow the steps. The first basis extension yields 18 polynomials in step **B3**, 15 of which are found to be in the computational universe in step **B4**. They are  $f_1, \dots, f_6$  and

$$\begin{aligned}
 f_7 &= 0.017z^3 + 0.558y - 0.830z \\
 f_8 &= 0.064yz^2 + 0.001z^3 - 0.996y - 0.05z \\
 f_9 &= 0.707y^2z - 0.707yz^2 + 0.002y - 0.0005z \\
 &\vdots \\
 f_{15} &= 0.707x^2y - 0.707x^2z
 \end{aligned}$$

Since we found 9 new polynomials, step **B5** forces us to repeat the basis extension. The second time around we find 32 polynomials in the extended basis, 29 of which are in the universe. The third iteration yields a basis consisting of 52 polynomials, 49 of which are in the universe, and the fourth iteration yields 77 polynomials in the basis and 49 polynomials in the universe.

At this point steps **B6** and **B7** are executed and show that the iteration is finished. It remains to apply the Final Reduction Algorithm. The computed order ideal is  $\mathcal{O} = \{1, x, x^2, x^3, x^4, y, z\}$ , its border is

$$\partial\mathcal{O} = \{x^5, x^4y, x^4z, x^3y, x^3z, x^2y, x^2z, xy, y^2, xz, yz, z^2\},$$

and the resulting border basis consists of  $f_1, \dots, f_6$  together with

$$\begin{aligned}
 g_1 &= 0.062x^2y - 0.998y \\
 g_2 &= 0.062x^2z - 0.998y \\
 g_3 &= 0.016x^3y - 0.9999y \\
 g_4 &= 0.016x^3z - 0.9999y \\
 g_5 &= 0.004x^4y - 0.99999y \\
 g_6 &= 0.004x^4z - 0.99999y
 \end{aligned}$$

This result is in good numerical agreement with the exact result in [11], Ex. 20.

## 5. APPROXIMATE MEMBERSHIP FOR ZERO-DIMENSIONAL IDEALS

Given a polynomial ideal  $I = \langle f_1, \dots, f_s \rangle \subseteq P$  and a polynomial  $g \in P$ , the classical ideal membership problem asks whether we have  $g \in I$ . If this is the case, explicit membership is the search for a concrete representation  $g = h_1f_1 + \dots + h_sf_s$  with  $h_1, \dots, h_s \in P$ . The standard way to solve the decision problem is to choose a term ordering  $\sigma$ , compute a  $\sigma$ -Gröbner basis  $G = \{g_1, \dots, g_t\}$  of  $I$  and use the division algorithm [12], 1.6.4, to decide whether  $\text{NF}_{\sigma, G}(g) = 0$ . The standard method for solving the explicit membership problem consists of invoking

the extended Buchberger algorithm (cf. [12], 2.5.11), computing the syzygy module of the Gröbner basis and transforming the syzygies (cf. [12], 3.1.8 and 3.1.9).

In the approximate setting, these methods fail for several reasons:

- (1) The polynomial  $g$  could be “almost” contained in  $I$ , i.e. the normal form  $\text{NF}_{\sigma,G}(g)$  could be “close to” zero.
- (2) The computations of the Gröbner bases involved are numerically unstable and should be replaced by appropriate approximate algorithms.
- (3) Solutions to the approximate explicit membership problem are highly non-unique: every generator  $f_i$  can be modified by a “small” polynomial, there exist syzygies of “small” polynomials, and the set of “small” polynomials has apparently no usable algebraic structure.

Nevertheless, in many industrial applications there are strong incentives to seek explicit representations  $g = h_1 f_1 + \dots + h_s f_s$ . Since there are infinitely many such representations, which one is the one realized by the physical system? Is there an “approximate normal form” which enables us to find a candidate for the “simplest” (and hence a candidate for the “true”) representation?

In this section we examine these questions in the case of zero-dimensional polynomial ideals. For the decision problem for zero-dimensional vanishing ideals, the solution is simple: just check whether the evaluation vector of  $g$  is “small”. Now let us tackle the explicit membership problem.

Given an empirical zero-dimensional polynomial ideal  $I$ , compute an order ideal  $\mathcal{O} = \{t_1, \dots, t_\mu\}$  and an  $\mathcal{O}$ -border basis  $G$  of  $I$ . (If  $I$  is defined as the vanishing ideal of an approximate set of points, use Theorem 3.3. If  $I$  is given by an approximate system of generators, use Theorem 4.10.)

Suppose that the order ideal  $\mathcal{O}$  is of the form  $\mathcal{O}_\sigma(I) = \mathbb{T}^n \setminus \text{LT}_\sigma(I)$  for some degree compatible term ordering  $\sigma$ . Then  $G$  is automatically a  $\sigma$ -Gröbner basis, and henceforth a Macaulay basis of  $I$  (cf. [13], 6.4.18 and 4.2.15). If the order ideal  $\mathcal{O}$  is not of this form, we can still use Corollary 3.12 or [17], Section 4. In either case, we assume that we now have a Macaulay basis  $H = \{h_1, \dots, h_\lambda\}$  of  $I$ . Our next step is to pass to a completely reduced orthogonal Macaulay basis. This “Macaulay bases analogue” of a reduced Gröbner basis is defined as follows.

**Definition 5.1.** Let  $H = \{h_1, \dots, h_\lambda\}$  be a Macaulay basis of  $I$ .

- (1) A polynomial  $f \in P$  is called *completely reduced* with respect to  $H$  if in the canonical representation  $f = g_1 h_1 + \dots + g_\lambda h_\lambda + \text{NF}_I(f)$  (cf. [17], 6.2 or [18], 4.3) we have  $f = \text{NF}_I(f)$ .
- (2) The Macaulay basis  $H$  is called a *completely reduced, orthogonal Macaulay basis* of  $I$  if all  $h_i - \text{DF}(h_i)$  are completely reduced w.r.t.  $H$  and

$$\langle \text{DF}(h_i), t \text{DF}(h_j) \rangle = 0 \quad \text{for } i \neq j \quad \text{and } t \in P_{\leq \deg(h_i) - \deg(h_j)}$$

Here  $\text{DF}(f)$  denotes the *degree form* of a polynomial  $f$  (see [13], 4.2.8).

Given  $H$ , a completely reduced, orthogonal Macaulay basis of  $I$  can be computed easily (cf. [17], 6.4). Moreover, it is essentially unique (cf. [17], 6.5). Therefore we shall from now on assume that  $H$  has this property.

**Remark 5.2.** [Approximate Membership Using Macaulay Bases]

Let  $H = \{h_1, \dots, h_\lambda\}$  be a completely reduced, orthogonal Macaulay basis of  $I$ . For any polynomial  $f \in P$  of degree  $d$ , we use the Macaulay Division Algorithm

(cf. [18], 3.1 and [17], 6.2), to find a representation

$$f = g_1 h_1 + \cdots + g_\lambda h_\lambda + r_0 + \cdots + r_d$$

with  $g_i \in P$  satisfying  $\deg(g_i) \leq \deg(f) - \deg(h_i)$  and such that  $r_j \in P_j$  is homogeneous of degree  $j$ . The polynomial  $\text{COP}_I(f) = r_0 + \cdots + r_d$  is called the *canonical orthogonal projection* of  $f$  w.r.t.  $I$ .

If we set  $f \approx 0 \Leftrightarrow \|\text{COP}_I(f)\| < \varepsilon$  for a given threshold value  $\varepsilon > 0$ , we can solve the *approximate membership decision problem* for zero-dimensional ideals, presumably in a numerically stable way. The reason for the believed stability of this procedure is that border bases (and hence completely reduced, orthogonal Macaulay bases) of a zero-dimensional ideal tend to vary very little if we change the ideal slightly (cf. [20] and [14]).

Of course, if the degree of  $f$  is high, the accuracy of the canonical orthogonal projection depends on the number of reduction steps involved in the Macaulay division. A precise error estimate should take this into account.

To solve the explicit membership problem for empirical zero-dimensional ideals, we have to be even more careful: the representations obtained in the preceding remark can easily be modified by “almost” syzygies of  $(h_1, \dots, h_\lambda)$ . To get this ambiguity under control, we proceed as follows.

**Remark 5.3.** Starting from an order ideal  $\mathcal{O}_\sigma(I)$  and the  $\mathcal{O}_\sigma(I)$ -border basis  $G = \{g_1, \dots, g_\nu\}$  of  $I$ , we compute the completely reduced, orthogonal Macaulay basis  $H = \{h_1, \dots, h_\lambda\}$  of  $I$ . Since the residue classes of the terms in  $\mathcal{O}_\sigma(I) = \{t_1, \dots, t_\mu\}$  form a vector space basis of  $P/I$ , we may then calculate a set of polynomials  $\mathcal{P} = \{p_1, \dots, p_\mu\}$  such that  $\mathcal{P}_{\leq d}$  is an ONB of the orthogonal complement of the vector subspace  $I_{\leq d}$  in  $P_{\leq d}$  for every  $d \geq 0$ . Note that this condition is well-defined, since  $I_{\leq d+1} \cap \langle \mathcal{P}_{\leq d} \rangle = \{0\}$  implies that one ONB is contained in the next. (If  $I$  is the vanishing ideal of an approximate set of points, we can use Corollary 3.12 to get  $\mathcal{P}$ . If  $I$  is given by an approximate set of generators, we can use Theorem 4.10 and apply the Gram-Schmidt orthogonalization procedure to the set  $\mathcal{O}_{\leq d}$  to get  $\mathcal{P}_{\leq d}$ .)

Next we modify the Macaulay Division Algorithm (cf. [18], 3.1 and [17], 6.2) as follows: if we want to project an element  $f \in P_{\leq d}$ , we take its degree form  $\text{DF}(f)$  and project it onto  $\langle t \text{DF}(h_i) \mid t \in \mathbb{T}^n, \deg(th_i) = d \rangle_{\mathbb{R}}$ . The result is the degree  $d$  part of the canonical orthogonal projection and can be expressed as a linear combination of the elements of  $\mathcal{P}_d$ .

This yields an algorithm which computes a representation

$$f = g_1 h_1 + \cdots + g_\lambda h_\lambda + c_1 p_1 + \cdots + c_\mu p_\mu$$

with  $c_i \in \mathbb{R}$  and  $g_j \in P$ . Then  $\text{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$  of  $f$  is the unique representation of  $\text{COP}_I(f)$  in terms of the ONB  $\mathcal{P}$ .

Using an ONB  $\mathcal{P}$  of the complement of  $I$  in  $P$ , we can finally define approximate normal forms and solve the approximate explicit membership problem.

**Definition 5.4.** Let  $f \in P$ , let  $\mathcal{P} = \{p_1, \dots, p_\mu\}$  be an ONB of the orthogonal complement of  $I$  in  $P$ , and let  $\varepsilon > 0$ . We write  $\text{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$  with  $c_i \in \mathbb{R}$ . Then the polynomial  $\text{NF}_{\mathcal{P}, I}^{\text{ap}}(f) = \sum_{\{i: |c_i| \geq \varepsilon\}} c_i p_i$  is called the *approximate normal form* of  $f$  with respect to  $\mathcal{P}$  and  $I$ .

Now the preceding discussion can be summarized as follows.

**Proposition 5.5** (Approximate Explicit Membership for Zero-Dimensional Ideals).

Let  $I \subset P$  be a zero-dimensional ideal, let  $H$  be a completely reduced, orthogonal Macaulay basis of  $I$ , let  $\mathcal{P} = \{p_1, \dots, p_\mu\}$  be an ONB of the orthogonal complement of  $I$  in  $P$ , let  $f \in P$ , and let  $\varepsilon > 0$ .

- (1) The polynomial  $f$  is “almost” contained in  $I$  if and only if  $\text{NF}_{\mathcal{P}, I}^{\text{ap}}(f) = 0$ . More precisely, if  $\text{NF}_{\mathcal{P}, I}^{\text{ap}} = 0$  then  $f$  is  $\varepsilon\sqrt{\mu}$ -close to  $I$ , and if  $f$  is  $\varepsilon$ -close to  $I$  then  $\text{NF}_{\mathcal{P}, I}^{\text{ap}}(f) = 0$ .
- (2) If  $f$  is  $\varepsilon$ -close to  $I$ , we use the Macaulay Division Algorithm to compute a representation

$$f = g_1 h_1 + \dots + g_\lambda h_\lambda + c_1 p_1 + \dots + c_\mu p_\mu$$

with  $g_i \in P$  of degree  $\deg(g_i) \leq \deg(f) - \deg(h_i)$  and  $|c_i| < \varepsilon$ . Then the relation  $f \approx g_1 h_1 + \dots + g_\lambda h_\lambda$  is called an approximate explicit representation of  $f$ . If another polynomial  $f' \in P$  which is also  $\varepsilon$ -close to  $I$  has the same approximate explicit representation, then we have  $\text{NF}_{\mathcal{P}, I}^{\text{ap}}(f - f') = 0$ .

*Proof.* To show the first claim, we start by assuming that we have  $\text{COP}_I(f) = c_1 p_1 + \dots + c_\mu p_\mu$  with  $|c_i| < \varepsilon$ . Since  $\mathcal{P}$  is an ONB of the orthogonal complement of  $I$  in  $P$ , the length of the perpendicular from  $f$  to  $I$  is therefore  $\sqrt{c_1^2 + \dots + c_\mu^2} \leq \varepsilon\sqrt{\mu}$ . Conversely, if the canonical projection  $\text{COP}_I(f) = c_1 p_1 + \dots + c_\mu p_\mu$  satisfies  $\sqrt{c_1^2 + \dots + c_\mu^2} \leq \varepsilon$ , then we have  $|c_i| < \varepsilon$  for  $i = 1, \dots, \mu$ .

Now we prove the second claim. Let  $f = g_1 h_1 + \dots + g_\lambda h_\lambda + c_1 p_1 + \dots + c_\mu p_\mu$  and  $f' = g'_1 h_1 + \dots + g'_\lambda h_\lambda + c'_1 p_1 + \dots + c'_\mu p_\mu$  be the representations of  $f$  and  $f'$  provided by the Macaulay Division Algorithm. Since the approximate explicit representation of  $f$  and  $f'$  are equal, we have  $g_i = g'_i$  for  $i = 1, \dots, \lambda$ . The hypotheses that  $f$  and  $f'$  are  $\varepsilon$ -close to  $I$  yield  $\sqrt{c_1^2 + \dots + c_\mu^2} \leq \varepsilon$  and  $\sqrt{(c'_1)^2 + \dots + (c'_\mu)^2} \leq \varepsilon$ . Hence the norm of  $f - f' = (c_1 - c'_1)p_1 + \dots + (c_\mu - c'_\mu)p_\mu$  is at most  $2\varepsilon$ , and the approximate normal form of  $f - f'$  (with respect to the threshold number  $2\varepsilon$ ) is zero.  $\square$

## 6. COMPUTATIONAL RESULTS

In this section we provide some timings for the calculation of approximate vanishing ideals and approximate border bases using our algorithms. Moreover, we show the effects of a proper data scaling on the quality of the calculated approximate border or Gröbner bases. The following timings are based on an implementation of the algorithms of Section 3 in the ApCoCoA library (see [3]). They were produced using a notebook having an AMD Athlon processor, 1 GB Ram, and running at 2.2 GHz.

**Calculating Approximate Vanishing Ideals.** The first computational test is based on a real-world data set coming from an application in oil industry (see Section 7). This data set contains 2445 points  $p_i = (p_{i1}, \dots, p_{i9})$  in  $[-1, 1]^9 \subset \mathbb{R}^9$ . Since these data have large inherent measurement errors, for meaningful numbers  $\varepsilon$  the resulting order ideals have a much smaller number of elements than 2445. The following table shows the results for the computation of an approximate border basis of an approximate vanishing ideal using Algorithm 3.3 with  $\tau$  chosen according to the machine precision, with  $\sigma = \text{DegLex}$ , and with the stated values of  $\varepsilon$ .



$\varepsilon$	# BB	deg	# $\mathcal{O}$	max error	avg error	time
1	126	$\leq 4$	20	0.0868	0.000289	0.44 s
0.5	175	$\leq 4$	29	0.1044	0.000238	1.06 s
0.1	301	$\leq 5$	54	0.0259	0.000136	3.5 s
0.01	543	$\leq 6$	107	0.0006	0.000023	13.4 s
$10^{-5}$	1815	$\leq 7$	484	$< 10^{-6}$	$< 10^{-6}$	441 s

TABLE 1. Calculating Approximate Border Bases for 2445 Points in  $\mathbb{R}^9$ 

The numbers in the **max error** column are the maximal absolute values of the evaluations of the border basis polynomials at the given points, and the numbers in the **avg error** column are the averages of these absolute values. In view of the fact that the initial data typically contain errors of about 10% of the values, it is not advisable to use a very large or very small  $\varepsilon$ . The numbers for  $\varepsilon = 10^{-5}$  correspond in essence to the exact case  $\varepsilon = 0$ . Even in that case the algorithm produces an order ideal of less than 2445 terms because of the finite machine precision.

Table 2 shows the timings and numerical quality of a calculation using the same data set, but the variant of Remark 3.8 to compute an approximate Gröbner basis for an approximate vanishing ideal. In the case at hand, the computed polynomials are indeed an approximate Gröbner basis and no numerical instability surfaces.

$\varepsilon$	#GB	deg	# $\mathcal{O}$	max error	avg error	time
1	31	$\leq 3$	16	0.0385	0.000423	0.08 s
0.5	35	$\leq 4$	22	0.0385	0.000402	0.12 s
0.1	66	$\leq 5$	42	0.0259	0.000157	0.5 s
0.01	112	$\leq 5$	78	0.0006	0.000055	1.7 s
$10^{-5}$	531	$\leq 6$	366	$< 10^{-6}$	$< 10^{-6}$	78.1 s

TABLE 2. Calculating a DegLex-Gröbner Basis for 2445 Points in  $\mathbb{R}^9$ 

Our second computational test (see Table 3) is the calculation of an approximate border basis of an  $\varepsilon$ -approximate vanishing ideal of a set of 10105 points in  $\mathbb{R}^9$ . Note that this is also a real-world data set which corresponds to an application in steel industry described in Section 7.

$\varepsilon$	#BB	deg	# $\mathcal{O}$	max error	avg error	time
1	230	$\leq 4$	46	0.4089	0.000205	3.9 s
0.5	373	$\leq 5$	78	0.1898	0.000137	10.5 s
0.1	759	$\leq 6$	188	0.0579	0.000036	69.5 s
0.01	1765	$\leq 6$	524	$< 0.01$	$< 0.00001$	685 s

TABLE 3. Calculating Approximate Border Bases for 10105 Points in  $\mathbb{R}^9$ 

As before, also in this case the Gröbner basis version of the AVI-Algorithm works just fine:

$\varepsilon$	#GB	deg	# $\mathcal{O}$	max error	avg error	time
1	90	$\leq 4$	38	0.4089	0.000714	4.36 s
0.5	126	$\leq 4$	59	0.1898	0.000327	9.70 s
0.1	263	$\leq 5$	145	0.0579	0.000084	61.7 s
0.01	659	$\leq 6$	395	$< 0.01$	$< 0.00001$	705 s

TABLE 4. Calculating DegLex-Gröbner Bases for 10105 Points in  $\mathbb{R}^9$ 

**Applying the Approximate Border Basis Algorithm.** To test the Approximate Border Basis Algorithm (ABBA), we computed the approximate vanishing ideal of certain sets of approximate points and then applied ABBA to find a border basis for this approximate zero-dimensional ideal. The following timings were obtained with a top-level implementation of ABBA in CoCoAL using the ApCoCoA library (cf. [3]).

The first example is an ideal in  $\mathbb{R}[x, y, z]$  generated by the polynomial

$$\begin{aligned}
f_1 = & 0.041805x^3 + 0.017262x^2y + 0.016641x^2z + 0.020066xy^2 + 0.000575xyz \\
& + 0.020825xz^2 + 0.007537y^3 + 0.007845y^2z + 0.007695yz^2 + 0.007928z^3 \\
& - 0.22902x^2 - 0.12885xy - 0.13055xz - 0.092272y^2 - 0.067469yz \\
& - 0.095283z^2 + 0.51837x + 0.33792y + 0.34477z
\end{aligned}$$

and further similar 12 polynomials of degrees 4,4,4,4,5,5,5,5,5,6,6,6. The threshold number used for pivoting in Corollary 4.7 and Remark 4.8 is  $\tau = 10^{-5}$ . The number  $\varepsilon$  in the following table is the cut-off value for the SVD which has to be computed in step 3 of Remark 4.8.

	$\varepsilon$	#BB	deg	time
1	0.1	81	$\leq 6$	2.81 s
2	0.01	83	$\leq 6$	1.44 s
3	0.001	84	$\leq 6$	0.70 s
4	0.0001	84	$\leq 6$	0.70 s
5	0.00001	84	$\leq 6$	0.72 s
6	0.000001	84	$\leq 6$	0.71 s

TABLE 5. Calculating a Border Basis of 84 Polynomials with ABBA

Note that the execution time initially decreases with decreasing  $\varepsilon$ . This is due to the fact that the algorithm decides earlier to enlarge the computational universe.

In our second example we use an ideal generated by 29 polynomials of degrees 6,7,8, and 9. The threshold number is  $\tau = 10^{-10}$ .

**Data Scaling.** The example calculations we have performed indicate strongly that data scaling is an important factor for the numerical quality of the results of our algorithms. From the algebraic point of view, scaling does not affect the problem, but from the numerical point of view scaling provides additional *stability* for the solution.

To show this effect, we use a real-world data set consisting of 2541 points in  $\mathbb{R}^7$ . For both computations, we truncate singular values less than 0.0001. The scaled

	$\varepsilon$	#BB	deg	time
1	0.1	220	$\leq 9$	59.60 s
2	0.01	173	$\leq 9$	27.70 s
3	0.001	216	$\leq 9$	11.95 s
4	0.0001	220	$\leq 9$	5.34 s
5	0.00001	220	$\leq 9$	4.94 s
6	0.000001	220	$\leq 9$	5.17 s

TABLE 6. Calculating a Border Basis of 220 Polynomials with ABBA

version is calculated in approx. 2 seconds, while the unscaled version takes approx. 4 seconds. The following figure visualizes the effect of numerical instabilities. The left picture shows the mean length of the evaluation vectors of the computed Gröbner basis polynomials without scaling. The right picture shows the same measure of numerical quality for a computation which used data scaling.

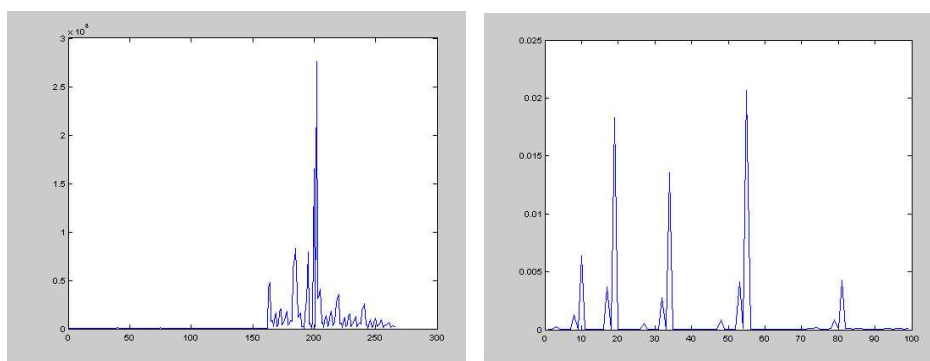


FIGURE 3. Mean Evaluation Errors Without and With Data Scaling

From an algebraic point of view, the algorithm produces a “correct” result in both cases. However, numerical inaccuracies and high degrees of the computed polynomials lead to undesirable results if we do not use data scaling. We consider the mean length of the evaluation vectors of a Gröbner basis polynomial at the given points. In the left figure the maximal mean evaluation error is  $\approx 2.8 \cdot 10^8$ , i.e. these polynomials are completely useless. If we use data scaling, we get a maximal mean evaluation error of less than 0.025 in the right figure. Moreover, in the scaled example we have about 100 elements in the Gröbner basis, compared to more than 280 polynomials in the unscaled version.

Not surprisingly, this example (and many similar ones we tried) suggests that data scaling provides several benefits: shorter running times, smaller output size, and a dramatically improved numerical quality of the result.

## 7. APPLICATIONS

Our goal in this final section is to describe some practical problems which originated our interest in the topic at hand and to explain how our techniques can be

applied to those practical problems. Let us begin by explaining the “meta” application which motivates much of our continuing research. After that we look at two concrete applications.

**Searching for Relations in Measured Data.** Assume that we are given a set of measurements, for example several time series. We want to find algebraic relations in these data, i.e. polynomial formulas which are almost fulfilled at every sampling point. These formulas can then be used to determine a minimal independent set of inputs or to indulge in other forms of data-mining. Polynomial relations which are exactly satisfied at each data point can be found using the classical Buchberger-Möller algorithm (see [2] and [15]). However, time series coming from *real measurements* tend to contain measurement errors. In this case exact interpolation yields misleading results.

Instead, we should be looking for polynomial relations which are *almost* satisfied by the time series. Thus the AVI-Algorithm 3.3 provides us with the kind of relations we are searching for. In fact, we get a border basis that also provides additional stability with respect to variations of the input data (cf. [20]).

Nevertheless, there is still one problem which has to be addressed: what happens if the “true” approximate relations which exist in the data sets are not of a polynomial nature? With our algorithms, we can only find polynomials of low degree passing almost through the points. To pass this hurdle, it may be worthwhile to examine the (partial) differential equations or other laws governing the physical system in which the data were measured. If the variable corresponding to some time series appears within a non-polynomial function (e.g. within a logarithm or a square root), we should perform the inverse transformation on the time series, so that we have a fair chance that the variable underlying the new time series appears in a polynomial relation. Of course, this approach requires a thorough understanding of the physical system. In any case, the quality of the polynomial relations we find using our algorithms can be easily judged from the degree and the size of the resulting polynomials and by comparing the remaining evaluation errors to the estimated size of the measurement errors.

**Oil Industry.** Certain problems one encounters in oil and gas production were the motivation to start this investigation. We will present here briefly a couple of these motivating problems. An in-depth description of these applications has been provided in a separate paper (cf. [8]).

A common situation in production operations in oil industry is that a number of wells produce into a large piece of tubing called the *bulk header*, and from this header the common production flows to the *bulk separator*, where the different phases, namely oil, water and gas, are separated and the production rates of the separated phases are measured. The productions from the individual wells are obtained through a *well test*. A well test is an experiment where the well is decoupled from the bulk header and connected to the *test header* which in turn is connected to the *test separator*. Here again the phase productions are measured, but this time those of the well-on-test only. The phase productions from the well-on-test are recombined downstream from the test separator, and added to the production from the other wells, and this common production is processed by the bulk separator. Apart from the phase productions also quantities like pressures, temperatures, and

injected “lift-gas” are measured. These additional measured quantities are examples of what we have called *inputs* in Section 1, whereas the phase productions are examples of what we have called *outputs* there. Fitting the inputs measured during the test to the measured oil production, assuming a polynomial model and using standard least squares techniques (specifically the techniques described in [4]), gives *some* production polynomial for the well-on-test.

Here we have our first encounter with polynomials having small evaluations over the *test set*  $\mathbb{X}$ : *different* production polynomials (in terms of total degree and support), for the *same* well may result in equivalent goodness of fit results in the validation experiments (see Section 1). Pairwise differences between these production polynomials yield polynomials having small evaluations. Relations among the input variables are causing this ambiguity. Hence the calculation of an approximate vanishing ideal from Section 3 of the test set points helps us establish these relations among the input variables. Having computed an approximation for the ideal of relations  $I$ , the construction of the production polynomial can be repeated, but this time by fitting against the  $\mathbb{R}$ -vector space basis  $\mathcal{O}$  of  $P/I$ , where  $P$  is the polynomial ring associated with the well test experiment, or with respect to the orthonormal vector space basis of a complement of  $I$  in  $P$  from Corollary 3.12.

Alternatively, an algebraic equation for the well production may be established directly from an approximate vanishing ideal calculation by adding the points associated with the measured production to the set  $\mathbb{X}$  for this calculation. This may lead to an implicit equation in the well production. We obtain simplifications in this connection by using physical knowledge about the problem at hand. More precisely, we may construct new indeterminates from our original ones on the basis of this physical knowledge. To be specific, an example of such a constructed indeterminate could be  $x = \sqrt{(x_i - x_j)x_j}$ , where  $x_i$  and  $x_j$  are original indeterminates related to pressures and where  $x_i - x_j$  is related to the pressure drop over a *restriction*. Here restriction should be interpreted in a broad sense like a valve, a piece of tubing, or the inflow opening from the reservoir to the production tubing. Then the quantity  $x$  is associated with the driving force over this restriction. But most of all we should realize that our term ordering, and specifically the ordering of the indeterminates, is here not just “any” abstract ordering. A particular physical meaning is associated with the indeterminates and the terms. A judicious choice of the term ordering that is also based on physical knowledge about the problem provides further possibilities to investigate this important problem of determining the relations among the variables.

Next we explain how our results from Sections 4 and 5 can be applied in this setting. Let us assume that we have production polynomials  $f_1, \dots, f_s$  available which describe the wells when they are producing *in isolation*, that is purely as a separate production system. This is a strong assumption. Our justification for it is that it serves our present motivation purposes well. Moreover, assume that we also have the total production polynomial  $f$  available. If none of the wells is producing, there is also no total production. And since no production implies a zero of the concerning production polynomial, it follows that the total production polynomial vanishes on the set of common zeros of the well production polynomials. By Hilbert’s Nullstellensatz, the total production polynomial is a member of the radical of the ideal generated by the well production polynomials. Furthermore, since the *empirical* well production polynomials are generic, they generate a radical

ideal. The results of Section 4 enable us to compute the ideal generated by the empirical production polynomials, whereas the results of Section 5 enable us to solve the approximate membership problem for the total production polynomial for this ideal. This means that we get an explicit representation  $f = h_1 f_1 + \dots + h_s f_s$  for the total production in terms of the separate well productions. The polynomials  $h_i$  express the *interactions* in the well production system. Now the total volume of oil that can be produced from an oil reservoir is called the *ultimate recovery*. The current state of the art allows an ultimate recovery of at most 30%. The main reason for this figure being so low is the fact that the above indicated interactions are unknown. This is partly induced by the technological set-up described above. Thus our results allow a direct attack on the ultimate recovery problem, which is to date the most challenging problem in oil and gas production operations.

**Steel Industry.** Finally, we briefly describe a problem arising in steel industry which can be tackled using the above techniques. Given a process chain, it is of utmost importance that the melted metal has a guaranteed minimum temperature at the corresponding stages in the process chain. After the metal is melted the annealing process lowers the temperature. This annealing depends strongly on the ingredients initially put into the melting process. Some of these ingredients are cheap, others are quite expensive. Some of them slow the annealing process down, others speed it up. Moreover, there are non-trivial interactions between the ingredients. The aim is to determine a good mixture of ingredients to control the annealing behavior.

This problem could be seen a classical optimization problem. Unfortunately, no good model describing the annealing process is known. As such a model would also have to account for the physical environment (e.g. the surrounding air temperature, humidity, diameter of the container, ...), it would necessarily be quite complicated.

Instead, the methods we described make it possible to search for a model describing the annealing process for a specific setting. Classical approaches predict the annealing behavior only up to an error between 20% and 30%. Consequently, to ensure the necessary temperature at every stage in the process chain, the steel often has to be heated much more than necessary, driving up the cost of production substantially. Using Section 3, we get a much more accurate prediction with an overall error between 7% and 12% in the tested example. This enables us to determine a better mixture for the actual melting process and to simultaneously lower the initial temperatures.

**Acknowledgements.** Part of this work was conducted during the Special Semester on Gröbner Bases, February 1 to July 31, 2006, organized by the RICAM Institute (Austrian Academy of Sciences) and the RISC Institute (Johannes Kepler University) in Linz, Austria. The authors thank these institutions, and in particular Bruno Buchberger, for the warm hospitality and the excellent work atmosphere they experienced in Linz. We thank Robert Corless and Lorenzo Robbiano for helpful discussions on the subjects of this paper, as well as the referees for numerous insightful comments. Important examples that helped us improve our algorithms were found by John Abbott, Claudia Fassino and Mihaela Popoviciu. Finally, we gratefully acknowledge the support of the Shell Research Foundation for the “Algebraic Oil” project which contributed substantially to the success of this research project.

## REFERENCES

- [1] J. Abbott, A. Bigatti, M. Kreuzer and L. Robbiano, Computing ideals of points, *J. Symbolic Comput.* **30** (2000), 341–356.
- [2] B. Buchberger and H.M. Möller, The construction of multivariate polynomials with preassigned zeros, in: J. Calmet (ed.), *Proceedings of EUROCAM'82*, Lect. Notes in Comp. Sci. **144**, Springer, Heidelberg 1982, 24–31.
- [3] The ApCoCoA Team, *ApCoCoA: Applied Computations in Commutative Algebra*, available at <http://www.apcocoa.org>.
- [4] A. Desrochers and S. Mohseni, On determining the structure of a non-linear system, *Int. J. Control* **40** (1984), 923–938.
- [5] C. Fassino, An approximation to the Gröbner basis of ideals of perturbed points: Part I, available at <http://arxiv.org/abs/math/0703154v1>.
- [6] G.H. Golub and C.F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore 1989.
- [7] P.P.N. de Groen, An introduction to total least squares, *Nieuw Archief voor Wiskunde* **14** (1996), 237–254.
- [8] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse, Algebraische Modellierung mit Methoden der approximativen Computeralgebra und Anwendungen in der Ölindustrie, *OR News* **28** (2006), 15–18.
- [9] J.W. Hoffman, J.J. Madden, and H. Zhang, Pseudozeros of multivariate polynomials, *Math. Comp.* **72** (2003), 975–1002.
- [10] A. Kehrein and M. Kreuzer, Characterizations of border bases, *J. Pure Appl. Alg.* **196** (2005), 251–270.
- [11] A. Kehrein and M. Kreuzer, Computing border bases, *J. Pure Appl. Alg.* **205** (2006), 279–295.
- [12] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra 1*, Springer, Heidelberg 2000.
- [13] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra 2*, Springer, Heidelberg 2005.
- [14] M. Kreuzer and L. Robbiano, Deformations of border bases, *Coll. Math.* (to appear)
- [15] M. Marinari, H.M. Möller and T. Mora, Gröbner bases of ideals defined by functionals with an application to ideals of projective points, *Appl. Alg. Eng. Comm. Comput.* **4** (1993), 103–145.
- [16] H.M. Möller and T. Sauer, H-bases II: application to numerical problems, in: A. Cohen *et al.* (eds.), *Curve and Surface Fitting*, Vanderbilt University Press, 1999, 333–342.
- [17] H.M. Möller and T. Sauer, H-bases for polynomial interpolation and system solving, *Adv. in Comp. Math.* **12** (2000), 335–362.
- [18] T. Sauer, Gröbner bases, H-bases and interpolation, *Trans. Amer. Math. Soc.* **353** (2001), 2293–2308
- [19] K. Shirayanagi, M. Sweedler, Remarks on automatic algorithm stabilization, *J. Symb. Comput.* **26** (1998), 761–765.
- [20] H. Stetter, *Numerical Polynomial Algebra*, SIAM, Philadelphia 2004.

FACHBEREICH MATHEMATIK, UNIVERSITÄT DORTMUND, D-44221 DORTMUND, GERMANY  
*E-mail address:* [daniel.heldt@mathematik.uni-dortmund.de](mailto:daniel.heldt@mathematik.uni-dortmund.de)

FAKULTÄT FÜR INFORMATIK UND MATHEMATIK, UNIVERSITÄT PASSAU, D-94030 PASSAU, GERMANY  
*E-mail address:* [martin.kreuzer@uni-passau.de](mailto:martin.kreuzer@uni-passau.de)

FACHBEREICH MATHEMATIK, UNIVERSITÄT DUISBURG-ESSEN, LOTHARSTR. 65, D-47057 DUISBURG, GERMANY  
*E-mail address:* [sebastian.pokutta@uni-due.de](mailto:sebastian.pokutta@uni-due.de)

SHELL INT. EXPLORATION AND PRODUCTION, EXPLORATORY RESEARCH, KESSLER PARK 1, NL-2288 GD RIJSWIJK, THE NETHERLANDS  
*E-mail address:* [hennie.poulisse@shell.com](mailto:hennie.poulisse@shell.com)