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 coodinates of the points are (imprecise) measured data, the resulting Gröbner basis is numerically unstable. In this paper we introduce a numerically stable *Approximate Buchberger-Möller Algorithm*. Moreover, we provide modifications of this algorithm which produce approximate border bases and approximate Macaulay bases of zero-dimensional vanishing ideals, respectively. We also generalize the Border Basis Algorithm ([9]) 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

In his ground-breaking book [17], Stetter presents the following definition of empirical multivariate polynomials which are to be used for modelling real-life phenomena.

Definition 1.1. Let $P = \mathbb{R}[x_1, \ldots, x_n]$ be the polynomial ring over $\mathbb{R}$ in n indeterminates. An *empirical polynomial* is a pair (p, e) where $p = \sum_{j\in J} \alpha_j \mathbf{x}^j \in P$ and the *tolerance* e is a tuple $e = (\varepsilon_j)_{j\in\tilde{J}} \in \mathbb{R}^m_+$. Here $\tilde{J} \subseteq J$ is the *empirical support* of p, the pairs $(\alpha_j, \varepsilon_j)$ are the *empirical coefficients* of p, and $m = \#\tilde{J}$.

Every polynomial $\tilde{p} = \sum_{j\in J} \tilde{\alpha}_j \mathbf{x}^j$ with

$$\|\tilde{p} - p\|^*_e := \left\| (\ldots, \frac{|\tilde{\alpha}_j - \alpha_j|}{\varepsilon_j}, \ldots)_{j\in\tilde{J}} \right\|^* \leq O(1)$$

is called a *valid instance* of p.

The premises of this definition are that there is a *specified* polynomial p and that the uncertainty associated with real-life phenomena can be represented by the uncertainty in one or more of the coefficients of p. Specifically, the empirical support of p is considered to be given. Whereas such a situation may occur in certain problems, in many real-life industrial applications it is not realistic. Yet there is a strong incentive for pursuing approaches using polynomial rings in industrial applications. In addition to the usual linear algebra methods, they offer the opportunity to uncover the always present, performance determining, non-linear interactions between variables or groups of variables in these applications.

Date: August 29, 2006.
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.

To explain the deficiencies of the above definition, we consider the following common situation in industrial applications. Frequently a finite set of data points $\mathbb{X} = \{p_1, \ldots, p_s\} \subset \mathbb{R}^n$, 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*. Now we assume that the outputs are *polynomial functions* of the coordinates of the points p_i. Our goal it to construct polynomial functions *fitting* the measured inputs to the measured outputs. The usefulness, or in modelling jargon the *goodness of fit*, of the polynomial model is then checked in a *validation experiment*: the inputs from another set $\mathbb{Y}$ of measured data points, which have not been used in the fitting process, are substituted as values for the corresponding indeterminates in the constructed polynomial. Then the evaluations obtained in this way are compared to the actual measured outputs.

Clearly there is not anything like a specified polynomial given here. Nor is there structural information in the form of a fixed support available. In other words, in dealing with real-life industrial applications, the starting point is not a *specified* polynomial, but noisy measured data from which a polynomial must be *constructed*. Its construction should be such that it may serve as input for ensuing computer algebra computations. The subsequent algebraic considerations and computations present new mathematical challenges, some of which are addressed in this paper. Acknowledging this new branch of commutative algebra, a new name was coined recently for this sort of developments: *Approximate Commutative Algebra*.

Before we elaborate further on our contributions to this subject presented in this paper, we would like to state a few restrictions we have made. Their justification is mainly based upon the early stage of this new development. 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. We have no doubts, however, that results in *Approximate Differential Algebra* will follow in due course. A second assumption is that, again with reference to the above sketched construction of polynomials, 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.

The rest of this paper is organized as follows. In Section 2 we recall a ubiquitous tool from numerical linear algebra: the singular value decomposition (SVD) of a matrix of real numbers. For easy reference, we include a definition of the *approximate kernel* of a real matrix and an interpretation in terms of a total least squares problem.

Then, in Section 3, we use the SVD to adapt the well-known algorithm of Buchberger-Möller (cf. [2]) to the computation of approximate vanishing ideals. Let us explain what this means.

Definition 1.2. Let $\mathbb{X} = \{p_1, \ldots, p_s\}$ be a finite set of (empirical) points in $\mathbb{R}^n$, and let $P = \mathbb{R}[x_1, \ldots, x_n]$.

(1) The $\mathbb{R}$-linear map $\text{eval} : P \longrightarrow \mathbb{R}^s$ defined by $\text{eval}(f) = (f(p_1), \ldots, f(p_s))$ 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 *ε-approximate vanishing ideal* of $\mathbb{X}$ if there exists a system of generators G of J such that $\|\text{eval}(g)\| < \varepsilon \cdot \|g\|$ for all $g \in G$, where $\|g\|$ denotes the Euclidean norm of the coefficient vector of g.

In the empirical setting described above, we do not want to consider the "true" vanishing ideal of a set of points $\mathbb{X}$ given by approximate data. For instance, there could be polynomials passing *almost* through the points as in the following picture.

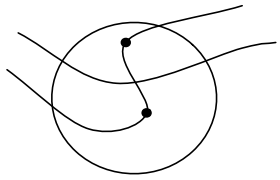

FIGURE 1. passing close vs. going through

Furthermore, the property of having "small" evaluations at the points of $\mathbb{X}$ is not preserved by multiplication of polynomials. Thus we cannot hope to find an ideal consisting entirely of polynomials having "small" evaluations at the points. Instead, we require this property only for the unitary polynomials in a system of generators of an approximate vanishing ideal. The motivation and application of this definition will become clear in Section 3.

Small modifications of the approximate Buchberger-Möller algorithm 3.1 enable us to compute border bases and Macaulay bases of approximate vanishing ideals. The detailed algorithms are spelled out in Corollary 3.3 and Corollary 3.4. A particular feature of the approximate Buchberger-Möller algorithm is that it sometimes produces the vanishing ideal of a nearby *smaller* set of points. This is due to the fact that close-by points should be regarded as approximately equal (see Example 3.5).

In Section 4 we consider a related problem: given polynomials $f_1, \ldots, f_s \in P = \mathbb{R}[x_1, \ldots, x_n]$ 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 approximate Buchberger-Möller algorithm: use the SVD to make the usual border basis algorithm (see [9], Props. 18 and 21) numerically more stable. The precise formulation of the approximate border basis algorithm is given in Theorem 4.5 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.

Unless explicitly stated otherwise, we use the definitions and notations introduced in [10] and [11]. 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.

Theorem 2.1 (The Singular Value Decomposition)**.**
Let $\mathcal{A} \in \mathrm{Mat}_{m,n}(\mathbb{R})$.

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

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

where $\mathcal{D} = \mathrm{diag}(s_1, \ldots, s_r)$ *is a diagonal matrix.*

(2) *In this decomposition, it is possible to achieve* $s_1 \geq s_2 \geq \cdots \geq s_r > 0$. *The numbers* $s_1, \ldots, s_r$ *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:*

first r *columns of* $\mathcal{U}$	$\equiv$	*ONB of the column space of* $\mathcal{A}$
last $m-r$ *columns of* $\mathcal{U}$	$\equiv$	*ONB of the kernel of* $\mathcal{A}^{\mathrm{tr}}$
first r *columns of* $\mathcal{V}$	$\equiv$	*ONB of the row space of* $\mathcal{A}$
	$\equiv$	*ONB of the column space of* $\mathcal{A}^{\mathrm{tr}}$
last $n-r$ *columns of* $\mathcal{V}$	$\equiv$	*ONB of the kernel of* $\mathcal{A}$

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

(1) *We have* $\min\{\|\mathcal{A} - \mathcal{B}\| : \mathrm{rank}(\mathcal{B}) \leq k\} = \|\mathcal{A} - \widetilde{\mathcal{A}}\| = s_{k+1}$.

(2) *The vector subspace* $\mathrm{apker}(\mathcal{A}, \varepsilon) = \ker(\widetilde{\mathcal{A}})$ *is the largest dimensional kernel of a matrix whose Euclidean distance from* $\mathcal{A}$ *is at most* ε. *It is called the* ε-approximate kernel *of* $\mathcal{A}$.

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

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

The number $\varepsilon > 0$ in this corollary is usually called the *threshold number*. For matrices arising from measured data, there is frequently a large gap in the sequence of singular values $(s_1, \ldots, s_r)$, so that there exists a natural choice for the threshold number. The matrix $\widetilde{\mathcal{A}}$ is also called the *ε-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 $\mathbb{X} = \{p_1, \ldots, p_s\} \subset \mathbb{R}^n$ and measured *output* values $q_1, \ldots, q_s \in \mathbb{R}$. If we want to interpolate these data linearly, we are looking for an n-dimensional affine subspace $\hat{c}^\perp = (c_1, \ldots, c_n, 1)^\perp$ that is nearest to the data points, i.e. that minimizes $J(c) = \sum_{i=1}^{s} (\mathbf{z}_i^{\mathrm{tr}} \cdot \hat{c})^2$ where $\mathbf{z}_i = (p_i, -q_i)$. Since $\mathbf{z}_i^{\mathrm{tr}} \cdot \hat{c}/\|\hat{c}\|$ is the distance from $\mathbf{z_i}$ to the subspace, we want to minimize $\mathbf{r} \mapsto \|(\mathbf{z}_1, \ldots, \mathbf{z}_s)^{\mathrm{tr}} \cdot \mathbf{r}\|^2$ 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 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 $\mathcal{A} \in \mathrm{Mat}_{m,n}(\mathbb{R})$ and $i \in \{1, \ldots, n\}$. 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^{th} column of $\mathcal{A}$, and let a_i be the i^{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 $(c_1, \ldots, \widehat{c_i}, \ldots, c_n)^\perp$. If it exists, it corresponds to the kernel of the minimizer of the Frobenius norm $\|\mathcal{A}-\mathcal{B}\|^2$ subject to $\mathrm{rank}(\mathcal{B}) < n$ (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^{th} TLS problem provided its i^{th} component is not zero.

If we use a threshold number $\varepsilon > 0$ and compute the ε-truncation $\widetilde{\mathcal{A}}$ 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 Gröbner basis or border basis polynomials). We will come back to this point later on.

3. Approximate Vanishing Ideals of Points

In this section we present an algorithm which uses the SVD to compute the approximate vanishing ideal of a finite set of points $\mathbb{X} = \{p_1, \ldots, p_s\} \subset \mathbb{R}^n$ given by their (approximate) coordinates. To this end, we modify the usual Buchberger-Möller algorithm (see [2], [13], and [1]). We shall measure the "size" of a polynomial by the Euclidean norm of its coefficient vector. If this norm is one, the polynomial will be called *unitary*. By scaling the coordinates of the points appropriately, we may assume that they are in the interval $[-1, 1]$.

Theorem 3.1 (The Approximate Buchberger-Möller Algorithm)**.**
Let $\mathbb{X} = \{p_1, \ldots, p_s\} \subset [-1, 1]^n \subset \mathbb{R}^n$, let $P = \mathbb{R}[x_1, \ldots, x_n]$, let $\operatorname{eval} : P \longrightarrow \mathbb{R}^s$ be the associated evaluation map $\operatorname{eval}(f) = (f(p_1), \ldots, f(p_s))$, and let $\varepsilon > \varepsilon' > 0$ be small positive numbers. Moreover, we choose a degree compatible term ordering σ. Consider the following sequence of instructions.

- **A1** *Start with lists $G = \emptyset$, $\mathcal{O} = [1]$, a matrix $\mathcal{M} = (1, \ldots, 1)^{\mathrm{tr}} \in \operatorname{Mat}_{s,1}(\mathbb{R})$, and $d = 0$.*
- **A2** *Increase d by one and let $L = [t_1, \ldots, t_\ell]$ be the list of all terms of degree d which are not contained in $\langle \operatorname{LT}_\sigma(g) \mid g \in G \rangle$, ordered decreasingly w.r.t. σ. If $L = \emptyset$, return the pair $(G, \mathcal{O})$ and stop.*
- **A3** *Let m be the number of columns of $\mathcal{M}$. Form the matrix*

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

  Using its SVD, calculate a matrix $\mathcal{B}$ who column vectors are an ONB of the approximate kernel $\operatorname{apker}(\mathcal{A}, \varepsilon)$.
- **A4** *Reduce $\mathcal{B} = (b_{ij})$ to column echelon form. Normalize each column after every reduction step. If at some point a row contains no pivot element of absolute value $> \varepsilon'$ in the untreated columns, replace the corresponding elements of absolute value $\leq \varepsilon'$ by zeros and continue with the next row. The result is a matrix $\mathcal{C} = (c_{ij}) \in \operatorname{Mat}_{\ell+m,k}(\mathbb{R})$ such that $c_{ij} = 0$ for $i < \nu(j)$ and $c_{\nu(j)j} = 1$. (Here $\nu(j)$ denotes the row index of the pivot element in the j^{th} column.)*
- **A5** *For all $i \in \{1, \ldots, \ell\}$ such that there exists a $j \in \{1, \ldots, k\}$ with $\nu(j) = i$ (i.e. for the row indices of the pivot elements), append the polynomial*

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

  to the list G, where $u_{i'}$ is the $(i' - \ell)^{\text{th}}$ element of $\mathcal{O}$.
- **A6** *For all $i = \ell, \ell - 1, \ldots, 1$ such that the i^{th} row of $\mathcal{C}$ contains no pivot element, append the term t_i as a new first element to $\mathcal{O}$, append the column $\operatorname{eval}(t_i)$ as a new first column to $\mathcal{M}$, and continue with step* **A2**.

This is an algorithm which computes a pair $(G, \mathcal{O})$. The list G is a unitary minimal σ-Gröbner basis of the ideal $I = \langle G \rangle \subset P$ and satisfies $\|\operatorname{eval}(g)\| < \delta$ for $\delta = \varepsilon\sqrt{\#G} + \varepsilon' s\sqrt{s}$ and all $g \in G$. Moreover, we have $\dim_{\mathbb{R}}(P/I) \leq s$.

The list $\mathcal{O}$ contains an order ideal of monomials whose residue classes form an $\mathbb{R}$-vector space basis of P/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 **A5**

we construct new elements of G which have leading terms not in $\langle \mathrm{LT}_\sigma(G)\rangle$. By Dickson's Lemma (cf. [10], 1.3.6), this can happen only finitely many times. In step **A6** we enlarge $\mathcal{M}$ by new first columns which are linearly independent of the other columns. Also this can happen only finitely many times. Eventually we arrive at a situation where all new columns $\mathrm{eval}(t_i)$ 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 correctness. In each degree, we show by downward induction on $i = \ell, \ell-1, \ldots, 1$ that the term t_i is either appended to $\mathcal{O}$ or is the leading term of a new element of the unitary minimal σ-Gröbner basis G. Notice that the columns of $\mathcal{A}$ are the evaluation vectors of terms which are ordered decreasingly w.r.t. σ. A column $(c_{1j}, \ldots, c_{\ell+m\,j})^{\mathrm{tr}}$ of $\mathcal{C}$ corresponds to a linear combination of these terms whose evaluation vector has norm $< \varepsilon$. Let $f_1, \ldots, f_k$ be the polynomials of degree $\le d$ given by these linear combinations of terms. By construction, we have $\mathrm{LT}_\sigma(f_j) = t_{\nu(j)}$ for $j = 1, \ldots, k$ and $\mathrm{LT}_\sigma(f_1) >_\sigma \cdots >_\sigma \mathrm{LT}_\sigma(f_k)$. (By step **A4**, the number $\nu(j)$ is the row index of the pivot element in column j of $\mathcal{C}$.)

We process the indices i (and thus the terms t_i) in the order $i = \ell, \ell-1, \ldots, 1$. If row i of $\mathcal{C}$ contains a pivot element $c_{\nu(j)j}$, the polynomial f_j constructed in step **A5** has a leading term $\mathrm{LT}_\sigma(f_j) = t_i$ which is not contained in the monomial ideal generated by the leading terms of the current elements of G. Hence the enlarged set G is unitary and has pairwise non-dividing leading terms.

If row i of $\mathcal{C}$ contains no pivot element, there is no linear relation among the terms in $\mathcal{O} \cup L$ with leading term t_i and leading coefficient $> \varepsilon'$. No leading term of an element of the current set G divides t_i. Since σ is degree compatible, no leading term of the final set G divides t_i. Thus t_i is an element of the complement of $\langle \mathrm{LT}_\sigma(G)\rangle$. It follows that the final set G is a unitary minimal σ-Gröbner basis of the ideal $I = \langle G\rangle$ and the final set $\mathcal{O}$ is the complement of $\mathrm{LT}_\sigma(I)$.

Finally it remains to prove the desired estimate for the size of the evaluation vectors of the elements of G. For $i = 1, \ldots, \ell + m$, let t_i be the term whose evaluation vector is the i^{th} column of $\mathcal{A}$. The column vectors of $\mathcal{B}$ are an ONB of $\mathrm{apker}(\mathcal{A}, \varepsilon)$. By Corollary 2.2, the norm of the vector $\sum_{i=1}^{\ell+m} b_{ij}\,\mathrm{eval}(t_i)$ is $< \varepsilon$ for every j. Thus the norm of the evaluation vector of $\sum_{i=1}^{\ell+m} b_{ij} t_i$ is $< \varepsilon$.

Let v be a unitary vector in the column space of $\mathcal{B}$ and let $f = \sum_{i=1}^{\ell+m} v_i t_i$ be its associated polynomial. We write $v = a_1 b_1 + \cdots + a_k b_k$ where $a_j \in \mathbb{R}$ and b_j is the j^{th} column of $\mathcal{B}$. Then we have $\|\mathrm{eval}(f)\| = \|\sum_{j=1}^{k} a_j\,\mathrm{eval}(\sum_{i=1}^{\ell+m} b_{ij} t_i)\| \le \sum_{j=1}^{k} |a_j|\,\varepsilon \le \varepsilon\sqrt{k}\sum_{j=1}^{k} |a_j|^2 = \varepsilon\sqrt{k}$.

In step **A4** the column space of $\mathcal{B}$ is modified only if the untreated part of some row contains no pivot element $> \varepsilon'$. In this case certain entries are set to zero. This changes the evaluation vector of the associated polynomial by at most $\varepsilon'\sqrt{s}$ because $\|\mathrm{eval}(t_i)\| \le \sqrt{s}$ for all i. Thus the evaluation vector of the associated polynomial of a unitary vector in the column space of $\mathcal{C}$ is $< \varepsilon\sqrt{k} + \varepsilon' s\sqrt{s}$. Since the number of columns of $\mathcal{B}$ is $< \#G$, we obtain $\|\mathrm{eval}(g)\| < \varepsilon\sqrt{\#G} + \varepsilon' s\sqrt{s}$ for all $g \in G$. This concludes the proof. □

Clearly, the assumption that the coordinates of the points of $\mathbb{X}$ are in the interval $[-1, 1]$ is not necessary for the correctness of this algorithm. It was only used to prove the stated bound for δ. 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.

Algorithm 3.1 can be optimized in a variety of ways. For instance, it is not strictly necessary that the "blocks" of terms used in the loop **A2** – **A6** are all the terms of a fixed degree outside $\langle \mathrm{LT}_\sigma(G)\rangle$. Especially in higher degrees it is useful to process "blocks" of terms for which the SVD can be computed efficiently. If the bound δ of the theorem yields insufficient results, we can use the following optimization.

Step control. In Algorithm 3.1 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 ε. The main idea behind this is that, after using a specific ε 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 ε is iteratively adapted until the resulting relations meet the quality requirements. This is a common principle used in numerical calculations which is called *step control*. Using step control for Algorithm 3.1 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. Step control is achieved by modifying the theorem as follows.

Corollary 3.2 (Approximate BM Algorithm with Step Control)**.**
In the setting of the theorem, let $\xi > 0$ the maximal tolerable size of the evaluation vectors of the calculated Gröbner basis polynomials. Replace steps **A5** *and* **A6** *by the following steps* **A5'** *and* **A6'**.

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

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

where $u_{i'}$ is the $(i'-\ell)^{\text{th}}$ element of $\mathcal{O}$. Calculate $\max_j\{\|\operatorname{eval}(f_j)\|\}$ and check whether it is $< \xi$. If this is the case, append the polynomials f_j to G. Otherwise, replace ε by $\varepsilon/10$, replace ε' by $\varepsilon'/10$, and continue with step **A3**.

A6' *For all $i = \ell, \ell-1, \ldots, 1$ such that the i^{th} row of $\mathcal{C}$ contains no pivot element, append the term t_i as a new first element to $\mathcal{O}$, append the column $\operatorname{eval}(t_i)$ as a new first column to $\mathcal{M}$, reset ε and ε' to their original values, and continue with step* **A2**.

The resulting algorithm computes a pair $(G, \mathcal{O})$. The list G is a unitary minimal σ-Gröbner basis of the ideal $I = \langle G\rangle \subset P$ and satisfies $\|\operatorname{eval}(g)\| < \xi$ for all $g \in G$ as well as $\dim_{\mathbb{R}}(P/I) \leq s$. The list $\mathcal{O}$ contains an order ideal of monomials whose residue classes form an $\mathbb{R}$-vector space basis of P/I.

Border Bases. In practice, border bases frequently have better numerical stability than Gröbner bases. (For a definition and the basic properties of border bases, see [11], Section 6.4.) Hence it makes sense to use border bases instead of Gröbner bases for approximate computations. It is easy to change Algorithm 3.1 so that it computes a border basis of an approximate vanishing ideal of a set of points. In fact, it suffices to replace one step in Algorithm 3.1 as follows.

Corollary 3.3 (Border Bases for Zero-Dimensional Vanishing Ideals)**.**
In the setting of the theorem, replace step **A2** *of the algorithm by the following step* **A2'**.

A2' *Increase* d *by one and let* L *be the list of all terms of degree* d*, ordered decreasingly w.r.t.* σ*. Remove from* L *all terms which are contained in* $\langle \mathrm{LT}_\sigma(g) \mid g \in G \rangle$*, but not the ones in the border of* $\mathcal{O}$*. If* $L = \emptyset$*, return the pair* $(G, \mathcal{O})$ *and stop. Otherwise, let* $L = [t_1, \ldots, t_\ell]$*.*

The resulting algorithm computes a pair $(G, \mathcal{O})$*. The set* $\{\mathrm{LC}_\sigma(g)^{-1} g \mid g \in G\}$ *is the* $\mathcal{O}$*-border basis of a* δ*-approximate vanishing ideal* $I = \langle G \rangle \subset P$ *of* $\mathbb{X}$ *where* $\delta < \varepsilon\sqrt{\#G} + \varepsilon' s\sqrt{s}$*. The list* $\mathcal{O}$ *consists of all terms which are not contained in* $\mathrm{LT}_\sigma(I)$*.*

Proof. The list L constructed in step **A2'** contains the terms which are put in the list L in step **A2**. Thus Algorithm 3.3 finds at least as many polynomials in the ε-approximate vanishing ideal of $\mathbb{X}$ in step **A5** as Algorithm 3.1. Since the correctness proof of the theorem is, *mutatis mutandis*, still valid, the computed set G is a σ-Gröbner basis of $I = \langle G \rangle$ which contains the Gröbner basis computed by the theorem.

Furthermore, the border $\partial\mathcal{O}$ of the order ideal $\mathcal{O}$ returned by the algorithm is contained in the union of the sets $\{x_1, \ldots, x_n\} \cdot \mathcal{O}^{(d)}$ where $\mathcal{O}^{(d)}$ is the set of elements of the list $\mathcal{O}$ after the loop **A2**–**A6** has been completed for a particular degree d. Hence every term $b \in \partial\mathcal{O}$ is contained in L at some point. Letting I be the ideal computed by Algorithm 3.1, there exists a polynomial $f = b - \sum_{t_i \in \mathcal{O}} a_i t_i \in I$ with $a_i \in \mathbb{R}$. Hence the polynomial $\|f\|^{-1} f$ is one of the elements appended to G in step **A5**. Thus the list G resulting from Algorithm 3.3 contains the unitary polynomials one obtains by normalizing an $\mathcal{O}$-border prebasis of I where $\mathcal{O} = \mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$. Since that $\mathcal{O}$-border prebasis contains a σ-Gröbner basis of I, it is in fact the $\mathcal{O}$-border basis of I by [8], Prop. 11. □

Clearly, the algorithm in this corollary can by equipped with a step control, too. We leave it to the reader to write down the corresponding version of the algorithm. In Section 6 we present some computational results for the calculation of Gröbner bases and border bases of ε-approximate vanishing ideals. In particular, it turns out that border bases are indeed numerically more stable in the examples considered there.

Macaulay Bases. The approximate Buchberger-Möller 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 [11], 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 [14] and [15]. Strictly speaking, since our Algorithm 3.1 calculates a reduced Gröbner basis with respect to a degree compatible term ordering, we have already found a Macaulay basis for the vanishing ideal. Of course, 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 [15], Thm. 6.4 are satisfied up to an

error whose size is of the order of magnitude of the given threshold number (see also Definition 5.1). In order to find such a basis, we modify Algorithm 3.1 as follows.

Corollary 3.4 (Macaulay Bases for Zero-Dimensional 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* $\mathcal{M} = (1, \ldots, 1)^{\mathrm{tr}} \in \mathrm{Mat}_{s,1}(\mathbb{R})$, *and* $d = 0$.
- **M2** *Increase* d *by one and let* $L = [t_1, \ldots, t_\ell]$ *be the list of all terms of degree* d *which are not contained in* $\langle \mathrm{LT}_\sigma(g) \mid g \in G \rangle$, *ordered decreasingly w.r.t.* σ. *If* $L = \emptyset$, *continue with step* **M8**.
- **M3** *Let* m *be the number of columns of* $\mathcal{M}$. *Form the matrix*

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

  Compute the matrix $\mathcal{B} = (b_{ij})$ *whose column vectors are an ONB of the approximate kernel* $\mathrm{apker}(\mathcal{A}, \varepsilon)$.
- **M4** *Reduce* $\mathcal{B} = (b_{ij})$ *to column echelon form. Normalize each column after every reduction step. If at some point a row contains no pivot element of absolute value* $> \varepsilon'$ *in the untreated columns, replace the corresponding elements of absolute value* $\leq \varepsilon'$ *by zeros and continue with the next row. The result is a matrix* $\mathcal{C} = (c_{ij}) \in \mathrm{Mat}_{\ell+m,k}(\mathbb{R})$ *such that* $c_{ij} = 0$ *for* $i < \nu(j)$ *and* $c_{\nu(j)j} = 1$.
- **M5** *Now start with* $\mathcal{C}$ *and undo all column operations performed in step* **M4**. *Call the resulting matrix* $\widetilde{\mathcal{B}} = (\tilde{b}_{ij})$. *For every* j, *append the polynomial*

$$h_j = \sum_{i'=1}^{\ell} \tilde{b}_{i'j} t_{i'} + \sum_{i'=\ell+1}^{\ell+m} \tilde{b}_{i'j} u_{i'}$$

  to the list H, *where* $u_{i'}$ *is the* $(i'-\ell)^{\mathrm{th}}$ *element of* $\mathcal{O}$.
- **M6** *For all* $i \in \{1, \ldots, \ell\}$ *such that there exists a* $j \in \{1, \ldots, k\}$ *with* $\nu(j) = i$ *(i.e. for the row indices of the pivot elements), append the polynomial*

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

  to the list G, *where* $u_{i'}$ *is the* $(i'-\ell)^{\mathrm{th}}$ *element of* $\mathcal{O}$.
- **M7** *For all* $i = \ell, \ell-1, \ldots, 1$ *such that the* i^{th} *row of* $\mathcal{C}$ *contains no pivot element, append the term* t_i *as a new first element to* $\mathcal{O}$ *and append the column* $\mathrm{eval}(t_i)$ *as a new first column to* $\mathcal{M}$. *Continue with step* **M2**.
- **M8** *Let* $\mathcal{O} = [t_1, \ldots, t_{m'}]$. *Compute the SVD* $\mathcal{M} = \mathcal{U}\mathcal{S}\mathcal{V}^{\mathrm{tr}}$ *of* $\mathcal{M}$. *Put the polynomials* $(q_1, \ldots, q_{m'}) = (t_1, \ldots, t_{m'}) \cdot \mathcal{V}\mathcal{S}^{-1}$ *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* δ*-approximate vanishing ideal* I *of* $\mathbb{X}$, *where we use* $\delta = \varepsilon\sqrt{\#G} + \varepsilon' s\sqrt{s}$ *again. 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 **M4**. 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 elemens of H are almost orthogonal to each other, because the columns of $\mathcal{B}$ form an ONB of apker$(\mathcal{A})$, and the passage from $\mathcal{B}$ to $\mathcal{C}$ is almost undone by the passage from $\mathcal{C}$ to $\widetilde{\mathcal{B}}$ except for the small changes introduced by the second part of step **M4**.

After the last degree d has been treated by the algorithm, the list $\mathcal{O}$ contains a vector space basis of a complement of $\langle G\rangle$ in P, and hence also of a complement of $\langle H\rangle$. From the SVD of $\widetilde{\mathcal{M}}$ we see that $\widetilde{\mathcal{M}}\,\widetilde{\mathcal{V}}\,\widetilde{\mathcal{S}}^{-1} = (u_1, \ldots, u_{m'}, 0, \ldots, 0)$ where the u_i are the first m' columns of $\widetilde{\mathcal{U}}$. Consequently, the evaluation vectors of the elements of $\mathcal{Q}$ are an ONB of the column space of $\widetilde{\mathcal{M}}$. Since the columns of $\widetilde{\mathcal{M}}$ are exactly the evaluation vectors of the elements of $\mathcal{O}$, the claim follows. □

We would like to point out that the algorithm of this corollary can be optimized substantially: if no element has to be set equal to zero in step **M4**, we can take the matrix $\mathcal{B}$ instead of computing $\widetilde{\mathcal{B}}$ in step **M5**. Similarly, if the elements we set equal to zero in step **M4** are few and much smaller than ε, 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 an 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 3.1 we were careful to state that the computed ideal I satisfies the inequality $\dim_{\mathbb{R}}(P/I) \leq s$ but not necessarily the equality. Apparently the zero-set of I may consist of *less* than s points. Is this really possible? Yes, it is! Below we exhibit an easy example 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 "almost" vanish on one point if and only if it "almost" vanishes 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 as in [5]. Now let us have a look at the example.

Example 3.5. (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 approximate Buchberger-Möller Algorithm 3.1. The evaluation matrix in degree one is

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

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, for $\delta = \varepsilon\sqrt{\#G} + \varepsilon' s\sqrt{s} \approx 0.85$, the δ-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!

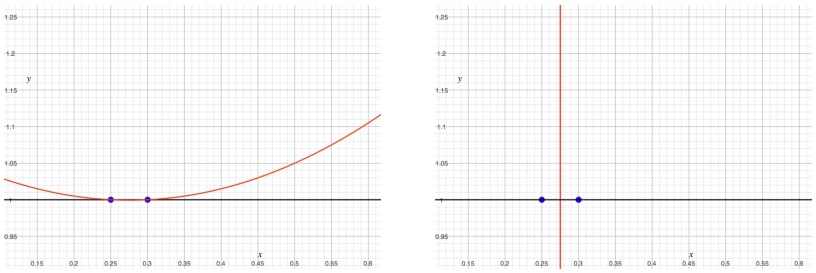

FIGURE 2. Two close-by points become one "approximate" point

4. Approximate Border Basis Computation

Next we want to address the following problem: Suppose we are given "empirical" polynomials $f_1, \ldots, f_s \in P = \mathbb{R}[x_1, \ldots, x_n]$ with $s \geq n$. If $s > n$, the ideal $\langle f_1, \ldots, f_s \rangle$ 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.

The first step is to represent finite dimensional vector spaces of polynomials by matrices. Let $V \subset P$ be a vector space, let $f_1, \ldots, f_s$ be a basis of V, and let $\mathrm{Supp}(f_1) \cup \cdots \cup \mathrm{Supp}(f_s) = \{t_1, \ldots, t_r\}$. Then V can be represented by the matrix

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

where $f_j = a_{1j}t_1 + \cdots + a_{rj}t_j$ with $a_{ij} \in \mathbb{R}$.

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 *ε-small* if the norm of its coefficient vector is less than ε. Thus the next step is to use the SVD of $\mathcal{A}$ to filter out some ε-small polynomials in V.

For this purpose, we compute the SVD of $\mathcal{A}$ and get $\mathcal{A} = \mathcal{U}\,\mathcal{S}\,\mathcal{V}^{\mathrm{tr}}$ as in Theorem 2.1. If we now replace all singular values $s_i < \varepsilon$ by zeroes, the resulting matrix $\widetilde{\mathcal{A}} = \mathcal{U}\,\widetilde{\mathcal{S}}\,\mathcal{V}^{\mathrm{tr}}$ represents the polynomial vector space V_{ap} of smallest rank which is "close" to V (see Corollary 2.2).

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 a threshold number $\varepsilon > 0$, we first replace V by V_{ap}. Let $\mathcal{A} = (a_{ij}) \in \mathrm{Mat}_{r,s}(\mathbb{R})$ be a matrix representing V_{ap}. By choosing the ONB $\{f_1, \ldots, f_s\}$ of V_{ap} provided by the columns of the matrix $\mathcal{U}$ in the SVD, we may assume that $r \geq s$ and $\mathcal{A} = \mathcal{U}_1$ where $\mathcal{U}_1$ consists of the first s columns of the orthogonal matrix $\mathcal{U}$ of size $r \times r$.

Definition 4.1. Let $\varepsilon > 0$ be a given threshold number, and let σ be a degree compatible term ordering.

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

If the threshold number ε 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 $\mathrm{LT}^{\mathrm{ap}}_{\sigma}(V)$. 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 polynomials in V. For this purpose we may have to modify V slightly as follows.

Remark 4.2. [Approximate Leading Term Bases]
Let $\mathcal{A} = (a_{ij}) \in \mathrm{Mat}_{r,s}(\mathbb{R})$ be a matrix representing a vector space of polynomials $V = \langle f_1, \ldots, f_s \rangle_{\mathbb{R}}$ with $V = V_{\mathrm{ap}}$ as above. Here (and later) the notation $\langle \ldots \rangle_{\mathbb{R}}$ denotes the $\mathbb{R}$-linear span. Recall that the rows of $\mathcal{A}$ are indexed by the terms in $\mathrm{Supp}(V)$. We choose a degree compatible term ordering σ and order the terms such that larger terms w.r.t. σ correspond to higher rows (i.e. to rows having smaller row indices). Then we perform the following steps.

(1) Bring $\mathcal{A}$ into column echelon form, using only pivot elements whose absolute value is $> \varepsilon$. Normalize the columns after each reduction step.
(2) If the untreated part of some row does not contain an element of absolute value $> \varepsilon$, replace the corresponding elements of absolute value $\leq \varepsilon$ by zeros and continue with the next row.
(3) If any column of $\mathcal{A}$ is zero, delete it.

The resulting matrix $\mathcal{A}'$ represents a vector space of polynomials $V' \subset P$. Let $f'_1, \ldots, f'_s \in V'$ be the unitary vector space basis of V' given by the columns of $\mathcal{A}'$. Then every unitary polynomial $f \in V'$ satisfies
$$\mathrm{LT}^{\mathrm{ap}}_{\sigma}(f) \in \mathrm{LT}^{\mathrm{ap}}_{\sigma}(V') = \{\mathrm{LT}^{\mathrm{ap}}_{\sigma}(f_1), \ldots, \mathrm{LT}^{\mathrm{ap}}_{\sigma}(f_s)\}$$
We shall say that $\{f'_1, \ldots, f'_s\}$ is an *approximate leading term basis* of V'. Moreover, the vector space V' is "close to" V in the sense that, for all $i \in \{1, \ldots, s\}$, there exists a unitary polynomial $\tilde{f}_i \in V$ such that $\|f'_i - \tilde{f}_i\| < (r - s)\varepsilon$. This follows from the observation that setting a coefficient of absolute value $< \varepsilon$ to zero changes the norm of a polynomial by less than ε.

Next we extend the procedure of passing from V to V_{ap} to the following construction. In the Border Basis Algorithm (cf. [9], Prop. 18) the vector space V is repeatedly replaced by
$$V^+ = V + x_1 V + \cdots + x_n V.$$
The approximate version of this construction works as follows.

Remark 4.3. [Approximate Leading Term Basis Extension]
Let $V \subset P$ be a vector space of polynomials, let $\varepsilon, \varepsilon' > 0$ be threshold numbers, and let σ be a degree compatible term ordering.

(1) Compute the matrix $\mathcal{A}'$ which represents an approximate leading term basis $\{f'_s, \ldots, f'_s\}$ of V'_{ap} as above. (Recall that higher rows of $\mathcal{A}'$ correspond to larger terms w.r.t. σ.)
(2) The representing matrix of $(V'_{\mathrm{ap}})^+ = V'_{\mathrm{ap}} + x_1 V'_{\mathrm{ap}} + \cdots + x_n V'_{\mathrm{ap}}$ is of the form $(\widehat{\mathcal{A}} \mid \mathcal{B})$ where $\widehat{\mathcal{A}}$ is obtained by enlarging $\mathcal{A}'$ by some zero rows corresponding to new terms in the support of $(V'_{\mathrm{ap}})^+$. If necessary, resort the rows of the matrix $(\widehat{\mathcal{A}} \mid \mathcal{B})$ such that higher rows correspond to larger terms. Using the pivot entries in $\widehat{\mathcal{A}}$ corresponding to the approximate leading terms of the polynomials f'_i, clean out the corresponding rows of $\mathcal{B}$ and get a matrix $\mathcal{B}'$.
(3) Delete the zero rows of $\mathcal{B}'$. Compute the ε'-truncation of the SVD of the resulting matrix and get $\widetilde{\mathcal{B}} = \mathcal{U}' \cdot \mathcal{S}' \cdot (\mathcal{V}')^{\mathrm{tr}}$.
(4) Let $\mathcal{U}'_1$ be the first columns of $\mathcal{U}'$ which form an ONB of the column space of $\widetilde{\mathcal{B}}$ (see Theorem 2.1.4). Using the method of Remark 4.2, compute an approximate leading term basis $\{f'_{s+1}, \ldots, f'_{s'}\}$ of a vector space W' close to the vector space W represented by $\widetilde{\mathcal{B}}$.

Then the set $\{f'_1, \ldots, f'_{s'}\}$ is an approximate leading term basis of the vector space $V'_{\mathrm{ap}} \oplus W' \subset P$ which is close to the vector space $(V'_{\mathrm{ap}})^+$.

The last ingredient we need is an approximate version of the computation of a stable span explained in [9], Prop. 13. In the following we say that a polynomial $f \in P$ is *δ-close* to a vector subspace V of P if there exists a polynomial $v \in V$ such that $\|f - v\| < \delta$.

Proposition 4.4. *Let $f_1, \ldots, f_s \in P$ be linearly independent unitary polynomials, let $V = \langle f_1, \ldots, f_s \rangle_{\mathbb{R}}$, let $U = \langle \mathbb{T}^n_{\leq d} \rangle_{\mathbb{R}}$ for some $d \geq \max\{\deg(f_1), \ldots, \deg(f_s)\}$, let σ be a degree compatible term ordering, and let $\varepsilon > 0$ be a threshold number. We perform the following steps.*

(1) *Using Remark 4.2, compute a unitary basis $\mathcal{V} = \{f'_1, \ldots, f'_r\}$ of a vector space V'_{ap} which is an approximate leading term basis of V'_{ap}.*
(2) *Using Remark 4.3, compute a unitary basis extension $\mathcal{W}'$ for $V'_{\mathrm{ap}} \subseteq (V'_{\mathrm{ap}})^+$ so that the elements of $\mathcal{V} \cup \mathcal{W}'$ are an approximate leading term basis of a vector space close to $(V'_{\mathrm{ap}})^+$.*
(3) *Let $\mathcal{W} = \{f'_{r+1}, \ldots, f'_{r+\varrho}\} = \{p \in \mathcal{W}' \mid \deg(p) \leq d\}$.*
(4) *If $\varrho > 0$ then replace $\mathcal{V}$ with $\mathcal{V} \cup \mathcal{W}$, increase r by ϱ, and go to step 2.*
(5) *Return $\mathcal{V}$.*

This is an algorithm which computes a set of unitary polynomials $\mathcal{V} = \{f'_1, \ldots, f'_r\}$ which is an approximate leading term basis of $\widetilde{V} := \langle f'_1, \ldots, f'_r \rangle_{\mathbb{R}}$, such that the original polynomials $f_1, \ldots, f_s$ are δ-close to $\widetilde{V}$ for $\delta = \varepsilon(\#\operatorname{Supp}(V) - s + 1)$, and such that $\widetilde{V}$ is approximately U-stable, i.e. such that we have $(\widetilde{V}'_{\mathrm{ap}})^+ \cap U = \widetilde{V}$.

Proof. The method of the proof of [9], 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 $\widetilde{V}$ 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 δ-close to the computed vector space $\widetilde{V}$ follows from the facts that V_{ap} is ε-close to V (see Corollary 2.2) and V'_{ap} is $\varepsilon(\#\operatorname{Supp}(V) - s)$-close to V_{ap} (see Remark 4.2). Clearly, extensions of this vector space cannot increase the distances under consideration. □

Combining the preceding steps, we can formulate an approximate version of the Border Basis Algorithm [9], Prop. 18. Recall that, for a polynomial ideal I, the order ideal of all terms not in $\mathrm{LT}_\sigma(I)$ is denoted by $\mathcal{O}_\sigma(I)$.

Theorem 4.5 (The Approximate Border Basis Algorithm (ABBA))**.**
Let $\{f_1, \ldots, f_s\} \subset P = \mathbb{R}[x_1, \ldots, x_n]$ be a linearly independent set of $s \geq n$ unitary polynomials, let $V = \langle f_1, \ldots, f_s\rangle_{\mathbb{R}}$, let σ be a degree-compatible term ordering, and let $\varepsilon > 0$. The following algorithm computes the $\mathcal{O}_\sigma(I)$-border basis $\{g_1, \ldots, g_\nu\}$ of an ideal $I = \langle g_1, \ldots, g_\nu\rangle$ such that $f_1, \ldots, f_s$ are δ-close to I for $\delta = \varepsilon(\#\operatorname{Supp}(V) - s + 1)$ and such that $\dim_{\mathbb{R}}(P/I)$ is as large as possible.

- **B1** *Let $d = \max\{\deg(f_i) \mid 1 \leq i \leq s\}$ and $U = \langle \mathbb{T}^n_{\leq d}\rangle_{\mathbb{R}}$.*
- **B2** *Using Remark 4.2, compute a unitary basis $\mathcal{V} = \{f'_1, \ldots, f'_r\}$ of a vector space V'_{ap} which is an approximate leading term basis of V'_{ap}.*
- **B3** *Using Remark 4.3, compute a unitary basis extension $\mathcal{W}'$ for $V'_{\mathrm{ap}} \subseteq (V'_{\mathrm{ap}})^+$ so that the elements of $\mathcal{V} \cup \mathcal{W}'$ are an approximate leading term basis of a vector space close to $(V'_{\mathrm{ap}})^+$.*
- **B4** *Let $\mathcal{W} = \{f'_{r+1}, \ldots, f'_{r+\varrho}\} = \{p \in \mathcal{W}' \mid \deg(p) \leq d\}$.*
- **B5** *If $\varrho > 0$ then replace $\mathcal{V}$ with $\mathcal{V} \cup \mathcal{W}$, increase r by ϱ, and go to (B3).*
- **B6** *Let $\mathcal{O} = \mathbb{T}^n_{\leq d} \setminus \{\mathrm{LT}^{\mathrm{ap}}_\sigma(f'_1) \ldots \mathrm{LT}^{\mathrm{ap}}_\sigma(f'_r)\}$.*
- **B7** *If $\partial\mathcal{O} \not\subseteq U$ then increase d by one, update $U := \langle \mathbb{T}^n_{\leq d}\rangle_{\mathbb{R}}$, and go to (B3).*
- **B8** *Apply the Final Reduction Algorithm and return its result $(g_1, \ldots, g_\nu)$.*

Here the Final Reduction Algorithm is the algorithm described in Prop. 17 of [9].

Proof. Mutatis mutandis, it suffices to follow the proof of Prop. 18 in [9] 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 $\{f'_1, \ldots, f'_r\}$ 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 $t' = \mathrm{LT}^{\mathrm{ap}}_\sigma(f'_i)$ for some i. There exists a polynomial f'_i whose coefficient at t' is not small. If we multiply f'_i by the appropriate product of indeterminates, the coefficient of t in the resulting polynomial is not small. Thus, after several extensions of V, the term t has to be the approximate leading term of some polynomial f'_j, in contradiction to our assumption.

The claim that the original polynomials are δ-close to the computed ideal I was shown in Proposition 4.4. The last claim follows from Corollary 2.2: Since the SVD truncation produces the smallest dimension of V_{ap} resp. V^+_{ap} at each step, the codimension of I will be maximal under the hypothesis that $f_1, \ldots, f_s$ are δ-close to I. □

Notice that steps B2)–B5) are nothing but the algorithm of the preceding proposition. As mentioned in [9], this algorithm can be optimized substantially. In particular, the proof of [9], Prop. 21 shows that we can reduce the size of the space U (the "computational universe") dramatically. We get the following improved algorithm.

Corollary 4.6 (Improved Approximate Border Basis Algorithm (IABBA))**.** *In the setting of the theorem, the following algorithm computes the $\mathcal{O}_\sigma(I)$-border basis $\{g_1, \ldots, g_\nu\}$ of an ideal $I = \langle g_1, \ldots, g_\nu \rangle$ such that $f_1, \ldots, f_s$ are δ-close to I and such that $\dim_{\mathbb{R}}(P/I)$ is as large as possible.*

- **C1** *Let $\mathcal{L}$ be the order ideal spanned by $\bigcup_{i=1}^{r} \operatorname{Supp}(f_i)$.*
- **C2** *Using Remark 4.2, compute a unitary basis $\mathcal{V} = \{f'_1, \ldots, f'_r\}$ of a vector space V'_{ap} which is an approximate leading term basis of V'_{ap}.*
- **C3** *Using Remark 4.3, 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 $\mathcal{W} = \{f \in \mathcal{W}' \mid \operatorname{LT}_\sigma(f) \in \mathcal{L}\}$.*
- **C5** *If $\bigcup_{f \in \mathcal{W}} \operatorname{Supp}(f) \not\subseteq \mathcal{L}$, then replace $\mathcal{L}$ with the order ideal spanned by $\mathcal{L}$ and $\bigcup_{f \in \mathcal{W}} \operatorname{Supp}(f)$ and continue with step* **C4**.
- **C6** *If $\mathcal{W} \neq \emptyset$ then replace $\mathcal{V}$ with $\mathcal{V} \cup \mathcal{W}$ and continue with step* **C3**.
- **C7** *Let $\mathcal{O} = \mathcal{L} \setminus \{\operatorname{LT}_\sigma(v) \mid v \in \mathcal{V}\}$.*
- **C8** *If $\partial\mathcal{O} \not\subseteq \mathcal{L}$ then replace $\mathcal{L}$ with 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 $g_1, \ldots, g_\nu$ computed by it.*

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

Example 4.7. Consider the (approximately) unitary polynomials

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

We apply ABBA with $\varepsilon = 0.001$ 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 $f_1, \ldots, f_6$ and

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

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 $f_1, \ldots, f_6$ together with

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

This result is in good numerical agreement with the exact result in [9], Ex. 20.

5. Approximate Membership for Zero-Dimensional Ideals

Given a polynomial ideal $I = \langle f_1, \ldots, f_s\rangle \subseteq P$ and a polynomial $g \in P$, the classical ideal membership problem asks whether we have $g \in I$ or not. If this is the case, explicit membership is the quest for a concrete representation $g = h_1f_1 + \cdots + h_sf_s$ with $h_1, \ldots, h_s \in P$. The standard way to solve the decision problem is to choose a term ordering σ, compute a σ-Gröbner basis $G = \{g_1, \ldots, g_t\}$ of I and use the division algorithm [10], 1.6.4, to decide whether $\mathrm{NF}_{\sigma,G}(g) = 0$. The standard method for solving the explicit membership problem consists of invoking the extended Buchberger algorithm (cf. [10], 2.5.11), computing the syzygy module of the Gröbner basis and transforming the syzygies (cf. [10], 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 $\mathrm{NF}_{\sigma,G}(g)$ 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 "uniquely determined" explicit representations $g = h_1f_1 + \cdots + h_sf_s$. If 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 general problem.

Given an empirical zero-dimensional polynomial ideal I, compute an order ideal $\mathcal{O} = \{t_1, \ldots, t_\mu\}$ and an $\mathcal{O}$-border basis G of I. (If I is defined as the vanishing ideal of an approximate set of points, use Corollary 3.3. If I is given by an approximate system of generators, use Theorem 4.5.)

Suppose that the order ideal $\mathcal{O}$ is of the form $\mathcal{O}_\sigma(I) = \mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$ for some degree compatible term ordering σ. Then G is automatically a σ-Gröbner basis,

and henceforth a Macaulay basis of I (cf. [11], 6.4.18 and 4.2.15). If the order ideal $\mathcal{O}$ is not of this form, we can still try to use Corollary 3.4 or [15], Section 4. In either case, we assume that we now have a Macaulay basis $H = \{h_1, \ldots, h_\lambda\}$ 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 $H = \{h_1, \ldots, h_\lambda\}$ 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 $f = g_1 h_1 + \cdots + g_\lambda h_\lambda + \mathrm{NF}_I(f)$ (cf. [15], 6.2 or [16], 4.3) we have $f = \mathrm{NF}_I(f)$.

(2) The Macaulay basis H is called a *completely reduced, orthogonal Macaulay basis* of I if all $h_i - \mathrm{DF}(h_i)$ are completely reduced w.r.t. H and

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

Here $\mathrm{DF}(f)$ denotes the *degree form* of a polynomial f (see [11], 4.2.8).

Given H, a completely reduced, orthogonal Macaulay basis of I can be computed easily (cf. [15], 6.4). Moreover, it is essentially unique (cf. [15], 6.5). Therefore we shall from now on assume that H has this property.

Remark 5.2. [Approximate Membership Using Macaulay Bases]
Let $H = \{h_1, \ldots, h_\lambda\}$ be a completely reduced, orthogonal Macaulay basis of I. For any polynomial $f \in P$ of degree d, we can use the Macaulay Division Algorithm (cf. [16], 3.1 and [15], 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 $\mathrm{COP}_I(f) = r_0 + \cdots + r_d$ is called the *canonical orthogonal projection* of f w.r.t. I.

If we set $f \approx 0 \Leftrightarrow \| \mathrm{COP}_I(f) \| < \varepsilon$ 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. [17] and [12]).

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 $(h_1, \ldots, h_\lambda)$. 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 $G = \{g_1, \ldots, g_\nu\}$ of I, we compute the completely reduced, orthogonal Macaulay basis $H = \{h_1, \ldots, h_\lambda\}$ of I. Since the residue classes of the terms in $\mathcal{O}_\sigma(I) = \{t_1, \ldots, t_\mu\}$ form a vector space basis of P/I, we may then calculate a set of polynomials $\mathcal{P} = \{p_1, \ldots, p_\mu\}$ 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 3.4 to get $\mathcal{P}$. If I is given by an approximate set of generators, we can use Theorem 4.5 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. [16], 3.1 and [15], 6.2) as follows: if we want to project an element $f \in P_{\leq d}$, we take its degree form $\mathrm{DF}(f)$ and project it onto $\langle \mathrm{DF}(h_i) \mid \deg(h_i) = d \rangle_{\mathbb{R}}$. 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 $\mathrm{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$ of f is the unique representation of $\mathrm{COP}_I(f)$ 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 $\mathcal{P} = \{p_1, \ldots, p_\mu\}$ be an ONB of the orthogonal complement of I in P, and let $\varepsilon > 0$. We write $\mathrm{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$ with $c_i \in \mathbb{R}$. Then the polynomial $\mathrm{NF}^{\mathrm{ap}}_{\mathcal{P},I}(f) = \sum_{\{i:|c_i|\geq\varepsilon\}} c_i p_i$ 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 $\mathcal{P} = \{p_1, \ldots, p_\mu\}$ 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 $\mathrm{NF}^{\mathrm{ap}}_{\mathcal{P},I}(f) = 0$. More precisely, if $\mathrm{NF}^{\mathrm{ap}}_{\mathcal{P},I} = 0$ then f is $\varepsilon\sqrt{\mu}$-close to I, and if f is ε-close to I then $\mathrm{NF}^{\mathrm{ap}}_{\mathcal{P},I}(f) = 0$.*
2. *If f is ε-close to I, we use the Macaulay Division Algorithm to compute a representation*

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

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

Proof. To show the first claim, we start by assuming that we have $\mathrm{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$ 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 $\sqrt{c_1^2 + \cdots + c_\mu^2} \leq \varepsilon\sqrt{\mu}$. Conversely, if the canonical projection $\mathrm{COP}_I(f) = c_1 p_1 + \cdots + c_\mu p_\mu$ satisfies $\sqrt{c_1^2 + \cdots + c_\mu^2} \leq \varepsilon$, then we have $|c_i| < \varepsilon$ for $i = 1, \ldots, \mu$.

Now we prove the second claim. Let $f = g_1 h_1 + \cdots + g_\lambda h_\lambda + c_1 p_1 + \cdots + c_\mu p_\mu$ and $f' = g'_1 h_1 + \cdots + g'_\lambda h_\lambda + c'_1 p_1 + \cdots + c'_\mu p_\mu$ 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 $i = 1, \ldots, \lambda$. The hypotheses that f and f' are ε-close to I yield $\sqrt{c_1^2 + \cdots + c_\mu^2} \leq \varepsilon$ and $\sqrt{(c'_1)^2 + \cdots + (c'_\mu)^2} \leq \varepsilon$.

Hence the norm of $f - f' = (c_1 - c'_1)p_1 + \cdots + (c_\mu - c'_\mu)p_\mu$ is at most 2ε, and the approximate normal form of $f - f'$ (with respect to the threshold number 2ε) is zero. □

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 Gröbner or border bases. All computations were done on a Apple Powerbook G4 (1.5 GHz) using a prototypical implementation. An implementation using CoCoA (cf. [3]) will be available soon.

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 $p_i = (p_{i1}, \ldots, p_{i9})$ in $\mathbb{R}^9$. The absolute values of the coordinates are less than

(3.2492, 8.2528, 3.2501, 2.3963, 3.5115, 1.6226, 7.1821, 1.8006, 6.5006)

The following table shows the results for the computation of a `DegLex`-Gröbner basis of the approximate vanishing ideal using Algorithm 3.1 with ε' chosen according to the condition number of the resulting matrices and the stated values of ε.

	ε	**#GB**	**deg**	**max $f(p_i)$-error**	**mean $f(p_i)$-error**	**time**
1	1	15	≤ 3	0.0115	0.0025	0.77 s
2	0.5	19	≤ 3	0.0063	$8.1262 \cdot 10^{-4}$	0.84 s
3	0.1	24	≤ 3	$6.9260 \cdot 10^{-4}$	$2.7851 \cdot 10^{-4}$	0.82 s
4	0.01	39	≤ 4	$9.0761 \cdot 10^{-5}$	$1.7142 \cdot 10^{-5}$	1.43 s
5	0.001	59	≤ 4	$1.5547 \cdot 10^{-6}$	$1.0135 \cdot 10^{-6}$	2.34 s
6	0.0001	82	≤ 4	$4.8458 \cdot 10^{-7}$	$1.0858 \cdot 10^{-7}$	4.04 s
7	0.00001	125	≤ 4	$8.4620 \cdot 10^{-8}$	$8.0948 \cdot 10^{-9}$	7.31 s
8	0.0 (full)	260	≤ 5	$4.6954 \cdot 10^{-11}$	$2.0193 \cdot 10^{-11}$	22.04 s

Table 1. Calculating a `DegLex`-Gröbner Basis for 2445 Points in $\mathbb{R}^9$

Table 2 shows the timings and numerical quality of a calculation using the same data set, but the algorithm of Corollary 3.3 to compute a border basis for the approximate vanishing ideal. It may be worth mentioning that the algorithms identify close-by data points even if we choose $\varepsilon = 0$. (Since we are using floating point computations, this amounts to choosing the machine precision for ε.) For instance, in the example at hand the corresponding order ideal $\mathcal{O}$ has only 179 elements.

Our second computational test (see Table 3) is the calculation of a Gröbner basis of an ε-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.

	ε	**#BB**	**deg**	**max $f(p_i)$-error**	**mean $f(p_i)$-error**	**time**
1	1	94	≤ 6	0.0115	$5.5523 \cdot 10^{-4}$	1.86 s
2	0.5	88	≤ 5	0.0063	$3.4916 \cdot 10^{-4}$	1.72 s
3	0.1	185	≤ 8	$6.9626 \cdot 10^{-4}$	$3.1908 \cdot 10^{-5}$	4.19 s
4	0.01	163	≤ 5	$9.0761 \cdot 10^{-5}$	$3.3018 \cdot 10^{-6}$	4.60 s
5	0.001	223	≤ 5	$1.5547 \cdot 10^{-6}$	$2.1958 \cdot 10^{-7}$	8.07 s
6	0.0001	306	≤ 6	$4.8458 \cdot 10^{-7}$	$2.4311 \cdot 10^{-8}$	13.55 s
7	0.00001	345	≤ 5	$8.4620 \cdot 10^{-8}$	$3.4753 \cdot 10^{-9}$	20.52 s
8	0.0 (full)	539	≤ 5	$3.6442 \cdot 10^{-11}$	$1.3308 \cdot 10^{-11}$	53.37 s

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

	ε	**#GB**	**deg**	**max $f(p_i)$-error**	**mean $f(p_i)$-error**	**time**
1	0.1	63	≤ 3	$1.1890 \cdot 10^{-4}$	$6.5252 \cdot 10^{-5}$	7.64 s
2	0.01	272	≤ 5	$6.4377 \cdot 10^{-7}$	$8.1200 \cdot 10^{-8}$	104.53 s
3	0.001	464	≤ 5	$1.9933 \cdot 10^{-7}$	$1.9786 \cdot 10^{-8}$	196.38 s
4	0.0001	464	≤ 5	$1.9933 \cdot 10^{-7}$	$1.9786 \cdot 10^{-8}$	317.51 s
5	0.00001	822	≤ 6	$6.0032 \cdot 10^{-9}$	$5.7087 \cdot 10^{-10}$	927.94 s
6	0.000001	1217	≤ 6	$2.0084 \cdot 10^{-9}$	$2.1326 \cdot 10^{-10}$	3683.35 s
7	0.0 (full)	1965	≤ 7	$6.7836 \cdot 10^{-11}$	$4.5715 \cdot 10^{-12}$	11632.41 s

TABLE 3. Calculating a `DegLex`-Gröbner Basis for 10105 Points in $\mathbb{R}^9$

Calculating Approximate Border Bases. To test the Approximate Border Basis Algorithm, 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 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 12 polynomials of degrees 4,4,4,4,5,5,5,5,5,6,6,6. The threshold number used for pivoting in Remarks 4.2 and 4.3 is $\varepsilon = 10^{-5}$. The number ε' in the following table is the cut-off value for the SVD computed in step 3 of Remark 4.3.

Note that the execution time initially decreases with decreasing ε'. 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 $\varepsilon = 10^{-10}$.

These timings were obtained with a prototype implementation on a small laptop. It is to be expected that the optimized implementation under development in CoCoA 5 (cf. [3]) using IABBA (see Corollary 4.6) will be significantly faster.

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,

	ε'	**#BB**	**deg**	**time**
1	0.1	81	≤ 6	2.81 s
2	0.01	83	≤ 6	1.44 s
3	0.001	84	≤ 6	0.70 s
4	0.0001	84	≤ 6	0.70 s
5	0.00001	84	≤ 6	0.72 s
6	0.000001	84	≤ 6	0.71 s

TABLE 4. Calculating a Border Basis of 84 Polynomials with ABBA

	ε	**#BB**	**deg**	**time**
1	0.1	220	≤ 9	59.60 s
2	0.01	173	≤ 9	27.70 s
3	0.001	216	≤ 9	11.95 s
4	0.0001	220	≤ 9	5.34 s
5	0.00001	220	≤ 9	4.94 s
6	0.000001	220	≤ 9	5.17 s

TABLE 5. Calculating a Border Basis of 220 Polynomials with ABBA

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 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.

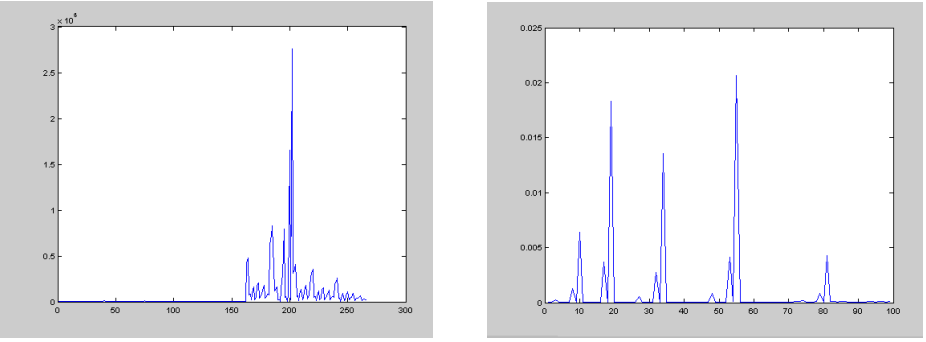

FIGURE 3. Mean Evaluation Errors Without and With Data Scaling

From an algebraic point of view, the algorithm produces the 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.

Altogether, 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 [13]). 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 approximate Buchberger-Möller algorithm 3.1 provides us with the kind of relations we are searching for. In fact, if we use the variant given in Corollary 3.3, we get a border basis that also provides additional stability with respect to variations of the input data (cf. [17]).

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 treatment of these applications will be provided in a separate paper.

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 *a* 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 the approximate vanishing ideal from Section 3 of the test set points helps us establish these relations among the input variables. Having computed 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 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 3.4.

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 our problem. 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 $f_1, \ldots, f_s$ 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 $f = h_1 f_1 + \cdots + h_s f_s$ 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. 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 CoCoA Team, CoCoA*: a system for doing Computations in Commutative Algebra*, available at `http://cocoa.dima.unige.it`.

[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, preprint 2006.

[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] A. Kehrein and M. Kreuzer, Characterizations of border bases, J. Pure Appl. Alg. **196** (2005), 251–270.

[9] A. Kehrein and M. Kreuzer, Computing border bases, J. Pure Appl. Alg. **205** (2006), 279–295.

[10] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra 1*, Springer, Heidelberg 2000.

[11] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra 2*, Springer, Heidelberg 2005.

[12] M. Kreuzer and L. Robbiano, Deformations of border bases, in preparation.

[13] 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.

[14] 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.

[15] H.M. Möller and T. Sauer, H-bases for polynomial interpolation and system solving, Adv. in Comp. Math. **12** (2000), 335–362.

[16] T. Sauer, Gröbner bases, H-bases and interpolation, Trans. Amer. Math. Soc. **353** (2001), 2293–2308

[17] 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

Fachbereich Mathematik, Universität Dortmund, D-44221 Dortmund, Germany
E-mail address: martin.kreuzer@uni-dortmund.de

Fachbereich Mathematik, Universität Duisburg-Essen, Lotharstr. 65, D-47057 Duisburg, Germany
E-mail address: 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