Advanced Studies in Pure Mathematics 77, 2018
The 50th Anniversary of Gröbner Bases
pp. 87–107

BM algorithms for noisy data and implicit regression modelling

Claudia Fassino, Hans Michael Möller and Eva Riccomagno

Abstract.

In this paper we consider the problem of finding a set of monomials $\mathcal{O}$ and a polynomial f whose support is contained in $\mathcal{O}$, such that (1) f is almost vanishing at a set of points $\mathbb{X}$ whose coordinates are not known exactly and (2) $\mathcal{O}$ exhibits structural stability, that is the model/design matrix associated to $\mathcal{O}$ is full rank for each set of points differing only slightly from $\mathbb{X}$. We review some numerical versions of the Buchberger-Möller (BM) algorithm for computing the set $\mathcal{O}$ and the polynomial f and we present a variant, called LDP-LP, which integrates one of these methods with a classical statistical least squares algorithm for implicit regression from [1]. To illustrate the usefulness of these numerical BM algorithms, we review some of their application in the analyses of data sets for which standard techniques did not yield satisfactory results.

§1. Introduction

For n-distinct points $\mathbb{X} \subset \mathbb{R}^k$ we consider the generic problem of finding a polynomial function f which almost vanishes at $\mathbb{X}$. The Buchberger-Möller (BM) algorithm would return polynomials whose zero sets include the zero-dimensional variety $\mathbb{X}$. Nevertheless such polynomials might be too complex, e.g. of too high degree, to be useful for some practical purposes in some areas such as statistical modelling [10, 11]. For example from the four points $\mathbb{X} = (\pm 1, \pm 1)$, the two polynomials $x_1^2 - 1, x_2^2 - 1$ are the generators of the polynomial ideal of $\mathbb{X}$ and the vector space basis of the quotient space $\mathbb{R}[x_1, x_2]/\langle x_1^2 - 1, x_2^2 - 1\rangle$ returned by the algorithm is $\mathcal{O} = \{1, x_2, x_1, x_1x_2\}$ for any term ordering. Both generators are simple and the model/design matrix (also called evaluation

Received August 9, 2016.

2010 *Mathematics Subject Classification.* Primary: 13P, 62J, 65F. Secondary: 13P10, 65F20.

Key words and phrases. least squares estimates, vanishing ideals, Gröbner bases.

matrix) $M = [s(d)]_{d\in\mathbb{X},s\in\mathcal{O}}$ is not only full rank, but also well conditioned. Nevertheless slightly perturbing the point $(1,1)$ into $(0.9,1.1)$ leads to a non unique nor nice choice of generators, for example a generator set with respect to the DegRevLex term ordering is given by the three polynomials

$$x_1x_2 - 19x_2^2 + x_1 + x_2 + 20,\ x_1^2 + \frac{19}{21}x_2^2 - \frac{40}{21},\ x_2^3 - \frac{11}{10}x_2^2 - x_2 + \frac{11}{10}$$

with corresponding $\mathcal{O} = \{1, x_1, x_2, x_2^2\}$. The model/design matrix is ill-conditioned. In statistics this is known as collinearity of the explanatory functions labelling the columns of M. These functions depend on the variables representing the characteristics of the population under study and of which $\mathbb{X}$ can be considered a collection of instances. Regression models based on $\mathcal{O}$ would be unstable, specifically stability of algorithms commonly used in statistics cannot be guaranteed.

To overcome this, numerical versions of the BM algorithm are available in the literature which take into account the fact that the coordinates of the points in $\mathbb{X}$ might be effected by random error. We discuss some of them in this paper. The general strategy is to consider a sequence of embedded $\mathbb{R}$-vector spaces of increasing dimension $(V_m)_{m=1,2,\ldots}$ such that V_m is a subset of polynomials. At step m the objective is to determine a polynomial $p \in V_m$ such that $\sum_{u\in\mathbb{X}} p^2(u)$ is minimal. A variant requires to find p such that not only this sum is minimised, but also the variety $\{y : p(y) = 0\}$ admits a n-point subset, each point of which can be paired to a point in $\mathbb{X}$ so that the distances between paired points is smaller than a given threshold ε, namely for each $u \in \mathbb{X}$ find a y in $\{y : p(y) = 0\}$ such that $\|u-y\|$ is smaller than ε. In Section 6 p and the y's are chosen so that $\min\{\|u - y\| \mid p(y) = 0\}$ is reached by following a classical algorithm for implicit regression analysis.

The paper is organised as follows. In Section 2 the classical BM algorithm is recalled together with an example illustrating the issues dealt with in the paper. In Section 3 two classical notions of numerical independence of vectors used to avoid collinearity are outlined. The first one is based on singular values and the second one on least squares. The latter leads to the NBM algorithm in Section 4 and tightened into the LDP algorithm (Section 5) which in Section 6 is integrated with the classical Britt-Luecke method for implicit regression from Statistics.

§2. BM algorithm for zero dimensional varieties and noisy data

The BM algorithm [2] is a classical method for computing the Gröbner basis of the ideal $\mathcal{I}(\mathbb{X})$ of a set of points $\mathbb{X} \subset \mathbb{R}^k$ with respect to a term ordering τ. It can be shortly described as follows.

The BM algorithm (BM)

- **Input:** a set of points $\mathbb{X} \subset \mathbb{R}^k$ and a term ordering τ.
- **Output:** the τ-Gröbner basis of $\mathcal{I}(\mathbb{X})$.
- **Core:** stepwise construction of a monomial basis $\mathcal{O}$ of the quotient space $\mathbb{R}[x_1, \ldots, x_k]/\mathcal{I}(\mathbb{X})$
- **Step Zero:** set $\mathcal{O} = \{1\}$.
- **Generic Step:** given $\mathcal{O}$ and $t >_\tau s$ for $s \in \mathcal{O}$, check if the vector $t(\mathbb{X}) = [t(d)]_{d \in \mathbb{X}}$ and the columns of $M(\mathbb{X}) = [s(u)]_{u \in \mathbb{X}, s \in \mathcal{O}}$ are linearly dependent.
  If the answer is **Yes**, the polynomial $g = t - \sum_{s \in \mathcal{O}} \alpha_s s$, where α is s.t. $M(\mathbb{X})\alpha = t(\mathbb{X})$, is added to the Gröbner basis.
  If the answer is **No**, the monomial t is added to $\mathcal{O}$.

The main check of the BM algorithm is strongly effected by data perturbations, since small variations of the coordinates of linear dependent vectors can turn them into linear independent vectors. For this reason small perturbations of the points in $\mathbb{X}$ can correspond to a very different Gröbner bases, as illustrated in the following example.

Example 2.1. The set $\widetilde{\mathbb{X}} = \{(1,1),(2,3),(3,5.1)\}$ is obtained by slightly changing a coordinate of a point in $\mathbb{X} = \{(1,1),(2,3),(3,5)\}$. The DegLex-Gröbner bases $\widetilde{\mathcal{G}}$ and $\mathcal{G}$ of their vanishing ideals are very different, since $\mathcal{G} = \{x - 0.5y - 0.5,\ y^3 - 9y^2 + 23y - 15\}$ and $\widetilde{\mathcal{G}} = \{y^2 - 80x + 36y + 43,\ xy - 45x + 20y + 24,\ x^2 - 26.1x + 11.55y + 13.55\}$ and the corresponding algebraic varieties are very different as well.

In order to avoid this drawback and to obtain an $\mathcal{O}$ set robust with respect to small perturbations, the check of exact linear dependence of vectors can be relaxed and substituted with that of numerical linear dependence. In literature [4, 7, 9] there are several ways for testing the dependence of a set of vectors from the numerical point of view. Exploiting these ideas of numerical dependence of a set of vectors, it is possible to design different numerical versions of the BM algorithm, for which at each step the main check consists in testing whether $t(\mathbb{X})$ is numerically, and not exactly, dependent on the columns of $M(\mathbb{X}) = [s(\mathbb{X})]_{s \in \mathcal{O}}$.

§3. Numerical dependence of vectors

3.1. An approach based on singular value decomposition

A classical strategy for testing the numerical dependence of a set of vectors is based on the Singular Values Decomposition (SVD) [7]. Given a set W of vectors, let $\sigma_1 \geq \dots \sigma_p > 0$ be the singular values of the matrix $A = [v]_{v \in W}$ whose columns are the elements of W and p is the rank of A. Given a threshold ε, if $\sigma_1 \geq \cdots \geq \sigma_r \geq \varepsilon > \sigma_{r+1} \geq \cdots \geq \sigma_p$, then r is called the numerical rank of A w.r.t. ε. The numerical rank gives information about the numerical dependence of the vectors of W. Indeed, it is well known that there exists a rank deficient matrix B such that the rank of B is r and $\|A - B\|_2 \leq \sigma_{r+1} < \varepsilon$, where, for a matrix M, $\|M\|_2$ is its largest singular value. In this case the columns of A can be considered numerically dependent. Some numerical versions of the BM algorithm exploit this notion of numerical dependence, for example the BM Approximation algorithm in [9].

Nevertheless, the main check of the BM algorithm, which involves the monomial t, the set of monomials $\mathcal{O}$ and the matrix $M(\mathbb{X})$, tests if the vector $t(\mathbb{X})$ is a linear combination of the columns of $M(\mathbb{X})$ in order to compute a polynomial vanishing at $\mathbb{X}$. If the points in $\mathbb{X}$ are effected by data errors, then it is of interest to check if there exists a small perturbation $\widetilde{\mathbb{X}}$ of $\mathbb{X}$ such that $g(\widetilde{\mathbb{X}}) = 0$, for some polynomial g with support contained in $\mathcal{O} \cup \{t\}$. For this reason the generic step of the algorithm should test if there exists a set $\widetilde{\mathbb{X}}$ close to $\mathbb{X}$ such that the vectors $\{s(\widetilde{\mathbb{X}})\}_{s \in \mathcal{O} \cup \{t\}}$ are linearly dependent. The computation of the singular values of the matrix $[M(\mathbb{X}), t(\mathbb{X})]$ in the general case does not give such information usually, as illustrated in the following example.

Example 3.1. For the three points set $\mathbb{X} = \{1, 3, 3.1\}$ in $\mathbb{R}$, the set of monomials $\mathcal{O} = \{1, x\}$ and the monomial $t = x^2$, the evaluation matrix is

$$\widehat{M}(\mathbb{X}) = [M(\mathbb{X}), t(\mathbb{X})] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 3.1 & 3.1^2 \end{bmatrix}.$$

For $\varepsilon = 0.03$, since $\{13.9931, 1.0744, 0.0279\}$ are the singular values of $\widehat{M}(\mathbb{X})$ and since $\sigma_2 > \varepsilon > \sigma_3$, the numerical rank of $\widehat{M}(\mathbb{X})$ w.r.t. ε is equal to 2 and thus there exists a rank deficient matrix B such that $\|\widehat{M}(\mathbb{X}) - B\|_2 < \varepsilon$. A possible polynomial almost vanishing at $\mathbb{X}$ is $f(x) = x^2 - 4.0549x + 3.0599$.

Nevertheless, there does not exist any small perturbation $\widetilde{\mathbb{X}}$ of $\mathbb{X}$ such that $t(\widetilde{\mathbb{X}})$ is a linear combination of the columns of $M(\widetilde{\mathbb{X}})$. In fact, for a generic perturbation $\widetilde{\mathbb{X}} = \{1+\delta_1, 3+\delta_2, 3.1+\delta_3\}$, the evaluation matrix is

$$\widehat{M}(\widetilde{\mathbb{X}}) = \begin{bmatrix} 1 & 1+\delta_1 & (1+\delta_1)^2 \\ 1 & 3+\delta_2 & (3+\delta_2)^2 \\ 1 & 3.1+\delta_3 & (3.1+\delta_3)^2 \end{bmatrix}.$$

Since $\det(\widehat{M}(\widetilde{\mathbb{X}})) = (2+\delta_2-\delta_1)(2.1+\delta_3-\delta_1)(0.1+\delta_3-\delta_2)$, does not vanish for any $\delta_i < \varepsilon$, $i = 1,2,3$, it follows that there does not exist any set close to $\mathbb{X}$ by less than ε such that the columns of $\widehat{M}(\widetilde{\mathbb{X}})$ are linearly dependent. In conclusion there do not exist an $\widetilde{\mathbb{X}}$ small perturbation of $\mathbb{X}$ and a polynomial f with support $\{1, x, x^2\}$ such that f vanishes at $\widetilde{\mathbb{X}}$.

3.2. An approach based on least squares method

A different definition of numerical linear dependence is presented in [4] and [5], which formalises and exploits the concept of *admissible perturbation* of $\mathbb{X}$. Given a set of points $\mathbb{X}$ known to be effected by experimental error and an estimation ε of the maximum componentwise error, a set $\widetilde{\mathbb{X}}$ is an admissible perturbation of $\mathbb{X}$ if for each $u \in \mathbb{X}$ there exists $\delta_u \in \mathbb{R}^k$ such that $\|\delta_u\|_\infty = \max_{i=1,\dots,k} |\delta_i| < \varepsilon$ and $\widetilde{\mathbb{X}} = \{u + \delta_u \mid u \in \mathbb{X}\}$.

The sets $\mathbb{X}$ and $\widetilde{\mathbb{X}}$ are indistinguishable from the numerical points of view and so the vector $t(\mathbb{X})$ can be considered linearly dependent on the columns of the matrix $M(\mathbb{X})$ with respect to ε if there exists an admissible perturbation $\widetilde{\mathbb{X}}$ of $\mathbb{X}$ such that $t(\widetilde{\mathbb{X}})$ is exactly linearly dependent on the columns of $M(\widetilde{\mathbb{X}})$. According to this definition, the columns of the matrix $\widehat{M}(\mathbb{X})$ in Example 3.1 are linearly independent w.r.t. $\varepsilon = 0.03$, even if $\widehat{M}(\mathbb{X})$ is numerically rank deficient w.r.t. ε.

Let $\rho(\mathbb{X}) = t(\mathbb{X}) - M(\mathbb{X})\alpha$ be the residual of the least squares problem $M(\mathbb{X})\alpha = t(\mathbb{X})$. Two necessary conditions for $t(\mathbb{X})$ and the columns of $M(\mathbb{X})$ to be numerically dependent are

$$(1) \qquad |\rho(\mathbb{X})| \leq \varepsilon |I - M(\mathbb{X})M^+(\mathbb{X})| \sum_{i=1}^{k} \left| \frac{\partial g}{\partial x_i}(\mathbb{X}) \right| + O(\varepsilon^2) \quad \text{and}$$

$$(2) \qquad \|\rho(\mathbb{X})\|_2^2 \leq \varepsilon^2 \sum_{u \in \mathbb{X}} \|\nabla g(u)\|_2^2 + O(\varepsilon^3),$$

where the absolute value is intended componentwise and $M^+(\mathbb{X}) = (M^t(\mathbb{X})M(\mathbb{X}))^{-1}M^t(\mathbb{X})$ is the pseudo inverse of $M(\mathbb{X})$. The first upper bound is presented, in a slightly different formulation, in [4, Th. 3.5] while the second one is in [5, Prop. 4]. Thus if upper bound (1) or upper bound (2) are not satisfied, then there does not exist any polynomial with support contained in $\mathcal{O} \cup \{t\}$ which vanishes at any admissible perturbation of $\mathbb{X}$. If both upper bounds (1) and (2) are satisfied, we cannot conclude that there is a polynomial with support in $\mathcal{O} \cup \{t\}$ vanishing at some $\widetilde{\mathbb{X}}$. Nevertheless the polynomial $g = t - \sum_{s \in \mathcal{O}} \alpha_s s$ assumes small values at $\mathbb{X}$, since $g(\mathbb{X}) = \rho(\mathbb{X})$, with $\|\rho(\mathbb{X})\|_2 \sim O(\varepsilon)$.

§4. The NBM algorithm

A numerical version of the BM algorithm should check, in its generic step, whether the vector $t(\mathbb{X})$ and the columns of $M(\mathbb{X})$ are linearly dependent from a numerical point of view. Obviously, in the lucky case when a polynomial vanishing on $\mathbb{X}$ is available, no check is required, both in the cases where $\mathbb{X}$ is exactly known or includes noisy data.

For not noisy data, when the residual $\rho(\mathbb{X}) = t(\mathbb{X}) - M(\mathbb{X})\alpha$ of the least squares problem $M(\mathbb{X})\alpha = t(\mathbb{X})$ is different from zero, then the polynomial $g = t - \sum_{s \in \mathcal{O}} \alpha_s s$ does not vanish at $\mathbb{X}$ and thus the BM algorithm inserts t into $\mathcal{O}$.

For noisy data, when neither the upper bound (1) or (2) are satisfied, then the monomial t can be inserted in $\mathcal{O}$ to give a normal set for each admissible perturbation of $\mathbb{X}$. If instead (1) and (2) are both satisfied, then, as already mentioned, a polynomial g almost vanishing at $\mathbb{X}$ can be computed.

Exploiting the above, we present below a slight modification of the Numerical BM algorithm (NBM) in [4], which is a numerical version of the BM algorithm.

The Numerical BM algorithm

- **Input:** A set of points $\mathbb{X}$, an error estimation ε and a term ordering τ.
- **Output:** A set $\mathcal{O}$ of monomials and a set $\mathcal{G}$ of polynomials.
- **Core:** stepwise construction of a set $\mathcal{O}$ of monomials.
- **Step Zero:** Set $\mathcal{O} = \{1\}$.
- **Generic Step:** Given $\mathcal{O}$ and $t >_\tau s$ for $s \in \mathcal{O}$, solve the least squares problem $M(\mathbb{X})\alpha = t(\mathbb{X})$, with $M(\mathbb{X}) = [s(\mathbb{X})]_{s \in \mathcal{O}}$. Check if $\rho(\mathbb{X})$ satisfies upper bound (2).
  If the answer is **No**, the monomial t is added to $\mathcal{O}$.
  If the answer is **Yes**, then check upper bound (1) and
  - if the answer is **No**, the monomial t is added to $\mathcal{O}$;
  - if the answer is **Yes**, $g = t - \sum_{s \in \mathcal{O}} \alpha_s s$ is added to $\mathcal{G}$.

Importantly, the set $\mathcal{O}$ computed by the NBM algorithm is a basis of the quotient space $\mathbb{R}[x_1, \ldots, x_n]/\mathcal{I}(\widetilde{\mathbb{X}})$ for *each* admissible perturbation $\widetilde{\mathbb{X}}$. In fact, if at least one coordinate of $\rho(\mathbb{X})$ does not satisfy the upper bound (1) or if the 2-norm of $\rho(\mathbb{X})$ does not satisfy the upper bound (2), then there are no admissible perturbations $\widetilde{\mathbb{X}}$ of $\mathbb{X}$ such that $\rho(\widetilde{\mathbb{X}}) = 0$, that is such that the vector $t(\widetilde{\mathbb{X}})$ and the columns of $M(\widetilde{\mathbb{X}})$ are dependent. This implies that if $M(\mathbb{X})$ is a full rank matrix then the matrix $[M(\widetilde{\mathbb{X}}), t(\widetilde{\mathbb{X}})]$ is a full rank matrix too, for each admissible perturbation $\widetilde{\mathbb{X}}$ and thus $\mathcal{O}$ is a basis of the quotient space. Furthermore, since by construction, $\mathcal{O}$ is closed under taking divisors, we conclude that $\mathcal{O}$ is a normal set.

Furthermore, if each point $u \in \mathbb{X}$ belongs to $[-1, 1]^k$, each polynomial $g \in \mathcal{G}$ is almost vanishing, that is $\|g(\mathbb{X})\|_2/\|c\|_2 = O(\varepsilon)$, where c is the coefficient vector of g. Such result is shown in [4, Th. 5.1] if the upper bound (1) is used, while it follows from the fact that $\|\nabla g(u)\|_2 \leq \sqrt{v}\sqrt{\sum_{j=1}^{k} D_j^2}\|c\|_2$, where v is the cardinality of the support of g and D_j is the maximum degree of each x_j in g (see Lemma 7.1), monomials of g, if the upper bound (2) is implemented.

In the algorithm, we choose to check first upper bound (2) since it is easier to compute than (1), because it does not require the computation of a pseudo inverse matrix.

§5. The LDP algorithm

Example 5.1. The polynomial $f = y^2 + 0.01x^2 - 1$ has coefficient vector $c = [1, 0.01, 1]$ and it is almost vanishing at the point $u = (12, 0)$ for all $\varepsilon > 0.312$ since $|f(u)|/\|c\|_2 = 0.312$. Nevertheless, the minimum euclidean distance between u and the zero set of f is equal to 2, as shown in Figure 1. Thus even if f is almost vanishing at u, its affine variety does not pass close to u.

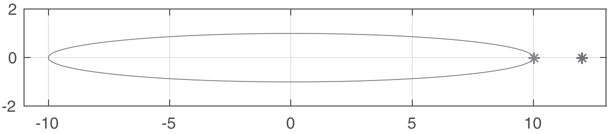

Fig. 1. The variety $Z(f) = \{(x, y) \,|\, y^2 + 0.01x^2 - 1 = 0\}$ does not pass close to $(12, 0)$, w.r.t. $\varepsilon = 0.4$, even if $|f(u)|/\|c\|_2 = 0.312 < \varepsilon$.

Given $\mathbb{X}$ and ε, NBM returns a normal set $\mathcal{O}$ such that $M(\mathbb{X})$ and $M(\widetilde{\mathbb{X}})$ are full rank matrices for all $\widetilde{\mathbb{X}}$ admissible perturbation of $\mathbb{X}$. Moreover, it returns polynomials g such that $g(\mathbb{X})$ is almost zero. Now to workaround the issue in Example 1, the LDP algorithm below seeks an admissible perturbation $\widetilde{\mathbb{X}}$ and a polynomial $\widetilde{g}$ whose coefficients are a perturbation of those of g such that the zero set of $\widetilde{g}$ includes $\widetilde{\mathbb{X}}$. LDP or LPA stands for Low Degree Polynomial algorithm, first presented in [6].

As in NBM algorithm, the main check of LDP analyses the numerical dependence of a set of vectors using upper bounds (1) and (2). But LDP and NBM differ in the last step, where the g polynomials are constructed. In theory, with LDP we want to find a set $\widetilde{\mathbb{X}}$ such that $\rho(\widetilde{\mathbb{X}}) = 0$, i.e. $t(\widetilde{\mathbb{X}}) - M(\widetilde{\mathbb{X}})\alpha(\widetilde{\mathbb{X}}) = 0$. If such $\widetilde{\mathbb{X}}$ is an admissible perturbation of $\mathbb{X}$ then the polynomial $\widetilde{g} = t - \sum_{s\in\mathcal{O}} \alpha_s(\widetilde{\mathbb{X}})s$ is returned and LDP stops. Otherwise, t is added to $\mathcal{O}$. The algorithm ends because in the worst case it computes an element of the exact Gröwner basis of $\mathcal{I}(\mathbb{X})$. The LDP algorithm can be summarised as follows.

The Low Degree Polynomial Algorithm (LDP)

- **Input:** As NBM.
- **Output:** A set $\mathcal{O}$ of monomials, an admissible perturbation $\widetilde{\mathbb{X}}$ and *a* low degree polynomial $\widetilde{g}$ such that $\widetilde{g}(\widetilde{\mathbb{X}}) = 0$.

All equal to NBM with the exception of the last row which changes to

- **Key step:** if the answer is **Yes**, compute a set $\widetilde{\mathbb{X}}$ such that $\rho(\widetilde{\mathbb{X}}) = 0$.
  - If $\widetilde{\mathbb{X}}$ is an admissible perturbation of $\mathbb{X}$, the polynomial $\widetilde{g} = t - \sum_{s\in\mathcal{O}} \alpha(\widetilde{\mathbb{X}})_s s$ is formed.
  - Otherwise, the monomial t is added to $\mathcal{O}$.

In practice, solving $\rho(\widetilde{\mathbb{X}}) = 0$ is a difficult task. It can be reformulated introducing a vector of additive errors $e = (e_u)_{u\in\mathbb{X}}$, where each e_u is a vector whose coordinates represent the perturbation of the point u in $\mathbb{X}$. Since each point $\widetilde{u}$ belonging to a generic admissible perturbation $\widetilde{\mathbb{X}}$ of $\mathbb{X}$ can be expressed as $\widetilde{u} = u + e_u$, then $\widetilde{\mathbb{X}}$ and also $\rho(\widetilde{\mathbb{X}})$ can be expressed in function of e. To keep track of this, we denote a generic perturbation of $\mathbb{X}$ by $\mathbb{X}(e)$ and the residual of the least squares problem $M(\mathbb{X}(e))\alpha(\mathbb{X}(e)) = t(\mathbb{X}(e))$ by $\rho(e) = t(e) - M(e)\alpha(e)$.

In this formulation, to solve $\rho(\widetilde{\mathbb{X}}) = 0$ is equivalent to find a vector $\widetilde{e}$ such that $\rho(\widetilde{e}) = 0$ and this gives $\widetilde{\mathbb{X}} = \mathbb{X}(\widetilde{e})$. Since the exact solution of a nonlinear system of equations can be found in very few cases, LDP does not compute $\widetilde{e}$, but an approximation of $\widetilde{e}$. Obviously, different techniques can be adopted to approximate $\widetilde{e}$. The original version of the LDP, presented in [6], finds an approximation $\widehat{e}$ of $\widetilde{e}$ by means of a root finding method based on the Normal Flow Algorithm [14], a classical iterative method for approximating a solution of a non linear system. Section 6 presents an alternative method. Theorem 4.2 in [6] shows that the polynomial $\widehat{g}$ computed by the LDP algorithm is almost vanishing at $\mathbb{X}(\widehat{e})$ even if it does vanish at it. Furthermore, if $\widehat{g}$ satisfies some simple conditions similar to the hypotheses of the Kantorovich theorem on the convergence of Newton method, then Theorem 4.3 in [6] shows that the zero set of $\widehat{g}$ lies close to $\mathbb{X}$ by less than ε, that is $\widehat{g}$ vanishes at an admissible perturbation of $\mathbb{X}$.

Example 5.2. In the left hand side plot of Figure 2, for the set of points

$$\mathbb{X} = \{(0.95, 1), (5.05, 2.95), (5.05, -0.95), (9.98, 4), (10.05, -2), (17.01, 5)\},$$

obtained by perturbing less than 0.1 the coordinates of six points on $f = y^2 - x - 2y + 2 = 0$, the dot-dashed curve is the zero set of the minimal degree polynomial in the Gröbner basis returned by the BM algorithm with DegRevLex ordering; the dashed curve is the zero set of the minimal degree polynomial returned by the NBM algorithm with $\varepsilon = 0.1$ and the black curve is the zero set of the polynomial returned by the LDP algorithm with the same ε. The three polynomials with the coefficients rounded to the first 4 decimal digits are

$$\begin{aligned} f_{BM} &= y^3 + 0.012541x^2 - 1.0439xy - 4.1372y^2 + 2.0233x \\ &\quad + 6.7437y - 4.5483 \\ f_{NBM} &= y^2 - 1.004x - 2.009y + 2.128 \\ f_{LDP} &= y^2 - 0.975x - 2.005y + 1.977 \end{aligned}$$

The right hand side plot of Figure 2 zooms around the vertex of the parabola. The polynomials f, f_{NBM} and f_{LDP} have the same support and, in this case, denoting with α, α_{NBM} and α_{LDP} the coefficients of f, f_{NBM} and f_{LDP}, respectively, we have that

$$\frac{\|\alpha - \alpha_{NBM}\|_2}{\|\alpha\|_2} = \frac{\|[0, 0.004, 0.009, -0.128]\|_2}{\|[1, -1, -2, 2]\|_2} = 0.0406 < \varepsilon$$
$$\frac{\|\alpha - \alpha_{LDP}\|_2}{\|\alpha\|_2} = \frac{\|[0, -0.025, -0.005, 0.023]\|_2}{\|[1, -1, -2, 2]\|_2} = 0.0109 < \varepsilon$$

By construction, there are six points in the zero set of f_{LDP} which are an admissible perturbation of $\mathbb{X}$, specifically they are

$$\begin{aligned} \widetilde{\mathbb{X}} = \{&(0.9969, 1),\ (5.0892, 3),\ (5.0688, -0.99),\ (9.9680, 3.96), \\ &(10.0592, -1.97),\ (17.0603, 4.96)\}. \end{aligned}$$

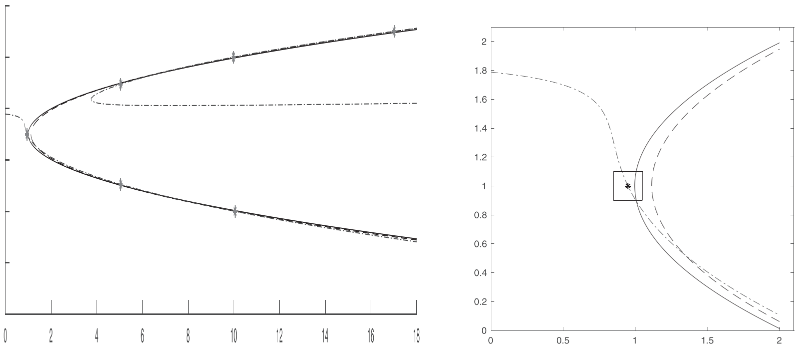

Fig. 2. The left plot shows the polynomials computed by BM (−-−), NBM (− − −) and LDP (black). The right plot shows around the point closest to the origin.

In summary, the vector $\widehat{e}$ found above is such that $\mathbb{X}(\widehat{e})$ is an admissible perturbation of $\mathbb{X}$ and it is an approximation of the zero set of ρ. By further assuming normally distributed random error with zero mean and known covariance matrix, the algorithm could be adapted so that $\widehat{e}$ is the solution a least squares problem as shown in Section 6 and a maximum likelihood problem.

§6. The LDP-BL algorithm

For the long vectors $z = (u+e_u)_{u\in\mathbb{X}}$, $e = (e_u)_{u\in\mathbb{X}}$ and a vector function $f(z,\theta)$ on the noisy data z and depending on a parameter vector θ, the problem of finding a solution to the following constrained minimisation problem: $\min_e\{e^tR^{-1}e\}$ subject to $f(z,\theta)=0$, is the subject of [1]. The authors assume that the measurement error e is normally distributed with zero mean and known covariance matrix R. The solution to the constrained minimisation problem is thus the least squares estimate of θ which can be shown to be the maximum likelihood estimate when the model is given in implicit form ($f=0$).

Under some regularity assumptions on f the authors of [1] propose an iterative method based on Lagrange multipliers. Let $z_0 = (u)_{u\in\mathbb{X}}$, z_c and θ_c be the current estimates of z and θ, F_θ and F_z be the Jacobian matrix of $f(z,\theta)$ with respect to θ and to z evaluated at z_c and θ_c, respectively. The vectors θ and z are updated as $\theta+\delta\theta$ and $z+\delta z$ where

$$\delta\theta = -\left(F_\theta^t(F_zRF_z^t)^{-1}F_\theta\right)^{-1}F_\theta^t(F_zRF_z^t)^{-1}\left(f(z_c,\theta_c)+F_z\cdot(z_0-z_c)\right)$$
$$\delta z = -RF_z^t(F_zRF_z^t)^{-1}\left(f(z_c,\theta_c)+F_\theta\cdot\delta\theta+F_z\cdot(z_0-z_c)\right)$$

A new step is performed until $\|\delta\theta\|_2$ and $\|\delta z\|_2$ satisfy a smallness criterion. Let us call this the Britt-Luecke algorithm.

It can be implemented inside the **key step** of the LDP algorithm, where the implicit function $f(z,\theta)$ is the evaluation at an admissible perturbation $\mathbb{X}(e)$ of the polynomial $g = t - \sum_{s\in\mathcal{O}} \alpha_s s$, namely $\rho(\mathbb{X}(e))$ and the parameter vector θ is the coefficient vector of g. At each step the Jacobian matrix F_θ coincides with the evaluation matrix of the monomials in $\mathcal{O}$ at the current set of points derived from z and the updating steps are rather simple as the function is a polynomial. We call the LDP algorithm with the strategy due to Britt and Luecke for approximating the zeros of ρ the LDP-BL algorithm.

In general, the admissible perturbations of $\mathbb{X}$ computed by the LDP and the LDP-BL algorithms are different and the set of points computed by the LDP-BL algorithm is closer to $\mathbb{X}$ than the set computed by the LDP algorithm, since the Britt-Luecke procedure minimises the 2-norm of the vector e. Example 6.1 illustrates the different behaviour of the LDP and of the LDP-BL algorithm.

Example 6.1. For $\varepsilon = 0.55$, R the identity matrix and the ten point set

$$\begin{aligned}\mathbb{X} = \{&(1,2),\ (0.5,1.3),\ (1.98,2.05),\ (0,2.08),\ (-0.48,3.18),\\ &(2.95,5.05),\ (-0.95,5.05),\ (-1.45,7.2),\ (4,9.98),\ (-2,10.05)\},\end{aligned}$$

the NBM algorithm computes $g = y - 0.9668x^2 + 1.9308x - 2.2780$, which is almost vanishing at $\mathbb{X}$, since $\|g(\mathbb{X})\|_2 / \|[1, -0.9668, 1.9308, -2.2780]\|_2 = 0.2527 < \varepsilon$. But g does not vanish at any admissible perturbation of the point $(1,2)$, since the zero set of g does not intersect the ball with centre in $(1,2)$ and radius ε (see Figure 3).

The original LDP algorithm computes $g_1 = y - 0.9657x^2 + 1.8292x - 2.3703$ and the admissible perturbation

$$\begin{aligned}\mathbb{X}_1 = \{&(1.007,1.508),\ (0.719,1.554),\ (1.772,2.162),\ (0.129,2.150),\\ &(-0.385,3.218),\ (2.869,5.070),\ (-0.967,5.044),\ (-1.480,7.192),\\ &(3.913,9.997),\ (-2.027,10.045)\}\end{aligned}$$

while the LDP-BL algorithm computes the polynomial $g_2 = y - 0.9467x^2 + 1.7542x - 2.2715$ and the admissible perturbation

$$\begin{aligned}\mathbb{X}_2 = \{&(1.301,1.592),\ (0.630,1.542),\ (1.789,2.163),\ (0.083,2.132),\\ &(-0.429,3.200),\ (2.879,5.069),\ (-1.016,5.032),\ (-1.532,7.182),\\ &(3.929,9.992),\ (-2.083,10.03)\}.\end{aligned}$$

The polynomials g_1 and g_2 have similar behaviour: they are almost vanishing at $\mathbb{X}$ since

$$\frac{\|g_1(\mathbb{X})\|_2}{\|[1, -0.9657, 1.8292, -2.3703]\|_2} = 0.3372 < \varepsilon$$

$$\frac{\|g_2(\mathbb{X})\|_2}{\|[1, -0.9467, 1.7542, -2.2715]\|_2} = 0.3694 < \varepsilon,$$

they do not vanish at $\mathbb{X}_1$ and $\mathbb{X}_2$ exactly, nevertheless they take very small values at $\mathbb{X}_1$ and $\mathbb{X}_2$ as $\|g_1(\mathbb{X}_1)\|_2 = 0.00024$ and $\|g_2(\mathbb{X}_2)\|_2 = 0.00039$ and their zero sets contain admissible perturbations of $\mathbb{X}$ (see Figure 3). However note that the sum of squared euclidean distances between associated pairs of points in $\mathbb{X}$ and $\mathbb{X}_2$ is 0.6512 which is smaller than 0.6782, the analogue quantity for $\mathbb{X}$ and $\mathbb{X}_1$, because the LDP-BL algorithm aims at minimising this quantity.

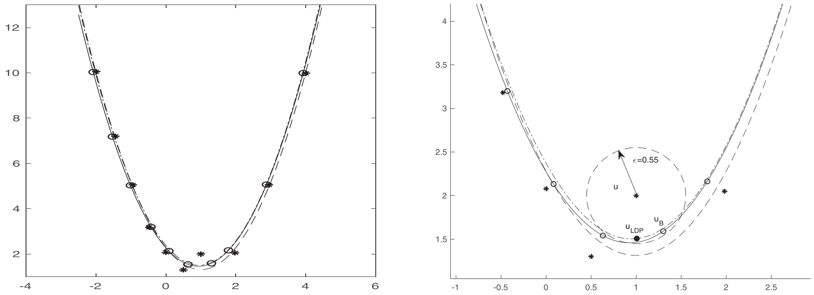

Fig. 3. The left plot shows the polynomials computed by NBM (dashed curve), LDP (dot-dashed curve) and LDP-BL (black curve). The right plot zooms the polynomials around the vertex. The LDP algorithm moves from the original point u to u_{LDP}, while the LDP-BL algorithm moves from u to u_B.

§7. Properties of almost vanishing polynomials

Let f be a monic polynomial almost vanishing at $\mathbb{X}$. If there exists a monic polynomial $\widetilde{f}$ with the same support as f and vanishing at an admissible perturbation $\widetilde{\mathbb{X}}$ of $\mathbb{X}$, then Theorem 7.1 shows that f and $\widetilde{f}$ have similar coefficients.

Theorem 7.1. *Let $\mathbb{X}$ be a set of points, ε a data error estimation, $\mathcal{O}$ a finite set of monomials, t a monomial not in $\mathcal{O}$, $M(\mathbb{X}) = [s(u)]_{u \in \mathbb{X}, s \in \mathcal{O}}$*

be full rank and $t(\mathbb{X}) = [t(u)]_{u\in\mathbb{X}}$. *Let* $f = t - \sum_{s\in\mathcal{O}} \alpha_s s$ *be a polynomial whose coefficient vector solves the least squares problem* $M(\mathbb{X})\alpha = t(\mathbb{X})$, *and let* $\widetilde{f} = t - \sum_{s\in\mathcal{O}} \widetilde{\alpha}_s s$ *be a polynomial vanishing at an admissible perturbation* $\widetilde{\mathbb{X}}$ *of* $\mathbb{X}$. *Then*

$$(3) \qquad \|\alpha - \widetilde{\alpha}\|_2 \leq \frac{1}{\sigma}\left(\|f(\mathbb{X})\|_2 + \varepsilon\sqrt{\sum_{u\in\mathbb{X}} \|\nabla\widetilde{f}(u)\|_2^2 + O(\varepsilon^2)}\right)$$

where σ *is the smallest singular value of* $M(\mathbb{X})$.

Proof. Let x be a non zero vector in $\mathbb{R}^{\#\mathcal{O}}$. By the min-max theorem, the smallest eigenvalue λ of $M(\mathbb{X})^t M(\mathbb{X})$ is such that $\lambda = \min_{x\neq 0} \frac{\|M(\mathbb{X})x\|_2^2}{\|x\|_2^2}$. Since the square roots of the eigenvalues of $M(\mathbb{X})^t M(\mathbb{X})$ are the singular values of $M(\mathbb{X})$, the smallest singular value σ of $M(\mathbb{X})$ is equal to $\min_{x\neq 0} \frac{\|M(\mathbb{X})x\|_2}{\|x\|_2}$.

For each $x \neq 0$, we have $\|x\|_2 \leq \frac{\|M(\mathbb{X})x\|_2}{\sigma}$ and in particular

$$\begin{aligned}
\|\alpha - \widetilde{\alpha}\|_2 &\leq \frac{1}{\sigma}\|M(\mathbb{X})(\alpha - \widetilde{\alpha})\|_2 = \frac{1}{\sigma}\|M(\mathbb{X})\alpha - M(\mathbb{X})\widetilde{\alpha}\|_2 \\
&= \frac{1}{\sigma}\|M(\mathbb{X})\alpha - t(\mathbb{X}) + t(\mathbb{X}) - M(\mathbb{X})\widetilde{\alpha}\|_2 \\
&= \frac{1}{\sigma}\|f(\mathbb{X}) - \widetilde{f}(\mathbb{X})\|_2 \leq \frac{1}{\sigma}(\|f(\mathbb{X})\|_2 + \|\widetilde{f}(\mathbb{X})\|_2)\ .
\end{aligned}$$

The thesis follows since $\widetilde{f}$ vanishes at $\widetilde{\mathbb{X}}$ and so, from Proposition 4 in [5], $\|\widetilde{f}(\mathbb{X})\|_2^2 \leq \varepsilon^2 \sum_{u\in\mathbb{X}} \widetilde{M}_u^2$, where $\widetilde{M}_u^2 = \|\nabla\widetilde{f}(u)\|_2^2 + O(\varepsilon^2)$. Q.E.D.

Sufficient conditions for the existence of $\widetilde{f}$ vanishing at $\widetilde{\mathbb{X}}$ are shown in Theorem 4.3 in [6].

Lemma 7.1 below gives a rough estimate of the sum in the square root of Equation (3).

Lemma 7.1. *Let* $p = \sum_{s\in\mathcal{T}} \beta_s s \in \mathbb{R}[x_1, \ldots, x_k]$ *be a polynomial with coefficients vector* β, v *the cardinality of the support of* p, D_j *the maximum degree of* x_j *in* p *(for* $j = 1, \ldots, k$*) and* $u = (u_1, \ldots, u_k) \in [-1, 1]^k$, *then*

$$\left(\frac{\|\nabla p(u)\|_2}{\|\beta\|_2}\right)^2 \leq v\sum_{j=1}^{k} D_j^2.$$

Proof. For a monomial $s = x_1^{n_1} \cdots x_k^{n_k}$, it holds $\nabla s(u) = [n_j u_1^{n_1} \cdots u_j^{n_j - 1} \cdots u_k^{n_k}]_{j=1,\dots,k}$ and for $u \in [-1,1]^k$ it holds

$$\|\nabla s(u)\|_2^2 = \sum_{j=1}^{k} n_j^2 u_1^{2n_1} \cdots u_j^{2(n_j-1)} \cdots u_k^{2n_k} \leq \sum_{j=1}^{k} n_j^2 \leq \sum_{j=1}^{k} D_j^2.$$

Since $\nabla p(u) = \sum_{s \in \mathcal{T}} \beta_s \nabla s(u)$, it holds

$$\|\nabla p(u)\|_2 \leq \sum_{s \in \mathcal{T}} |\beta_s| \, \|\nabla s(u)\|_2 \leq \sum_{s \in \mathcal{T}} |\beta_s| \sqrt{\sum_{j=1}^{k} D_j^2}$$

and the thesis follows since $\sum_{s \in \mathcal{T}} |\beta_s| = \|\beta\|_1 \leq \sqrt{v} \|\beta\|_2$. Q.E.D.

Now, substituting in Equation (3) and dividing by $\|\widetilde{\alpha}\|_2$ yields

$$\frac{\|\widetilde{\alpha} - \alpha\|_2}{\|\widetilde{\alpha}\|_2} \leq \frac{1}{\sigma} \left(\frac{\|f(\mathbb{X})\|_2}{\|\alpha\|_2} \frac{\|\alpha\|_2}{\|\widetilde{\alpha}\|_2} + \varepsilon \sqrt{n(m+1) \sum_{j=1}^{k} D_j^2 + \frac{O(\varepsilon^2)}{\|\widetilde{\alpha}\|_2^2}} \right)$$

indeed $v = \#\mathcal{O} + 1$ is the cardinality of $\mathcal{O}$ plus one. If f is almost vanishing at $\mathbb{X}$ w.r.t. ε, namely $\frac{\|f(\mathbb{X})\|_2}{\|\alpha\|_2} = O(\varepsilon)$, for example if it has been computed with NBM or LDP algorithms, and since $\|\widetilde{\alpha}\|_2 \geq 1$ because the leading coefficient of $\widetilde{f}$ is one, then

$$\frac{\|\widetilde{\alpha} - \alpha\|_2}{\|\widetilde{\alpha}\|_2} \leq \frac{1}{\sigma} \left(O(\varepsilon) \left(\frac{\|\widetilde{\alpha} - \alpha\|_2}{\|\widetilde{\alpha}\|_2} + 1 \right) + \varepsilon \sqrt{n(m+1) \sum_{j=1}^{k} D_j^2 + O(\varepsilon^2)} \right)$$

equivalently

$$\frac{\|\widetilde{\alpha} - \alpha\|_2}{\|\widetilde{\alpha}\|_2} \left(1 - \frac{O(\varepsilon)}{\sigma} \right) \leq \frac{O(\varepsilon)}{\sigma} + \frac{\varepsilon}{\sigma} \sqrt{n(m+1) \sum_{j=1}^{k} D_j^2 + O(\varepsilon^2)}$$

In our typical applications σ is order of magnitude larger than ε, $\mathcal{O}$ does not involve many terms and the D_j's are small integers, e.g. $0, 1, 2$. Thus the right hand side is small showing that the coefficients of the polynomials obtained with NBM, LDP, LDP-Britt are similar. Examples 5.2 and 6.1 show this in details.

§8. Numerical aspects

The test for numerical independence of the vector $t(\mathbb{X})$ and the columns of the matrix $M(\mathbb{X})$ in the generic step of the NBM and LDP algorithms is performed by computing the residual $\rho(\mathbb{X})$ of the least squares problem $M(\mathbb{X})\alpha = t(\mathbb{X})$. If $t(\mathbb{X})$ is numerically independent of the columns of $M(\mathbb{X})$, then t is included into $\mathcal{O}$ and, at the next step, a new monomial $\widehat{t}$ and a new least squares problem $\widehat{M}(\mathbb{X})\alpha = \widehat{t}(\mathbb{X})$ are considered, where the coefficient matrix $\widehat{M}(\mathbb{X})$ is obtained augmenting $M(\mathbb{X})$ with the column vector $t(\mathbb{X})$ as the last column. In essence, the NBM and LDP algorithms are based on a sequence of least squares problems where, at each step, the coefficient matrix is an update of the coefficient matrix at the previous step.

Using the QR decomposition of the coefficient matrices, the residual at the current step can be obtained exploiting the computation performed at the previous step, without solving explicitly the least squares problem. As shown in [5, Th. 3], if at the previous step $M(\mathbb{X})$ is an $n \times m$ matrix, $n \geq m$, and if $M(\mathbb{X}) = QR$ is its QR decomposition, where Q is an orthogonal $n \times n$ matrix and R is an upper triangular $n \times m$ matrix with positive diagonal entries, then the QR decomposition of $\widehat{M}(\mathbb{X}) = [M(\mathbb{X}), t(\mathbb{X})]$ at the current step is such that the first m columns of $\widehat{Q}$ coincide with the first m columns of Q, the $(m+1)$-th column of $\widehat{Q}$ is $\frac{\rho(\mathbb{X})}{\|\rho(\mathbb{X})\|_2}$, where $\rho(\mathbb{X})$ is the residual vector at the current step, and nothing can be said on the remaining columns of $\widehat{Q}$. The first m columns of $\widehat{R}$ are given by R and the first m elements of the last column of $\widehat{R}$ are given by the first m elements of $R\alpha$, the $(m+1)$-th element is given by $\|\rho(\mathbb{X})\|_2$ and the remaining positions of the last column are filled with zeros.

Thus for evaluating the residual $\rho(\mathbb{X})$ at the current step, it is sufficient to update the QR decomposition of $M(\mathbb{X}) = QR$ for obtaining the QR decomposition of $[M(\mathbb{X}), t(\mathbb{X})] = \widehat{Q}\widehat{R}$. If the last diagonal element of $\widehat{R}$, which is equal to $\|\rho(\mathbb{X})\|_2$, does not satisfy the upper bound (1) or the upper bound (2), then t is added to $\mathcal{O}$. Otherwise, both NBM and LDP algorithms compute a polynomial f whose coefficient vector α is the solution of the linear system whose coefficient matrix consists of the first m rows of R and whose right hand side is given by the first m elements of the last column of $\widehat{R}$.

A stable algorithm with low computational cost for updating the QR decomposition of $M(\mathbb{X})$ consists in the computation, at each step, of a Householder matrix H such that the last $(n - m - 1)$ elements of

$HQ^t t(\mathbb{X})$ are equal to zero. It follows that $HQ^t t(\mathbb{X})$ is the last column of $\widehat{R}$ and that HQ is the matrix $\widehat{Q}$ [7].

§9. Applications

The above algorithms have been employed for a number of analyses of data sets of various complexity for which standard techniques did not yield satisfactory results. For example they have been employed as part of a real time statistical classifier used in colour recognition [3]; as a tool to determine identifiable polynomial regression models from non-standard, noisy experiments used in a thermal spraying process producing particle coatings of surfaces [11]; in marine robotics for the approximation, through a polynomial curve, of the target path an unmanned marine vehicle should follow [12, 13]; for the modelling the trajectories of near-Earth asteroids [8]. Some of these applications have peculiarities which make a numerical version of the BM algorithm more suitable than another one.

9.1. Colour recognition

The first application we consider concerns the problem of colour recognition. Training surgeons can practice by collecting little scattered objects, i.e. coloured chickpeas, inside a laparoscopic training box, and group them according to their colour by looking at their movements in a monitor screen. The performance of the student is assessed in realtime through a computer system. This requires accurate and instantaneous colour identification of images such as the one shown in the left plot of Figure 4. Various sources of uncertainty and subjectivity effect the identification. For example colour is a personal interpretation of the reflected light and are influenced by random factors (light, texture, position, temperature, humidity, etc.), also objects reflect different colours when exposed to different sources of light.

In [3], the problem of colour recognition is addressed by applying a semi-parametric algebraic approach to statistical classification, based on a polynomial approximation to probability density functions. The left plot in Figure 4 is mapped into the right one using the CIELAB model [15] and for each colour 500 points are sampled. Out of these 500 points a subset representing the boundary is extracted, e.g. by using a method based on the Mahalanobis distance. This subset is the $\mathbb{X}$ for a colour given in input to the NBM algorithm. The output of NBM is a polynomial p used to construct a probability density function supported on the interior of the “chickpea”.

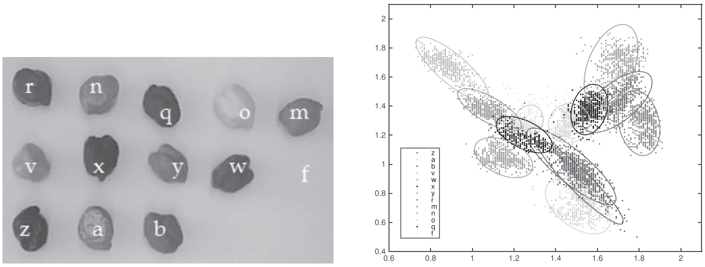

Fig. 4. Image of and representation on the colour system CIELAB.

The objective is thus to determine a smooth approximation p to the boundary of a closed bounded two-dimensional set. To ensure that p is a polynomial whose zero set is indeed a closed curve, only degree two polynomials are used in NBM. As the estimation of the probability densities supported on the clusters were robust to slight misclassification of the cluster boundaries, the precision given by LDP was not paying off for extra computational cost even if the algorithm was run only once for each colour. Anyway we tried and verified that the polynomials returned by the LDP and NBM algorithm were leading to 'ellipses' whose differences were not effecting the classification.

9.2. Industrial coating of surfaces

This project was carried out under the umbrella of the collaborative research center SFB823 of Dortmund University and published in [11]. Its objective was on determining a large set of $\mathcal{O}$ sets, called numerical fan, all of which were identifiable with observations or predictions from an experimental design. Thus, contrary to the standard set up in linear regression where outputs Y are observed at inputs X and a linear model f is sought in the form $Y = f(X) +$ random error, here the data generating process is from X to Z via a third set Y of observables which are easier to measure than Z. The interest is in determining a linear model g either from Y to Z or from X to Z via Y.

A regular design is often chosen on the controlled variables X which induces some canonical regression model f, but the regularity of the design on X is not transferred into Y because either the observed Y or the predicted $f(X)$ are affected by random error. The instability in the observed or predicted designs Y can naturally be dealt with some numerical version of the BM algorithms. Varying the term-ordering in the NBM algorithms yields a finite (possibly large depending on the coordinates of the Y) collection of $\mathcal{O}$ sets. As the issue is in avoiding collinearity any of NBM, LDP, LDP-BL could be safely employed.

In the specific application described in [11], related to a thermal spraying application, the design on the control variables X was a full factorial with central point on four variables in two levels, for a total of 17 points. An identifiable, robust to misspecification, linear regression model f from X to Y is supported on the set of square free monomials in four variables of degree not larger than four plus a simple quadratic term. Either the 17 values $\hat{\mathbb{Y}} = f(\mathbb{X})$ or the observed values at $\mathbb{Y}$ of the in-flight particles properties, are input to the NBM algorithm. In either cases the $\mathbb{Y}$ sets is a seemingly random design. By varying the term-ordering a set of 72 possible identifiable models is obtained. Each of the 72 possible identifiable models is used as maximal model in a forward backward selection based on the AIC criterion to identify a robust model for the coating properties Z in function of Y.

A leave-one-out cross validation analysis was performed based on the PRESS statistics in order to compare with standard methods. It showed the worthwhileness of the computational effort required to compute (a part of) the numerical fan of a design for model search.

9.3. Performance assessment in marine robotics

A topic of interest in marine robotics is the evaluation of the performance on Unmanned Marine Vehicles (UMVs) during a path-following or a path-tracking exercise based solely on the knowledge of the positions of the vehicle in time and of the reference target. Both reference and actual positions are give as sets of points $\mathcal{R}$ and $\mathcal{V}$ respectively, in a two or three dimensional space according to whether the vehicle is an Unmanned Surface Vehicle (USV) or an Unmanned Underwater Vehicle.

In [12, 13] a new criterion for evaluating the capability of an USV to follow a desired generic curvilinear path is presented and tested based on the LDP/LPA algorithm. By using the LDP algorithm a polynomial approximation f to $\mathcal{R}$ is computed with respect to a certain tolerance ε_1 and the value $f(p)$ is used to test whether the point $p \in \mathcal{V}$ is closed to $\mathcal{R}$. This test requires that f is near to $\mathcal{R}$ in the L_2-norm. This requirement is the reason why the LDP algorithm (and ideally its LPD-Britt version) is used and why the NBM algorithm, which only guarantees values of f small in a region around $\mathcal{R}$, might give misleading results. Other norms can be used and are discussed in [13], although the Euclidean norm seems a natural choice for the context.

The measure of closeness is quantified in two bounds called B_1 and B_2 whose formulæ are provided in [13] and which depend on the Jacobian vector and Hessian matrix of f. If $|f(p)| > B_1$ then the curve $f = 0$

does not cross an L_2-ball centred in $p \in \mathcal{V}$ and of radius ε_2 chosen by the experimenter; if $|f(p)| < B_2$ then it crosses it; otherwise it cannot be decided. The associated performance indices are the percentage of points in $\mathcal{V}$ crossing the ball and the percentage of points far from the path. Both indices can be computed efficiently online.

In summary, the requirements of a good estimation of the error, of high precision and of a speedy test based on close-by-ness, make the applicative scenarios of this section a good test bed for the LPD algorithm when used real time and online. The method proposed in [13] to evaluate when an UMV is close to the reference path is thus

(1) computation of an algebraic curve $f = 0$ that approximates the points in $\mathcal{R}$ within a tolerance ε_1;
(2) identification of the points in $\mathcal{V}$ far from the reference path $f = 0$ for more than a tolerance ε_2.

9.4. Near-Earth asteroid modelling

In [8] the trajectories of near-Earth asteroids (NEA), that is the whole set of heliocentric orbital elements with perihelion distance $q \leq 1.3\,au$ and eccentricity $e \leq 1$, are considered with the aim of studying their distances from the trajectory of the Earth. The authors analyse the distribution of the NEA asteroids with respect to some of their orbital elements. Their reconstruction is based on analytical knowledge of the phenomenon and consists of the computation of the minimum of specific analytic functions which at times can be hard to compute analytically.

We take a different perspective, not dissimilar to the one in Section 9.3, and starting from the available data we construct an implicit regression model for the distribution of the NEA asteroids. We have 300 noisy points in the two variables (q, ω) which we write as $\{(q, y) \,|\, y = \cos\omega,\, \omega \in [0, 2\pi]\} \subset \mathbb{R}^2$. The variable q is the perihelion distance and ω is the perihelion argument of a set of NEAs with absolute magnitude $H \geq 26$, the measurement error is estimated to be $\varepsilon = 0.002$.

The NBM algorithm can be applied for obtaining information about the distribution of NEAs directly from some measured parameters, without requiring, for each selected orbital parameters, the solution of a specific different analytical problem as it is done in [8]. The 300 input points have to be preprocessed in order to obtain a much smaller set of points excluding those in low density areas of the (q, ω) plane and taking representatives for high density areas (see Figure 5). This gives a set of 21 points along a curve which exhibits symmetries and singularities. Only the seven points for which $\omega \in [0, \pi/2]$ are used as input in the NBM algorithm which returns the polynomial $g(q, y) = q - 0.306687y - 0.708222$, and thus $g(q, \omega) = q - 0.306687\cos(\omega) - 0.708222$. By reflection and translation of g, one obtains the red (non affine) curve in Figure 5 which almost vanishes at the selected 21 points and is very close to the curve $\gamma(q,\omega) = 0$ presented in [8].

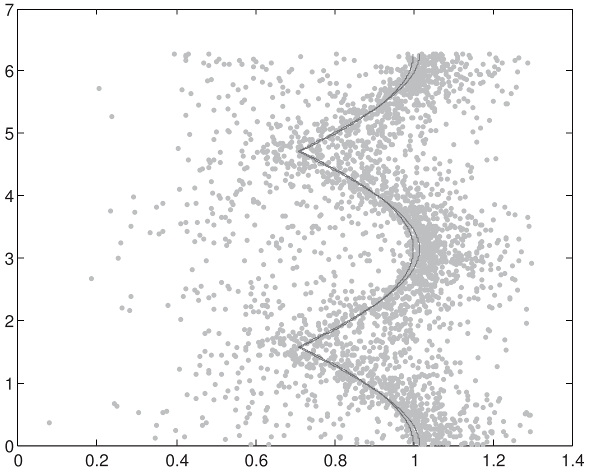

Fig. 5. The set of points $\mathbb{X}$, the (red) curve $g(q,\omega) = 0$ and the (blue) curve $\gamma(q,\omega) = 0$

References

[1] H. I. Britt and R. H. Luecke (1973). *The Estimation of Parameters in Nonlinear*, *Implicit Models.* Technometrics, 15 (2), 233–247.

[2] B. Buchberger and H. M. Möller (1982). *The construction of multivariate polynomials with preassigned zeros.* Proc. EUROCAM '82, LNCS, 144, 24–31.

[3] C. Cuevas Covarrubias, C. Fassino, E. Riccomagno and C. Villar-Patiño (2016). *Probability density functions on star domains with an application to classification.* (Submitted)

[4] C. Fassino (2010). *Almost vanishing polynomials for sets of limited precision points.* J. Symbolic Comput., 45 (1), 19–37.

[5] C. Fassino and H. M. Möller(2016). *Multivariate polynomial interpolation with disturbed data.* Numerical Algorithms, 71 (2), 273–292.

[6] C. Fassino and M.-L. Torrente (2013). *Simple varieties for limited precision points.* Theoretical Computer Science, 479, 174–186.

[7] G. H. Golub and C. F. Van Loan (1996). *Matrix Computations. 3rd Ed.*, John Hopkins University Press, Baltimore.

[8] G. F. Gronchi and G. B. Valsecchi (2013). *On the possible values of the orbit distance between a near-Earth asteroid and the Earth.* Monthly Notices of the Royal Astronomical Society, 429, 2687–2699.

[9] D. Heldt, M. Kreuzer, S. Pokutta, H. Poulisse (2009). *Approximate computation of zero-dimensional polynomial ideals.* J. Symbolic Comput., 44, 156–1591.

[10] G. Pistone, E. Riccomagno and H. P. Wynn (2001). Algebraic statistics: computational commutative algebra in statistics, Volume 89 of *Monographs on Statistics and Applied Probability.* Chapman & Hall/CRC, Boca Raton.

[11] N. Rudak, S. Kuhnt and E. Riccomagno (2016). *Numerical algebraic fan of a design for statistical model building.* Statistica Sinica, 26(3), 1021–1035.

[12] E. Saggini, M.-L. Torrente, E. Riccomagno, M. Bibuli, G. Bruzzone, M. Caccia and E. Zereik (2014). *Assessing path-following performance for Unmanned Marine Vehicles with algorithms from Numerical Commutative Algebra.* MED'14, 22nd Mediterranean Conference on Control and Automation, 16–19 June, Palermo, Italy.

[13] E. Saggini and M.-L. Torrente (2016). *An Euclidean norm based criterion to assess robots' 2D path-following performance.* Journal of Algebraic Statistics, 7, 45–71.

[14] H. F. Walker and L. T. Watson (1990). *Least-change secant update methods for underdetermined systems.* SIAM J. Numer. Anal., 27 (5), 1227–1262.

[15] Carmen Villar Patiño, Carlos Cuevas Covarrubias (2016). *Controlled condensation in k-NN and its application for real time color identification.* Revista de Matemática: teoriá y aplicationes, 23(1) : 143–154 (in Spanish).

(Claudia Fassino) *Department of Mathematics, University of Genova,*
via Dodecaneso 35, 16146 Genova, Italy
E-mail address: `fassino@dima.unige.it`

(Hans Michael Möller) *Universität Dortmund,*
Institut für Angewandte Mathematik,
Vogelpothsweg 87, D-44221 Dortmund, Germany
E-mail address: `moeller@mathematik.tu-dortmund.de`

(Eva Riccomagno) *Department of Mathematics, University of Genova,*
via Dodecaneso 35, 16146 Genova, Italy
E-mail address: `riccomagno@dima.unige.it`