Fitting affine varieties to data: a nonlinearly structured low-rank approximation problem

Ivan Markovsky

School of Electronics and Computer Science
University of Southampton
im@ecs.soton.ac.uk

Abstract

Algebraic and geometric data fitting problems for a model class of affine varieties with bounded complexity (dimension and degree) are equivalent to low-rank approximation of a polynomially structured matrix constructed from the data. In algebraic fitting problems, the approximating matrix is unstructured and the corresponding low-rank approximation problem can be solved analytically by the singular value decomposition. In geometric fitting problems, the approximating matrix is polynomially structured and, except for the case of an affine model class, no analytic solution is know.

The equivalence of data modeling and low-rank approximation unifies existing curve fitting methods, showing that algebraic fitting is a relaxation of geometric fitting, obtained by removing the structure constraint, and reveals new solution approaches. Literate programs for solving nonlinearly structured low-rank approximation problems are presented and their effectiveness is illustrated on a range of artificially constructed and real-life data fitting problems.

Keywords: Curve fitting, Orthogonal regression, Total least squares, Low-rank approximation, Errors-in-variables modeling, Reproducible research, Literate programming.

1 Introduction

Background and contribution

Identifying a curve in a set of curves that best fits given data points is a common problem in computer vision, statistics, and coordinate metrology, see [Van97, Part IV]. More abstractly, approximation by Fourier series, wavelets, splines, and sum-of-damped-exponentials are also curve fitting problems. In the applications, the fitted curve is a model for the data and, correspondingly, the set of candidate curves is a model class.

Data modeling problems are specified by choosing a model class and a fitting criterion. The fitting criterion is maximisation of a measure for fit between the data and a model. Equivalently, the criterion can be formulated as minimization of a measure for lack of fit (misfit) between the data and a model. Data modeling problems can be classified according to the type of model and the type of fitting criterion as follows:

- linear/affine vs nonlinear model class,
- algebraic vs geometric fitting criterion.

A model is a subset of the data space. The model is linear/affine if it is a subspace/affine set. Otherwise, it is nonlinear. A geometric fitting criterion minimises the sum-of-squares of the Euclidean distances from the data points to a model. An algebraic fitting criterion minimises an equation error (residual) in a representation of the model. In general, the algebraic fitting criterion has no simple geometric interpretation. Problems using linear model classes and algebraic criteria are easier to solve numerically than problems using nonlinear model classes and geometric criteria.

The contributions of the paper are as follows.

- In Section 2, we formulate exact and approximate data modeling problems for a nonlinear model class of bounded complexity. The considered models are affine varieties [CLO04], *i.e.*, kernels of systems of multivariable polynomials. The complexity of an affine variety is defined as the pair of the variety's dimension and the total degree of its polynomial representation.

- In Section 3, we establish equivalence between the data modeling problem and low-rank approximation of a polynomially structured matrix constructed from the data. As illustrated in Section 7, the result allows us to use a single algorithm and a piece of software for solving a wide variety of data fitting problems.

Explicit (Input/output) vs implicit (kernel) representations

In data modeling problems, the model is usually represented by a function $y = f(x)$. The corresponding statistical estimation problem is regression. We call the functional relation $y = f(x)$ among the variables x and y, an input/output representation of the model $\mathscr{B} = \{ (x,y) \mid y = f(x) \}$ that this relation defines. Indeed, the input/output representation $y = f(x)$ implies that x is an independent variable (an input) and y is a dependent variable (an output) of the model $\mathscr{B}$.

Input-output representations are appealing because they are *explicit functions*, mapping some variables (inputs) to other variables (outputs) and thus displaying a causal relation among the variables (the inputs cause the outputs). The alternative kernel representation $R(x,y) = 0$, used in the paper, defines an *implicit function*, which does not a priori bound one set of variables as a cause and another set of variables as an effect.

The a priori fixed causal relation, imposed on the to-be-found model by a postulated input/output representation, is restrictive. Consider, for example, data fitting by a model that is a conic section. Only parabolas and lines can be represented by functions. Hyperbolas, ellipses, and the vertical line $\{ (x,y) \mid x = 0 \}$ are not graphs of a function $y = f(x)$ and therefore can not be modelled by an input/output representation.

Related methods

Data modeling is a generic problem with applications throughout science and engineering. Our approach originates from the systems and control community, where of main interest is modeling of linear time-invariant dynamical systems (system identification). As discussed next, there are also links with problems and methods in computer vision, machine learning, computer algebra, and numerical linear algebra.

- Using a kernel instead of an input/output model representation is a significant generalization of the data modeling problem formulation. It is important, however, to detach the model from its representation and view it abstractly as a subset of the data space. This idea is promoted by Jan C. Willems in what is called behavioural approach to systems and control [Wil87, Wil07]. The work, reported in this paper, is an application of the behavioral approach to static nonlinear data modeling.

- In the systems and control literature, the geometric distance is called misfit and the algebraic distance latency, see [LD01]. Identification of a linear time-invariant dynamical systems, using the latency criterion, leads to the autoregressive moving average exogenous (ARMAX) setting [Lju99, SS89] and using the misfit criterion, leads to the errors-in-variables (EIV) setting [Söd07].

- In the computer vision literuture, there is a large body of work on ellipsoid fitting (see, *e.g.*, [Boo79, GGS94, Kan94, FPF99, MKV04]), which is a special case of the considered data fitting problem when the total degree of the polynomial is two.

- State-of-the art image segmentation methods are based on the level set method [Set99]. Level set methods use implicit equations to represent a contour in the same way in which we use kernel representations to represent a model. The methods used for parameter estimation in the level set literature, however, are based on solution of partial differntial equations while we use classical numerical optimization methods.

- Relaxation of the nonlinearly structured low-rank approximation problem, based on ignoring the nonlinear structure and thus solving the problem as unstructured low-rank approximation, (*i.e.*, the algebraic fitting method) is known in the machine learning literature as kernel principal component analysis [SSM99].

- The principal curves, introduced in [HS89], lead to a problem of minimising the sum of squares of the distances from data points to a curve. This is a polynomially structured low-rank approximation problem. More generally, dimensionality reduction by manifold learning, see, *e.g.*, [ZZ05] is related to the problem of fitting an affine variety to data, which is also polynomially structured low-rank approximation.
- Nonlinear (Vandermonde) structured total least squares problems are discussed in [LVD02, RPG98] and are applied to fitting a sum of damped exponentials model to data. It is well known [MWVD06, page 129], however, that fitting a sum of damped exponentials to data can be solved as a Hankel structured approximation problem. In contrast, the geometric data fitting problem considered in this paper can not in general be reduced to a linearly structured problem and is therefore a genuine application of nonlinearly structured low-rank approximation.
- We use multivariable polynomials as model representations, so that concepts and methods from algebraic geometry is relevant for our study. In particular, the problem of passing from image to kernel representation of the model is known as the implicialization problem [CLO04, Page 96] in computer algebra. We use also the reverse transformation—passing from a kernel to an image representation of the model, which is a problem of solving a system of multivariable polynomial equations.

2 A framework for static nonlinear data modeling

Data, model class, and model complexity

We consider static multivariate modeling problems. The to-be-modelled data w_{d} is a set of N observations

$$w_{\mathrm{d}} = \{\, w_{\mathrm{d}}(1), \ldots, w_{\mathrm{d}}(N) \,\} \subset \mathbb{R}^q,$$

where the observations $w_{\mathrm{d}}(i)$, $i = 1, \ldots, N$, (called data points) are real q-dimensional vectors. A model for w_{d} is a subset of the data space $\mathbb{R}^q$ and a model class $\mathscr{M}^q$ for w_{d} is a set of subsets of the data space $\mathbb{R}^q$, *i.e.*, $\mathscr{M}^q$ is an element of the powerset $2^{\mathbb{R}^q}$. For example, the linear model class in $\mathbb{R}^q$ consists of all subspaces of $\mathbb{R}^q$. An example of a nonlinear model class in $\mathbb{R}^2$ is the set of all conic sections. When the dimension q of the data space is understood from the context, it will be skipped from the notation of the model class.

In [MWVD06, page 110], the complexity of a linear model $\mathscr{B}$ is defined as the dimension of $\mathscr{B}$, *i.e.*, the smallest natural number $\mathtt{m}$, such that there is a linear function $P : \mathbb{R}^{\mathtt{m}} \to \mathbb{R}^q$ for which

$$\mathscr{B} = \mathrm{image}(P) := \{\, P(u) \mid u \in \mathbb{R}^{\mathtt{m}} \,\}. \tag{1}$$

Similarly, the dimension of a nonlinear model $\mathscr{B}$ is defined as the smallest natural number $\mathtt{m}$, such that there is a (possibly nonlinear) function $P : \mathbb{R}^{\mathtt{m}} \to \mathbb{R}^q$, for which (1) holds. In the context of nonlinear models, however, the model dimension alone is not sufficient to define the model complexity. For example, in $\mathbb{R}^2$ both a linear model (a line passing through the origin) and an ellipse have dimension equal to one, however, it is intuitively clear that the ellipse is a more “complex” model than the linear one.

The missing element in the definition of the model complexity in the nonlinear case is the “complexity” of the function P. In what follows, we restrict to models that can be represented as kernels of polynomial functions:

$$\mathscr{B} = \ker(R) := \{\, w \in \mathbb{R}^q \mid R(w) = 0 \,\}, \tag{2}$$

i.e., we consider models that are affine varieties. Complexity of an affine variety (1) is defined as the pair $(\mathtt{m}, d)$, where $\mathtt{m}$ is the dimension of $\mathscr{B}$ and d is the total degree of R. This definition allows us to distinguish a linear or affine model ($d = 1$) from a nonlinear model ($d > 1$) with the same dimension. For a model $\mathscr{B}$ with complexity $(\mathtt{m}, d)$, we call d the degree of $\mathscr{B}$.

The complexity of a model class is the maximal complexity (in a lexicographic ordering of the pairs $(\mathtt{m}, d)$) over all models in the class. The model class of complexity bounded by $(\mathtt{m}, d)$ is denoted by $\mathscr{M}_{\mathtt{m},d}$.

Exact and approximate data modeling

A model $\mathscr{B}$ is an exact model for the data w_d if $w_\text{d} \subset \mathscr{B}$. Otherwise, it is an approximate model. An exact model for the data may not exist in a model class of bounded complexity. This is generically the case when the data is noisy and the number of observations is sufficiently large relative to the model complexity. A practical data fitting problem must involve approximation, however, our starting point for data modeling is the simpler problem of exact data modeling. Justification for this decision, apart from its pedagogical value, is that exact modeling problems are an ingredient of approximate modeling problems.

Problem 1 (Exact modeling)**.** Given data $w_\text{d} \subset \mathbb{R}^q$ and a complexity bound $(\mathtt{m}, d)$, find a model $\widehat{\mathscr{B}}$ in the model class $\mathscr{M}_{\mathtt{m},d}$ that contains the data and has minimal (in the lexicographic ordering) complexity or assert that such a model does not exist.

Due to the bound on the model complexity, Problem 1 may not have a solution. The question occur:

> Under what conditions on the data w_d and the model class $\mathscr{M}_{\mathtt{m},d}$ a solution to Problem 1 exist?

If a solution exists it is unique. This unique solution is called the *most powerful unfalsified model (MPUM)* [Wil86] for the data w_d in the model class $\mathscr{M}_{\mathtt{m},d}$ and we denote it by $\mathscr{B}_\text{mpum}(w_\text{d})$.

Suppose that the data w_d is generated by a model $\bar{\mathscr{B}}$ in the model class $\mathscr{M}_{\mathtt{m},d}$, *i.e.*,

$$w_\text{d} \subset \bar{\mathscr{B}} \in \mathscr{M}_{\mathtt{m},d}.$$

Clearly, the exact identification problem has a solution in the model class $\mathscr{M}_{\mathtt{m},d}$, however, the solution $\mathscr{B}_\text{mpum}(w_\text{d})$ may not be the data generating model $\bar{\mathscr{B}}$. The question occurs:

> (Identifiability) Under what conditions on the data w_d, the data generating model $\bar{\mathscr{B}}$, and the model class $\mathscr{M}_{\mathtt{m},d}$, the MPUM $\mathscr{B}_\text{mpum}(w_\text{d})$ coincides with the data generating model $\bar{\mathscr{B}}$?

When an exact model does not exist in the considered model class, an approximate model that is in some sense "close" to the data is aimed at instead. This leads to the following approximate data modeling problem.

Problem 2 (Approximate modeling)**.** Given data $w_\text{d} \subset \mathbb{R}^q$, a complexity bound $(\mathtt{m}, d)$, and a distance measure $\text{dist}(w_\text{d}, \mathscr{B})$ between the data w_d and a model $\mathscr{B}$, find a model $\widehat{\mathscr{B}}$ in the model class $\mathscr{M}_{\mathtt{m},d}$ that is as close as possible to the data, *i.e.*,

$$\text{minimize} \quad \text{over } \mathscr{B} \in \mathscr{M}^q_{\mathtt{m},d} \quad \text{dist}(w_\text{d}, \mathscr{B}). \tag{3}$$

Two distance measures are often used for data fitting: the geometric distance

$$\text{dist}(w_\text{d}, \mathscr{B}) := \min_{\widehat{w} \subset \mathscr{B}} \sqrt{\sum_{i=1}^{N} \left\| w_\text{d}(i) - \widehat{w}(i) \right\|_2^2}, \tag{4}$$

where $\|\cdot\|_2$ is the 2-norm in the data space $\mathbb{R}^q$, and the algebraic "distance"

$$\text{dist}'(w_\text{d}, \mathscr{B}) := \sqrt{\sum_{i=1}^{N} \left\| R\big(w_\text{d}(i)\big) \right\|_\text{F}^2}, \tag{5}$$

where $\|\cdot\|_\text{F}$ is the Frobenius matrix norm (*i.e.*, 2-norm of the vector obtained by concatenation of the columns of a matrix) and R is a $q \times p$ polynomial matrix that defines a kernel representation (2) of the model $\mathscr{B}$.

Note that the algebraic distance depends on the choice of the kernel representation (which is not unique), while the geometric distance is representation invariant. In addition, the geometric distance is invariant to translation, rotation, and scaling of the data points, while the algebraic distance is not.

Special cases

The model class $\mathscr{M}^q_{\mathtt{m},d}$ and the related exact and approximate modeling problems 1 and 2 have as an important special case the linear model class and data modeling problems, based on the principal component analysis.

1. *Linear/affine model class of bounded complexity.* An affine model $\mathscr{B}$ (*i.e.*, a affine set in $\mathbb{R}^q$) is an affine variety, defined by a first order polynomial through kernel or image representation. The dimension of the affine variety, as defined in Section 2, coincides with the dimension of the affine set. Therefore, $\mathscr{M}^q_{\mathtt{m},1}$ is an affine model class in $\mathbb{R}^q$ with complexity bounded by $\mathtt{m}$. The linear model class in $\mathbb{R}^q$, with dimension bounded by $\mathtt{m}$, is a subset of $\mathscr{M}^q_{\mathtt{m},1}$ and is denoted by $\mathscr{L}^q_{\mathtt{m}}$.

2. *Exact data fitting by a linear model.* Existence of a linear model $\widehat{\mathscr{B}}$ of bounded complexity for the data w_{d} is equivalent to rank deficiency of the matrix

$$\Phi(w_{\mathrm{d}}) := \begin{bmatrix} w_{\mathrm{d}}(1) & \cdots & w_{\mathrm{d}}(N) \end{bmatrix} \in \mathbb{R}^{q\times N},$$

composed of the data. Moreover, the rank of the matrix $\Phi(w_{\mathrm{d}})$ is equal to the *minimal* dimension of an exact model for w_{d}

$$\text{existence of } \widehat{\mathscr{B}} \in \mathscr{L}^q_{\mathtt{m}}, \text{ such that } w_{\mathrm{d}} \subset \widehat{\mathscr{B}} \quad \iff \quad \operatorname{corank}\big(\Phi(w_{\mathrm{d}})\big) \geq \mathtt{p} := q - \mathtt{m}, \tag{6}$$

where

$$\text{for an } q\times N \text{ matrix } \Phi, \quad \operatorname{corank}(\Phi) := \min(q,N) - \operatorname{rank}(\Phi).$$

The exact model

$$\widehat{\mathscr{B}} = \operatorname{image}\big(\Phi(w_{\mathrm{d}})\big) \tag{7}$$

of minimal dimension

$$\mathtt{m} = \operatorname{rank}\big(\Phi(w_{\mathrm{d}})\big)$$

always exists and is unique.

The equivalence (6) between data modeling and the concept of rank is the basis for application of linear algebra and matrix computations to linear data modeling. Indeed, (7) provides an *algorithm* (compute a basis for the image of $\Phi(w_{\mathrm{d}})$) for exact linear data modeling. As shown next, (6) has also direct relevance to approximate data modeling.

3. *Geometric fitting by a linear model is a low-rank approximation problem.* Data modeling using the model class of linear models $\mathscr{L}^q_{\mathtt{m}}$ and the geometric fitting criterion (4) is a low-rank approximation problem

$$\begin{aligned} &\text{minimize} && \text{over } \widehat{w} \in (\mathbb{R}^q)^N \quad \big\|\Phi(w_{\mathrm{d}}) - \Phi(\widehat{w})\big\|_{\mathrm{F}} \\ &\text{subject to} && \operatorname{corank}\big(\Phi(\widehat{w})\big) \geq q - \mathtt{m}. \end{aligned} \tag{8}$$

The rank constraint in (8) is equivalent to the constraint that $\widehat{w}$ is exact for a linear model of dimension bounded by $\mathtt{m}$. This shows that exact modeling is an ingredient of approximate modeling.

4. *Low-rank approximation is equivalent to principal component analysis (PCA).* The PCA method for dimensionality reduction is usually introduced in a stochastic setting as maximisation of the variance of the projected data on a subspace. Computationally, however, the problem of finding the principal components and the corresponding principal vectors is an eigenvalue/eigenvector decomposition problem for the sample covariance matrix

$$\Psi(w_{\mathrm{d}}) := \Phi(w_{\mathrm{d}})\Phi^\top(w_{\mathrm{d}}).$$

From this algorithmic point view, the equivalence of PCA and low-rank approximation problem is a basic linear algebra fact.

Lemma 3 (Equivalence of PCA and low-rank approximation)**.** *The space spanned by the first* $\mathtt{m}$ *principal vectors of* w_d *coincides with the model* $\widehat{\mathscr{B}} = \text{image}\big(\Phi(\widehat{w})\big)$*, where* $\widehat{w}$ *is a solution of (8).*

5. *Errors-in-variables (EIV) modeling and low-rank approximation.* From a statistical point of view, low-rank approximation (and therefore PCA) is related to EIV modeling [Ful87, Gle81]. Closely related to the EIV framework for low-rank approximation is the probabilistic PCA framework of [TB99]. More specifically, a solution of the low-rank approximation problem (8) yields a maximum likelihood estimator for the true model in the EIV setup
$$w_\mathrm{d} = \bar{w} + \widetilde{w}, \qquad \text{where} \qquad \bar{w} \subset \bar{\mathscr{B}} \tag{9}$$
and $\widetilde{w}$ is a set of independent, zero mean, Gaussian random vectors, with covariance matrix $\sigma^2 I_q$. (I_q is the identity matrix of dimension q.)

6. *Algebraic fit by a linear model and regression.* A linear model class, defined by the input/output representation
$$\mathscr{B} = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \mid \Theta^\top x = y \right\}, \tag{10}$$
and algebraic fitting criterion (5), where $w := \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ and $R(w) := \Theta^\top x - y$ lead to the ordinary linear least squares problem
$$\text{minimize} \quad \text{over } \Theta \in \mathbb{R}^{q \times \mathtt{p}} \quad \left\| \Theta^\top \Phi(x_\mathrm{d}) - \Phi(y_\mathrm{d}) \right\|_\mathrm{F}. \tag{11}$$
The statistical setting for the least squares approximation problem (11) is the classical regression model.

   A mixture of input/output model representation and errors-in-variables stochastic model for static nonlinear data modeling is called polynomial errors-in-variables regression [CS98].

7. *Algebraic curves.* An affine variety of dimension one is called an algebraic curve. Famous examples of algebraic curves are:

   - conic sections $\mathscr{B} = \{\, w \mid w^\top A w + b^\top w + c \,\}$, where $A = A^\top$, b, and c are parameters,
   - cissoid $\mathscr{B} = \{\, (x,y) \mid y^2(1+x) = (1-x)^3 \,\}$,
   - folium of Descartes $\mathscr{B} = \{\, (x,y) \mid x^3 + y^3 - 3xy = 0 \,\}$,
   - limacon of Pascal $\mathscr{B} = \{\, (x,y) \mid y^2 + x^2 - (4x^2 - 2x + 4y^2)^2 = 0 \,\}$.

   In the special case of a curve in the plane, we use the notation $x := w_1$, $y := w_2$. Note that $w = (x,y)$ is not necessarily an input/output partitioning of the variables.

8. *Geometric distance for linear and quadratic models.* The two plots in Figure 1 illustrate the geometric distance (4) from a set of eight data points w_d in the plane to, respectively, linear $\mathscr{B}_1$ and quadratic $\mathscr{B}_2$ models. As its name suggests, $\text{dist}(w_\mathrm{d}, \mathscr{B})$ has geometric interpretation. In order to compute the geometric distance, we project the data points on the models. This is a simple task (linear least squares problem) for linear models but a nontrivial task (nonconvex optimization problem) for nonlinear models. In contrast, the algebraic "distance" (not visualised on the figure) has no simple geometrical interpretation but is easy to compute for linear and nonlinear models alike.

3 Equivalence of static nonlinear data modeling and low-rank approximation

Parametrisation of the kernel representations

Consider a kernel representation (2) of an affine variety. The polynomial R can be written as
$$R_\Theta(w) = \sum_{k=1}^{n} \Theta_k \phi_k(w) = \Theta^\top \phi(w), \tag{12}$$

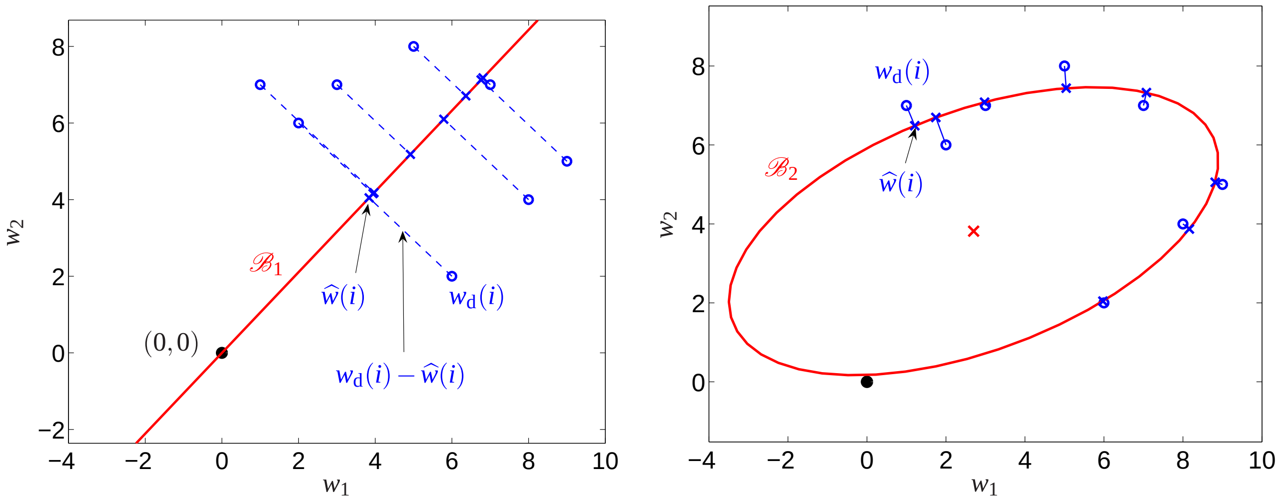


Figure 1: Geometric distance from eight data points to a linear (left) and quadratic (right) models

where Θ is an $n \times p$ parameter matrix with

$$n := \binom{q+d}{d}, \tag{13}$$

d is the total degree of R, and $\phi(w)$ is a vector of all monomials with degree up to d. The monomials are ordered in $\phi(w)$ in decreasing degree according to the lexicographic ordering (with alphabet $w_1, \dots, w_q$). For example, with $q = 2$, $d = 2$, and $w = (x, y)$,

$$n = 6 \qquad \text{and} \qquad \phi(w) = \begin{bmatrix} x^2 & xy & x & y^2 & y & 1 \end{bmatrix}^\top. \tag{14}$$

The kernel representation (2) is not unique due to

1. nonminimality — existence of linearly dependent columns of the matrix R and
2. nonuniqueness of R — pre-multiplication of R by a nonsingular matrix Q, define an equivalent representation of the model, *i.e.*, $\ker(R) = \ker(QR)$).

Minimality of the kernel representation is equivalent to the condition that the parameter Θ is full column rank. The nonuniqueness of R_Θ corresponds to a nonuniqueness of Θ. The parameters Θ and ΘQ, where Q is a nonsingular matrix, define the same model. Therefore, without loss of generality, we can assume that the representation is minimal and normalise it, so that $\Theta^\top \Theta = I_{\mathtt{p}}$.

Note that a $n \times \mathtt{p}$ full column rank matrix Θ defines via (12) a polynomial matrix R_Θ, which defines a minimal kernel representation (2) of a model $\mathscr{B}_\Theta$ in $\mathscr{M}^q_{\mathtt{m},d}$. The mapping $\mathscr{B}_\Theta : \mathbb{R}^{n \times \mathtt{p}} \to \mathscr{M}^q_{\mathtt{m},d}$ is a function. Vice verse, a model $\mathscr{B}$ in $\mathscr{M}^q_{\mathtt{m},d}$ corresponds to a (nonunique) $n \times \mathtt{p}$ full column rank matrix Θ, such that $\mathscr{B} = \mathscr{B}_\Theta$. For given q, there are one-to-one mappings $n \leftrightarrow d$ and $\mathtt{p} \leftrightarrow \mathtt{m}$, given by (13) and $\mathtt{p} = q - \mathtt{m}$, respectively.

Main results

Our first result is a generalization of example 2 — exact linear data modeling — in Section 2.

Proposition 4. *For a data set $w_{\rm d} \subset \mathbb{R}^q$ and a complexity specification $(\mathtt{m}, d)$,*

$$\text{existence of } \mathscr{B} \in \mathscr{M}^q_{\mathtt{m},d}, \text{ such that } w_{\rm d} \subset \mathscr{B} \quad \iff \quad \operatorname{corank}\big(\Phi_d(w_{\rm d})\big) \geq \mathtt{p} := q - \mathtt{m},$$

where

$$\Phi_d(w_{\rm d}) := \begin{bmatrix} \phi\big(w_{\rm d}(1)\big) & \cdots & \phi\big(w_{\rm d}(N)\big) \end{bmatrix}.$$

Proof. ($\Rightarrow$) Let $\mathscr{B} \in \mathscr{M}_{\mathtt{m},d}$ be an exact model for w_d and consider a minimal kernel representation (2) of $\mathscr{B}$, parameterized as in (12). The minimality of the kernel representation implies that the parameter matrix Θ is full column rank $n \times \mathtt{p}$, where n is defined in (13). We have,

$$\begin{aligned} w_\mathrm{d} \subset \mathscr{B} \quad &\iff \quad w_\mathrm{d}(i) \in \mathscr{B}, \quad \text{for } i = 1, \dots, N \\ &\iff \quad \Theta^\top \phi\big(\widehat{w}(i)\big) = 0, \quad \text{for } i = 1, \dots, N \\ &\iff \quad \Theta^\top \Phi_d(w_\mathrm{d}) = 0. \end{aligned} \tag{15}$$

Therefore, $\Phi_d(w_\mathrm{d})$ has at least $\mathtt{p}$ dimensional left null space, so that

$$\operatorname{corank}\big(\Phi_d(w_\mathrm{d})\big) \geq \mathtt{p}.$$

($\Leftarrow$) Let $\operatorname{corank}\big(\Phi_d(w_\mathrm{d})\big) \geq q - \mathtt{m} =: \mathtt{p}$. Then there exists a $n \times \mathtt{p}$ full rank matrix Θ, such that $\Theta^\top \Phi_d(w_\mathrm{d}) = 0$. The equivalences in (15) prove that the matrix Θ defines (via (2) and (12)) an exact model $\mathscr{B}$ for the data w_d. □

Proposition 4 answers the question of the existence of the MPUM in a model class of affine varieties with bounded complexity. Moreover, the proof is constructive and suggests an algorithm for identification of the MPUM (as well as detecting its existence). Under the assumptions of the proposition, the matrix $\Phi_d(w_\mathrm{d})$ has at lease $\mathtt{p} = q - \mathtt{m}$ dimensional left null space. A set of basis vectors for this space gives a parameter matrix Θ, which defines via (2) and (12) a kernel representation of the MPUM $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ in the model class $\mathscr{M}_{\mathtt{m},d}$.

Proposition 4 also provides an answer to the identifiability question.

Corollary 5. *Consider a data set $w_\mathrm{d} \subset \mathbb{R}^q$, generated by a model $\bar{\mathscr{B}} \in \mathscr{M}_{\mathtt{m},d}$, i.e., $w_\mathrm{d} \subset \bar{\mathscr{B}}$. The MPUM $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ in the model class $\mathscr{M}_{\mathtt{m},d}$ exists and coincides with the data generating model $\bar{\mathscr{B}}$ if and only if*

$$\operatorname{corank}\big(\Phi_d(w_\mathrm{d})\big) = q - \dim(\bar{\mathscr{B}}).$$

Proof. From Propositition 4 we have that

$$w_\mathrm{d} \subset \bar{\mathscr{B}} \in \mathscr{M}_{\mathtt{m},d} \qquad \iff \qquad \operatorname{corank}\big(\Phi_d(w_\mathrm{d})\big) \leq q - \dim(\bar{\mathscr{B}}).$$

If equality holds, $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d}) = \bar{\mathscr{B}}$. Othewise, $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d}) \subset \bar{\mathscr{B}}$. □

Proposition 6 (Algebraic fit $\iff$ unstructured low-rank approximation)**.** *The algebraic curve fitting problem*

$$\text{minimize} \quad \textit{over } \Theta \in \mathbb{R}^{n \times \mathtt{p}} \quad \sqrt{\sum_{i=1}^{N} \big\|R_\Theta\big(w_\mathrm{d}(i)\big)\big\|_\mathrm{F}^2} \quad \textit{subject to} \quad \Theta^\top \Theta = I_\mathtt{p} \tag{16}$$

is equivalent to the unstructured low-rank approximation problem

$$\begin{aligned} \text{minimize} \quad &\textit{over } \widehat{\Phi} \in \mathbb{R}^{q \times \mathtt{p}} \quad \|\Phi_d(w_\mathrm{d}) - \widehat{\Phi}\|_\mathrm{F} \\ \textit{subject to} \quad &\operatorname{corank}(\widehat{\Phi}) \geq \mathtt{p}. \end{aligned} \tag{17}$$

Proof. Using the polynomial representation (12), the squared cost function of (16) can be rewritten as a quadratic form

$$\begin{aligned} \sum_{i=1}^{N} \big\|R_\Theta\big(w_\mathrm{d}(i)\big)\big\|_\mathrm{F}^2 &= \big\|\Theta^\top \Phi_d(w_\mathrm{d})\big\|_\mathrm{F}^2 \\ &= \operatorname{trace}\big(\Theta^\top \Phi_d(w_\mathrm{d}) \Phi_d^\top(w_\mathrm{d}) \Theta\big) = \operatorname{trace}\big(\Theta^\top \Psi_d(w_\mathrm{d}) \Theta\big). \end{aligned}$$

Therefore, the algebraic fitting problem is equivalent to an eigenvalue problem for $\Psi_d(w_\mathrm{d})$ or, equivalently (Lemma 3), to low-rank approximation of Φ_d. □

Proposition 7 (Geometric fit $\iff$ polynomially structured low-rank approximation)**.** *The geometric curve fitting problem (3) is equivalent to the polynomially structured low-rank approximation problem*

$$\begin{aligned} &\text{minimize} \quad \textit{over } \widehat{w} \in (\mathbb{R}^q)^N \quad \sqrt{\sum_{i=1}^{N} \|w_\mathrm{d}(i) - \widehat{w}(i)\|^2} \\ &\textit{subject to} \quad \operatorname{corank}\big(\Phi_d(\widehat{w})\big) \geq \mathtt{p}. \end{aligned} \tag{18}$$

Proof. Follows directly from Proposition 4. □

Corollary 8. *The algebraic fitting problem (16) is a relaxation of the geometric fitting problem (3), obtained by removing the structure constraint of the approximating matrix* $\Phi_d(\widehat{w})$.

4 Algorithms

Complexity selection

Initial approximations

9a ⟨bclra 9a⟩≡

```
function [th, sh] = bclra(w, d)
[q, N] = size(w); D = monomials(d, q);
qext = nchoosek(q + d, d); td = ceil((q * d + 1) / 2);
⟨Define the Hermite polynomials 9b⟩
psi = zeros(qext, qext, td);
for i = 1:qext
  for j = 1:qext
    if i >= j
      Dij = D(i, :) + D(j, :);
      for l = 1:N
        psi_ijl = 1;
        for k = 1:q
          psi_ijl  = conv(psi_ijl, h{Dij(k) + 1}(w(k, l)));
        end
        psi_ijl = [psi_ijl zeros(1, td - length(psi_ijl))];
        psi(i, j, :) = psi(i, j, :) + reshape(psi_ijl(1:td), 1, 1, td);
      end
    end
  end
end
for k = 1:td, psi(:, :, k) = psi(:, :, k) + triu(psi(:, :, k)’, 1); end
[evec, ev] = polyeig_(psi); ev(find(ev < 0)) = inf;
[sh2, min_ind]  = min(ev); sh = sqrt(sh2); th = evec(:, min_ind);
```

9b ⟨*Define the Hermite polynomials* 9b⟩≡ (9a)

```
h{1} = @(x) 1; h{2} = @(x) x;
for k = 2:(2 * d)
  h{k + 1} = @(x) [x * h{k}(x) zeros(1, mod(k - 1, 2))] ...
                 - [0 (k - 1) * h{k - 1}(x)];
end
```

5 Properties

Invariance properties

Sensitivity analysis

6 Extensions

Centering and generalized low-rank approximation

Missing data

Constraints on the parameters Θ

Bounded models

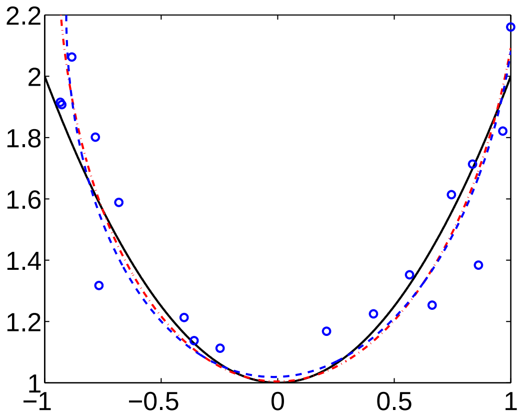
7 Examples

In this section, we apply the algebraic and geometric fitting methods on a range of algebraic curve fitting problems. In all examples, except for the last one, the data w_d is simulated in the errors-in-variables setup (9). The perturbations $\widetilde{w}(i)$, $i = 1, \ldots, N$ are independent, zero mean, normally distributed 2×1 vectors with covariance matrix $\sigma^2 I_2$. The number of data points N and the perturbation standard deviation σ are simulation parameters. The true model is plotted by a black solid line, the data points by circles, the algebraic fit by a red dashed dotted line, and the geometric fit by a blue dashed line.

Simulation 1: Parabola $\mathscr{B} = \{ (x,y) \mid y = x^2 + 1 \}$

10a ⟨examples 10a⟩≡ 10b▷

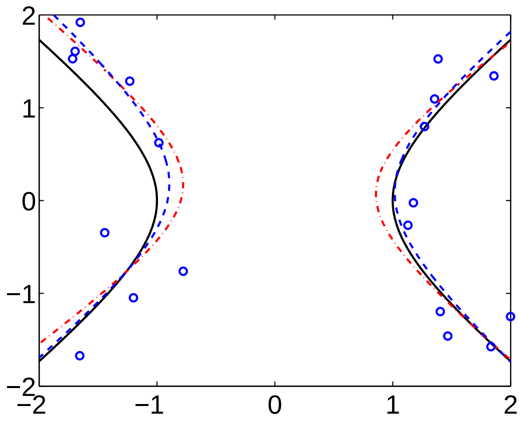
```
clear all
name = 'parabola';
N = 20; sigma = 0.1; d = 2;
syms x y;
r = x^2 - y + 1;
ax = [-1 1 1 2.2]; test
```



Simulation 2: Hyperbola $\mathscr{B} = \{ (x,y) \mid x^2 - y^2 - 1 = 0 \}$

10b ⟨examples 10a⟩+≡ ◁10a 11a▷

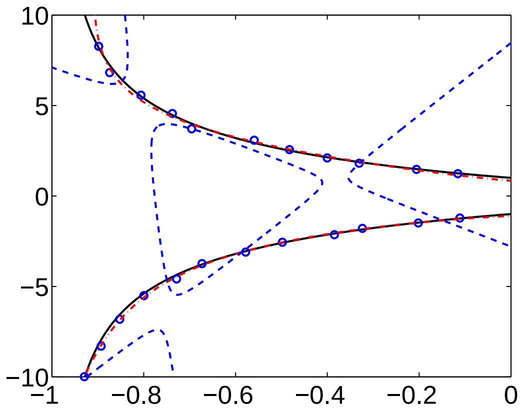
```
name = 'hyperbola';
N = 20; sigma = 0.3; d = 2;
syms x y;
r = x^2 - y^2 - 1;
ax = [-2 2 -2 2]; test
```



Simulation 3: Cissoid $\mathscr{B} = \{\,(x,y) \mid y^2(1+x) = (1-x)^3\,\}$

11a ⟨examples 10a⟩+≡ 11b▷

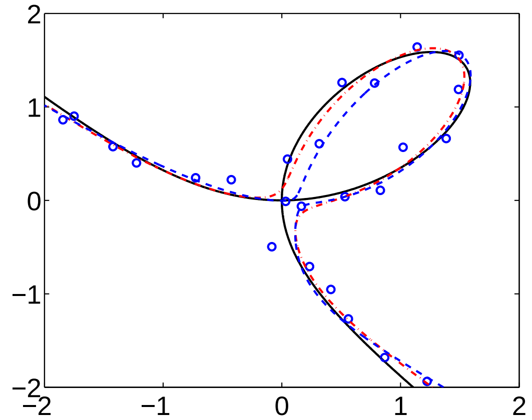
```
name = 'cissoid';
N = 25; sigma = 0.02; d = 3;
syms x y;
r = y^2 * (1 + x) - (1 - x)^3;
ax = [-1 0 -10 10]; test
```



Simulation 4: Folium of Descartes $\mathscr{B} = \{\,(x,y) \mid x^3 + y^3 - 3xy = 0\,\}$

11b ⟨examples 10a⟩+≡ 11c▷

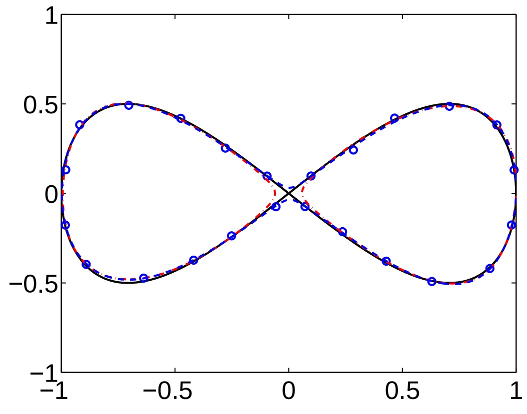
```
name = 'folium';
N = 25; sigma = 0.1; d = 3;
syms x y;
r = x^3 + y^3 - 3 * x * y;
ax = [-2 2 -2 2]; test
```



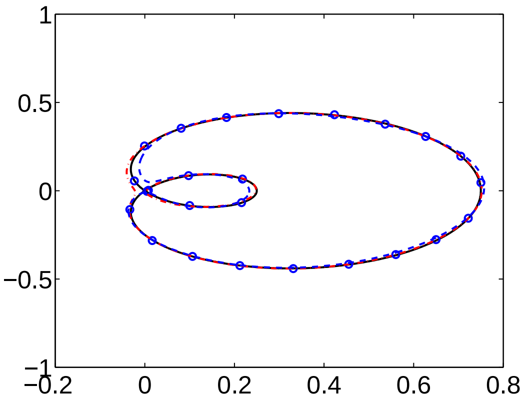
Simulation 5: Eight curve $\mathscr{B} = \{\,(x,y) \mid y^2 - x^2 + x^4 = 0\,\}$

11c ⟨examples 10a⟩+≡ 12a▷

```
name = 'eight';
N = 25; sigma = 0.01; d = 4;
syms x y;
r = y^2 - x^2 + x^4;
ax = [-1 1 -1 1]; test
```



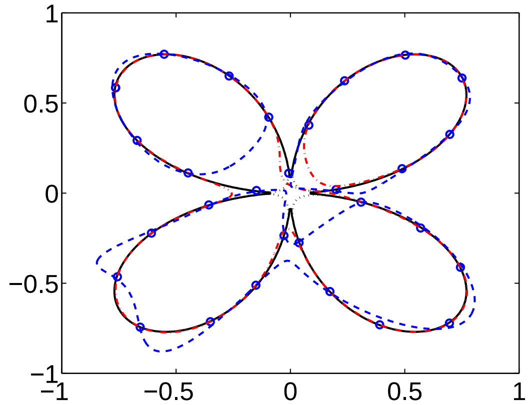
Simulation 6: Limacon of Pascal $\mathscr{B} = \{\,(x,y) \mid y^2 + x^2 - (4x^2 - 2x + 4y^2)^2 = 0\,\}$



12a ⟨`examples` 10a⟩+≡ 12b▷

```
name = 'limacon';
N = 25; sigma = 0.002; d = 4;
syms x y;
r = y^2 + x^2 - (4 * x^2 - 2 * x + 4 * y^2)^2;
ax = [-0.2 0.8 -1 1]; test
```

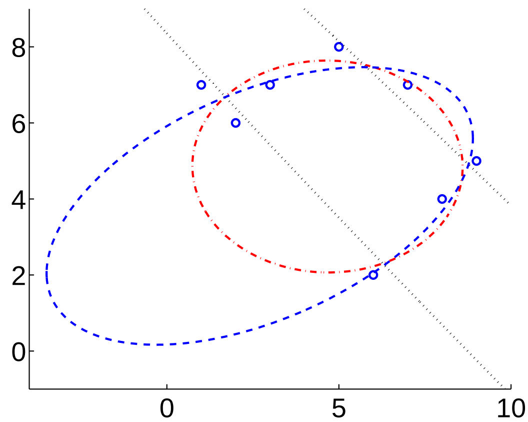
Simulation 7: Four-leaved rose $\mathscr{B} = \{\,(x,y) \mid (x^2 + y^2)^3 - 4x^2y^2 = 0\,\}$



12b ⟨`examples` 10a⟩+≡ 12c▷

```
name = 'rose';
N = 30; sigma = 0.002; d = 6;
syms x y;
r = (x^2 + y^2)^3 - 4 * x^2 * y^2;
ax = [-1 1 -1 1]; test
```

Simulation 8: "Special data" example from [GGS94]



12c ⟨`examples` 10a⟩+≡ ◁12b

```
name = 'special-data';
wd = [1 2 5 7 9 3 6 8 ;
      7 6 8 7 5 7 2 4 ]; wb = wd;
d = 2; ax = [-4 10 -1 9];
xini = [wd(:)' 1 0 0 1 0 -1]';
figure, ⟨Fit data 14c⟩
⟨Plot results 14d⟩
```

8 Literate programs

In this section we give the software implementation of the developed curve fitting methods and the scripts that reproduce the numerical expamples.

The function `monomials` generates a function that evaluates the q-variate vector of monomials ϕ, with total degree d. The monomials $w_1^{n_1} \cdots w_q^{n_q}$ are ordered in descending powers, according to the lexicographic order of the tuple $(n_1, \dots, n_q)$ (see for example (14)).

12d ⟨`monomials` 12d⟩≡

```
function [D, phi] = monomials(d, q)
```

```
Nt = (d + 1) ^ q; D = []; s  = [];
for ind = 1:Nt
    sub = fliplr(ind2sub(ind, (d + 1) * ones(1, q)) - 1);
    if (sum(sub) <= d)
        for i = q:-1:1,
            s = sprintf('.* w(%d,:) .^ %d %s', i, sub(i), s);
        end
        D = [sub; D]; s = sprintf('; %s', s(4:end));
    end
end
eval(sprintf('phi = @(w) [%s];', s(2:end)))
```

The unstructured low-rank approximation problem (8) is solved using the singular value decomposition.

13a ⟨lra 13a⟩≡

```
function [th, Phih] = lra(Phi, r);
[u, s, v] = svd(Phi, 0); th = u(:, (r + 1):end);
if nargout > 1, Phih = u(:, 1:r) * s(1:r, 1:r) * v(:, 1:r)'; end
```

The nonlinearly structured low-rank approximation problem (18) is solved numerically using Matlab's Optimization Toolbox.

13b ⟨nslra 13b⟩≡ 13c▷

```
function [th, wh, info] = nslra(wd, phi, r, xini)
[q, N] = size(wd); nt = size(phi(wd), 1);
```

If not specified, the initial approximation is taken as the algebraic fit and the noisy data points.

13c ⟨nslra 13b⟩+≡ ◁13b 13d▷

```
if (nargin < 4) | isempty(xini)
    [tini, Pini] = lra(phi(wd), r); %xini = [wd(:); tini(:)];
    nti = (nt - 1) / q;
    wh = Pini(nti:nti:end, :); xini = [wh(:); tini(:)];
end
```

Anonymous functions that extract the data approximation $\widehat{w}$ and the model parameter θ from the optimization parameter `x` are defined next.

13d ⟨nslra 13b⟩+≡ ◁13c 13e▷

```
w = @(x) reshape(x(1:(q * N)), q, N);
t = @(x) reshape(x((q * N + 1):end), nt - r, nt)';
```

The optimization problem is set and solved, using the Optimization Toolbox:

13e ⟨nslra 13b⟩+≡ ◁13d

```
prob = optimset();
prob.solver    = 'fmincon';
prob.options   = optimset('disp', 'iter');
prob.objective = @(x) norm(wd - w(x), 'fro');
prob.nonlcon   = @(x) deal([], ...
                           [t(x)' * phi(w(x)), t(x)' * t(x) - eye(nt - r)]);
prob.x0 = xini;
[x, fval, flag, info] = fmincon(prob); wh = w(x); th = t(x);
```

The test script `test` assumes that the simulation parameters—polynomial r in x and y, defined as a symbolic object; total degree d of r; number of data points N; noise standard deviation σ; and coordinates `ax` of a rectangle for plotting the results— are already defined.

13f ⟨test 13f⟩≡

⟨*Default parameters* 14a⟩
⟨*Generate data* 14b⟩
⟨*Fit data* 14c⟩
⟨*Plot results* 14d⟩

If not specified otherwise, $q = 2$, $\mathtt{m} = 1$.

14a ⟨*Default parameters* 14a⟩≡ (13f)

```
if ~exist('q'), q = 2; end
if ~exist('m'), m = 1; end
if ~exist('xini'), xini = []; end
```

The true (`wb`) and noisy (`wd`) data points are generated as follows:

14b ⟨*Generate data* 14b⟩≡ (13f)

```
figure,
H = plot_model(r, ax, 'LineStyle', '-', 'color', 'k');
wb = []; for h = H', wb = [wb [get(h, 'XData');
get(h, 'YData')]]; end
% sample N points on the curve
wb = wb(:, round(linspace(1, size(wb, 2), N)));
randn('seed', 0);
wd = wb + sigma * randn(size(wb)); % add noise
```

The data is fitted by the algebraic (`lra`) and geometric (`nslra`) fitting methods:

14c ⟨*Fit data* 14c⟩≡ (12c 13f)

```
qext = nchoosek(q + d, d); p = q - m;
[D, phi] = monomials(d, q);
th_exc   = lra(phi(wb), qext - p); % exact modeling
th_ini   = bclra(wd, d); % bias corrected algebraic fit
[th, wh] = nslra(wd, phi, qext - p, xini); % geometric fit
```

The noisy data and the two fitted models are plotted on top of the true model:

14d ⟨*Plot results* 14d⟩≡ (12c 13f)

```
hold on; plot(wd(1,:), wd(2,:), 'o', 'markersize', 7);
plot_model(th2poly(th_exc, phi), ax, ...
                   'LineStyle', ':', 'color', 'k');
plot_model(th2poly(th_ini, phi), ax, ...
                   'LineStyle', '-.', 'color', 'r');
plot_model(th2poly(th, phi), ax, ...
                   'LineStyle', '-', 'color', 'b');
axis(ax); print_fig(sprintf('%s-est', name))
```

Plotting the algebraic curve $\mathscr{B} = \{ w \mid \phi(w)\theta = 0 \}$ in a region, defined by `rect` is done with `plot_model`.

14e ⟨`plot_model` 14e⟩≡

```
function H = plot_model(r, rect, varargin)
try % q == 2
    H = ezplot(r, rect);
catch % q == 3
    s = solve(r, 'x', 'y', 'z');
    try % m == 1
        H = ezplot3(s.x(1), s.y(1), s.z(1), rect)
    catch % m == 2
        H = ezcontour(s.x(1), s.y(1), s.z(1), rect)
    end
end
if nargin > 2, for h = H', set(h, varargin{:}); end, end
```

14f ⟨`th2poly` 14f⟩≡

```
function r = th2poly(th, phi)
try % q == 2
   r = @(x, y) th' * phi([x y]');
catch % q = 3
   r = @(x, y, z) th' * phi([x y z]');
end
```

15 ⟨print_fig 15⟩≡

```
function print_fig(file_name)
xlabel('x'), ylabel('y'), title('t')
set(gca, 'fontsize', 25)
eval(['print -depsc ' file_name '.eps'])
```

References

[Boo79] F. L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:59–71, 1979.

[CLO04] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Springer, 2004.

[CS98] C.-L. Cheng and H. Schneeweiss. Polynomial regression with errors in the variables. *J. R. Stat. Soc. B*, 60(1):189–199, 1998.

[FPF99] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least-squares fitting of ellipses. *IEEE Trans. Pattern Anal. Machine Intelligence*, 21(5):476–480, 1999.

[Ful87] W. Fuller. *Measurement Error Models*. Wiley, New York, 1987.

[GGS94] W. Gander, G. Golub, and R. Strebel. Fitting of circles and ellipses: Least squares solution. *BIT*, 34:558–578, 1994.

[Gle81] L. Gleser. Estimation in a multivariate "errors in variables" regression model: large sample results. *The Annals of Statistics*, 9(1):24–44, 1981.

[HS89] T. Hastie and W. Stuetzle. Principal curves. *J. American Statistical Association*, 84:502–516, 1989.

[Kan94] K. Kanatani. Statistical bias of conic fitting and renormalization. *IEEE Trans. Pattern Anal. Machine Intelligence*, 16(3):320–326, 1994.

[LD01] P. Lemmerling and B. De Moor. Misfit versus latency. *Automatica*, 37:2057–2067, 2001.

[Lju99] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, NJ, 1999.

[LVD02] P. Lemmerling, S. Van Huffel, and B. De Moor. The structured total least squares approach for nonlinearly structured matrices. *Numer. Linear. Algebra Appl.*, 9(1–4):321–332, 2002.

[MKV04] I. Markovsky, A. Kukush, and S. Van Huffel. Consistent least squares fitting of ellipsoids. *Numerische Mathematik*, 98(1):177–194, 2004.

[MWVD06] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor. *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM, March 2006.

[RPG98] J. Rosen, H. Park, and J. Glick. Structured total least norm for nonlinear problems. *SIAM J. Matrix Anal. Appl.*, 20(1):14–30, 1998.

[Set99] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[Söd07] T. Söderström. Errors-in-variables methods in system identification. *Automatica*, 43:939–958, 2007.

[SS89] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.

[SSM99] B. Schölkopf, A. Smola, and K. Müller. *Kernel principal component analysis.*, pages 327–352. MIT Press, Cambridge, MA, 1999.

[TB99] M. Tipping and C. Bishop. Probabilistic principal component analysis. *J. R. Stat. Soc. B*, 61(3):611–622, 1999.

[Van97] S. Van Huffel, editor. *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*. SIAM, Philadelphia, 1997.

[Wil86] J. C. Willems. From time series to linear system—Part II. Exact modelling. *Automatica*, 22(6):675–694, 1986.

[Wil87] J. C. Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling. *Automatica*, 22, 23:561–580, 675–694, 87–115, 1986, 1987.

[Wil07] J. C. Willems. The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking. *Control Systems Magazine*, 27:46–99, 2007.

[ZZ05] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM J. on Scientific Computing*, 26:313–338, 2005.