Numerical Algorithms
https://doi.org/10.1007/s11075-025-02307-2

RESEARCH

An alternative variable projection formulation for small-size separable models

Costanza Conti[1] · **Annie Cuyt**[2] · **Wen-shin Lee**[2]

Received: 31 December 2024 / Accepted: 26 December 2025

Abstract
We consider several small-size Prony-like separable least squares interpolation problems in exponential, polynomial and mixed data fitting schemes, for which the variable projection objective function can easily be written down in closed form. An alternative formulation of variable projection is then applied to the closed form expression, yielding explicit formulas for the unknown parameters. The technique is especially useful in various exponential type spline generalizations where the free frequency parameter is otherwise mostly determined by trial and error.

Keywords Non-polynomial splines · Exponential splines · Parameter selection · Subdivision scheme · Prony problem · Variable projection

1 Introduction

Exponential analysis, also called Prony's method [1] or sparse interpolation, fits a linear combination of certain parameterized elementary or special functions [3] $b_j(\phi_j; x), j = 1, \ldots, N$ with constant coefficients $A_j, j = 1, \ldots, N$ to data y_k collected at regularly spaced sample points x_k for $k = 0, \ldots, M-1$ with $M \geq 2N$. Often

✉ Wen-shin Lee
wen-shin.lee@stir.ac.uk

Costanza Conti
costanza.conti@unifi.it

Annie Cuyt
annie.cuyt@stir.ac.uk

[1] Dipartimento di Ingegneria Industriale, University of Firenze, viale Mogagni 40/44, Firenze 50139, Italy

[2] Division of Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, Scotland, UK

Published online: 25 May 2026

used functions $b_j(\phi_j; x)$ are $\exp(\phi_j x)$, $\cos(\phi_j x)$, $\sin(\phi_j x)$, $\cosh(\phi_j x)$, $\sinh(\phi_j x)$, x^{ϕ_j}, $x\exp(\phi_j x), \ldots$ where the ϕ_j and A_j can be real or complex.

Similar models are encountered in subdivision schemes [4] and in generalisations of the classical polynomial splines, such as trigonometric splines [5], hyperbolic splines [6], tension splines [7], exponential splines [8, 9] and so on. The motivation for these generalisations is to better preserve the inherent shape present in the data, such as local convexity, monotonicity or sharp changes in the data. Often $N = 4$ as in the cubic spline case and the four functions $b_j(\phi_j; x)$ are trigonometric, hyperbolic or general exponential elements, all with a free frequency parameter. Even more general are the Chebyshevian splines, of which the previous are all a special case [10, 11]. The use of these generalized splines ranges from geometric design [12] to signal processing [13] and finance [14].

In exponential analysis, the parameters are generally retrieved in a two-step procedure, dealing with the nonlinear and the linear parameters separately. In a first step, the nonlinear parameters ϕ_j are retrieved from a generalized eigenvalue problem [3, 15]. In the second step, the linear coefficients A_j are obtained from a structured linear system. A drawback of this method is the fact that the interpolation data need to be collected equidistantly on the real line or the complex circle.

Variable projection [16] applies to so-called separable problems, in which a linear combination of simple functions characterised by some nonlinear parameters, such as the above models, is fitted to data $y_k, k = 0, \ldots, M-1$ that are not necessarily collected uniformly. In a first step, the nonlinear parameters $\phi_1, \ldots, \phi_N$ are obtained separately via an optimisation procedure. In the second step, the linear coefficients $A_1, \ldots, A_N$ are retrieved as the solution of a linear least squares problem.

The optimisation step is iterative and requires that the linear coefficients are substituted at each step by their solution at that point. A drawback is that the method, when applied to higher-dimensional problems, easily gets stuck in a local minimum, unless one can supply a quite accurate starting point for the optimisation.

In this paper we combine the best of both worlds. For Prony-type problems of small known size N, we consider either a mixture of a small number of regularly sampled data and a larger number of non-uniformly collected data or solely non-uniformly collected data. The considered N-term models depend on at most one unknown nonlinear parameter ϕ, meaning that a number of the ϕ_j are fixed and the remaining ϕ_j all equal ϕ. In the context of subdivision schemes or generalized splines, ϕ is a frequency parameter and a crucial problem is the preliminary selection of that free parameter and its multiplicity, since usually there is no a priori knowledge available. Even though some efforts have been made to provide selection techniques, they are quite involved and time consuming. We refer to [17] for a strategy based on annihilation operators, and [18] for one based on Tikhonov regularization adapted to hyperbolic penalized splines.

In a first step of our novel approach, the value of ϕ is determined. In the second step, the values of the constant coefficients A_j are obtained from some explicit formulas. Because we allow non-uniform data collection, which makes the procedure more applicable in practice, and since we allow coalescent ϕ_j parameters, the mentioned reformulation as a generalized eigenvalue problem is ruled out, also for the subset of uniform data.

In Section 2 some background material is summarized. A theoretical derivation of the new approach is presented in Section 3, while a numerical implementation is given in Section 4. The new approach is illustrated with several synthetic and real-life examples in Section 5. We close with a Conclusion and an accompanying Appendix.

2 Reproducible spaces

Let us define for given $N \in \mathbb{Z}^+$ and a nonzero $\mathrm{a} = (a_0, \ldots, a_N) \in \mathbb{C}^{N+1}$ the function space

$$V_{N,\mathrm{a}} = \left\{ f : \mathbb{R} \to \mathbb{C} : \sum_{j=0}^{N} a_j f^{(j)}(x) = 0 \right\},$$

where $f^{(j)}$ denotes the j-th derivative of the function f.

2.1 Characterisation

A characterisation of this function space is the following [4]. Let z_ℓ be a zero of multiplicity n_ℓ of the polynomial

$$p(z) = \sum_{j=0}^{N} a_j z^j = a_N \prod_{\ell=1}^{L} (z - z_\ell)^{n_\ell},$$

with $n_1 + \cdots + n_L = N$. Then

$$V_{N,\mathrm{a}} = \mathrm{span}\left\{ x^m \exp(z_\ell x) \text{ with } m = 0, \ldots, n_\ell - 1 \text{ and } \ell = 1, \ldots, L \right\}.$$

Now let $x_k = x_0 + k\Delta$ with $\Delta \in \mathbb{R}^+$ and $k = 0, \ldots, M-1 \geq 2N-1$ be some real interpolation points of $f \in V_{N,\mathrm{a}}$ with $y_k = f(x_k)$. Then the polynomial $p(z)$ is closely related to the monic Prony polynomial [19] given by

$$q(z) = \prod_{\ell=1}^{L} \left(z - \exp(z_l \Delta)\right)^{n_\ell} = \sum_{j=0}^{N-1} b_j z^j + z^N,$$

which plays a crucial role in the original Prony scheme [1, 2]: its coefficients b_j can directly be extracted from the samples y_k as in (2) below.

The zeroes z_l of $p(z)$ and their multiplicities n_l can also be found from the use of a so-called annihilation operator [17]. And the zeroes $\exp(z_\ell \Delta)$ of $q(z)$ can be retrieved as the generalized eigenvalues of a Hankel, Toeplitz+Hankel or otherwise structured generalized eigenvalue problem [3, 15].

In addition, the differential equation of order N in the definition of $V_{N,\mathrm{a}}$,

$$\left(a_0 f + a_1 f' + \ldots + a_N f^{(N)}\right)(x) = 0,$$

is closely related to the finite difference equation of order N,

$$y_{N+k} + b_{N-1} y_{N-1+k} + \ldots + b_0 y_k = 0, \qquad k = 0, \ldots, N-1,$$

satisfied by the equidistantly collected samples $y_k = f(x_k)$ of $f \in V_{N,\mathrm{a}}$ [1, 2]. The former terminology mainly stems from the subdivision literature, while the latter appears in the Prony literature.

The unambiguous extraction of z_ℓ from $\exp(z_\ell \Delta)$ requires the so-called Nyquist constraint $\max_{\ell=1,\ldots,L} |\Im(z_\ell)\Delta| < \pi$ to hold, where $\Im(\cdot)$ denotes the imaginary part. So the transition between the zeroes of $p(z)$ and those of $q(z)$ does not hold unconditionally, unless $z_\ell \in \mathbb{R}$.

Some interesting examples of the function spaces $V_{N,\mathrm{a}}$ that we study later in the paper are:

1) $V_{N,\mathrm{a}} = \operatorname{span}\{1, x, \ldots, x^n, \exp(\phi x), \exp(-\phi x)\}$ where $p(z) = z^{n+3} - \phi^2 z^{n+1}$ with $(z_1, n_1) = (\phi, 1), (z_2, n_2) = (-\phi, 1), (z_3, n_3) = (0, n+1)$ and $N = n + 3$,
2) $V_{N,\mathrm{a}} = \operatorname{span}\{1, x, \exp(\phi x), \exp(-\phi x), \ldots, \exp(n\phi x), \exp(-n\phi x)\}$ where $p(z) = z^2 \prod_{j=1}^{n} (z^2 - j^2\phi^2)$ with $(z_1, n_1) = (0, 2), (z_{2j}, n_{2j}) = (j\phi, 1), (z_{2j+1}, n_{2j+1}) = (-j\phi, 1)$ and $N = 2n + 2$.

Special cases of the former example are:

- $\operatorname{span}\{1, x, \ldots, x^{n+2}\}$ when $\phi = 0$,
- $\operatorname{span}\{1, x, \ldots, x^n, \cos(\phi x), \sin(\phi x)\}$ when ϕ is imaginary, and
- $\operatorname{span}\{1, x, \ldots, x^n, \cosh(\phi x), \sinh(\phi x)\}$ when ϕ is real.

Special cases of the latter example are:

- $\operatorname{span}\{1, x, \ldots, x^{2n+1}\}$ when $\phi = 0$,
- $\operatorname{span}\{1, x, \cos(\phi x), \sin(\phi x), \ldots, \cos(n\phi x), \sin(n\phi x)\}$ when ϕ is imaginary, and
- $\operatorname{span}\{1, x, \cosh(\phi x), \sinh(\phi x), \ldots, \cosh(n\phi x), \sinh(n\phi x)\}$ when ϕ is real.

Additional popular choices are:

- $\operatorname{span}\{\exp(\phi x), x\exp(\phi x), \exp(-\phi x), x\exp(-\phi x)\}$ for exponential splines,
- $\operatorname{span}\{\cos(\phi x), x\cos(\phi x), \sin(\phi x), x\sin(\phi x)\}$ for trigonometric splines,
- $\operatorname{span}\{\cosh(\phi x), x\cosh(\phi x), \sinh(\phi x), x\sinh(\phi x)\}$ for hyperbolic splines.

Note that all the above function spaces are spanned by what are essentially exponential functions with either real or imaginary parameters ϕ.

2.2 Reproduction

It is well-known from the Prony literature [20, p. 176] and the subdivision literature [4], that a function $f \in V_{N,\mathrm{a}}$ can be reproduced from (at least) $2N$ of its uniformly collected samples $y_k = f(x_k), k = 0, \ldots, M-1 \geq 2N-1$.

The easiest way to see this is from the nonlinear system (1) of interpolation conditions in the particular case of equidistant points $x_k = x_0 + k\Delta$ with $y_k = f(x_k), k = 0, \ldots, 2N-1$ where $f \in V_{N,\mathrm{a}}$. For simplicity, but without loss of generality, we consider the case of simple zeroes $p(z_\ell) = 0, \ell = 1, \ldots, N$.

At the sample collection, only the function values y_k are known and neither N, a nor the characterisation $V_{N,\mathrm{a}} = \mathrm{span}\{\exp(z_l x), l = 1, \ldots, N\}$ need to be given explicitly. So essentially, f or more generally $V_{N,\mathrm{a}}$, can even be a black box as in Section 5.4. The system of $2N$ interpolation conditions

$$\sum_{\ell=1}^{N} A_\ell \exp(z_\ell x_k) = y_k, \qquad k = 0, \ldots, 2N-1, \tag{1}$$

which is linear in the unknown coefficients $A_1, \ldots, A_N$ and nonlinear in the unknown parameters $z_1, \ldots, z_N$, is re-expressed in two linear systems [1, 2]:

- The first linear system,

$$\sum_{\ell=0}^{N-1} y_{\ell+k} b_\ell = -y_{N+k}, \qquad k = 0, \ldots, N-1, \tag{2}$$

  with Hankel structured coefficient matrix

$$H_N^{(0)} = \begin{pmatrix} y_0 & y_1 & \cdots & y_{N-1} \\ y_1 & & & \vdots \\ \vdots & \iddots & & \\ y_{N-1} & \cdots & & y_{2N-2} \end{pmatrix},$$

  reveals the vector $(b_0, \ldots, b_{N-1}, 1)$ from which $q(z)$ can be constructed. This step remains valid in case the zeroes $\exp(z_\ell \Delta)$ of $q(z)$ are not all simple, because $q(z)$ is the characteristic polynomial of a homogeneous linear recurrence with constant coefficients [21].

  When some coefficients A_j equal zero, then (2) may not have maximal rank N but rather rank $n < N$. In that case, any $n \times n$ Hankel structured submatrix of $H_N^{(0)}$ has rank n and so (2) can be replaced by a smaller size version. The rank of the coefficient matrix of (2) reveals the exact number of nonzero terms in (1). From the zeroes of $q(z)$, the values $z_\ell, \ell = 1, \ldots, N$ are obtained, if the Nyquist condition is satisfied for the $z_\ell \in \mathbb{C}$.

  In the situation where $M > 2N$ samples are available, we can consider an over-

determined version of (2), by extending the Hankel matrix to the rectangular $(M-N)\times N$ matrix $H^{(0)}_{M-N,N}$, extending the right hand side to y_{M-1} and solve (2) in the least squares sense.
It is known that (2) can be quite noise sensitive, both as a square or an overdetermined linear system, which we illustrate in Section 5. We therefore do not make (2) an essential step in our new approach.

- The second linear system results from substituting the computed z_ℓ in the Vandermonde structured linear system (1) or its $2N\times n$ smaller version of maximal column rank. When all zeroes are simple and all A_j nonzero, then (1) has rank N [22]. Again the Vandermonde system can be considered in a least squares sense if $M>2N$ samples are provided.

We now explain how explicit expressions for the linear coefficients A_j and for the objective function delivering the optimal nonlinear parameter ϕ, can be obtained for several $V_{N,\mathrm{a}}$, thereby bypassing the classical two-step procedure for (1) and the classical variable projection algorithm (3). At the same time, we overcome the restriction for all the interpolation values f_k to be uniformly sampled. As pointed out, the method is particularly suitable for small-size problems, meaning small N, and appears to be quite robust.

3 Alternative separable nonlinear least squares

We take our inspiration from the original method of variable projection [16] to allow for non-uniformly collected samples $y_k, k=0,\ldots,M-1\geq 2N-1$. No further restrictions on the data collection are imposed. In the current section there is no need to have a smaller subset of equidistantly collected samples.

3.1 Original variable projection idea

Let us denote

$$A:=\begin{pmatrix}A_1\\ \vdots\\ A_N\end{pmatrix},\quad Z:=\begin{pmatrix}\exp(z_1x_0) & \cdots & \exp(z_Nx_0)\\ \vdots & & \vdots\\ \exp(z_1x_{M-1}) & \cdots & \exp(z_Nx_{M-1})\end{pmatrix},\quad Y:=\begin{pmatrix}y_0\\ \vdots\\ y_{M-1}\end{pmatrix},$$

which are respectively of size $N\times 1, M\times N$ and $M\times 1$. So the nonlinear system (1) is replaced by the more general $ZA=Y$ which does not require that the values $y_k=f(x_k)$ are sampled at equidistant x_k. It is solved by minimising

$$R(A_1,\ldots,A_N;z_1,\ldots,z_N)=||Y-ZA||_2^2.$$

Remember that the parameters $z_\ell, \ell = 1, \ldots, N$ as well as the coefficients $A_j, j = 1, \ldots, N$ are unknown. In a first step, the nonlinear parameters z_ℓ are obtained by minimising

$$R(A_1, \ldots, A_N; z_1, \ldots, z_N) = ||Y - (ZZ^\dagger)Y||_2^2, \tag{3}$$

where $Z^\dagger$ is the Moore-Penrose generalized inverse of Z and the unkown vector A is replaced by $Z^\dagger Y$ at each estimate of the z_ℓ, so that the minimisation problem only involves the parameters z_ℓ. In a second step the coefficients A_j are computed from the linear least squares problem $ZA = Y$.

In our new approach we make use of the variable projection idea in a different way, which is especially suitable for use with function spaces $V_{N,\mathrm{a}}$ that are fully specified by some positive integer N and a single real or imaginary parameter ϕ. Examples of such $V_{N,\mathrm{a}}$ are given in 1) and 2) in Section 2. We reduce the two-step minimisation of $R(A_1, \ldots, A_N; z_1, \ldots, z_N)$ to a one-dimensional minimisation over ϕ and explicit formulas for the $A_j, j = 1, \ldots, N$. In the next section this theoretical reformulation is further simplified to obtain an easy numerical procedure.

3.2 Alternative variable projection formulation

From hereon we consider a data model of the form

$$\sum_{j=1}^{N} A_j b_j(\phi; x)$$

where all of the functions $b_j(\phi; x)$ at most depend on the same nonlinear parameter ϕ. The function spaces $V_{N,\mathrm{a}}$ in 1) and 2) are perfect illustrations of this situation. But it is easy to write down other popular function spaces, which we use in Section 5. At this point, neither the coefficients A_j nor the parameter ϕ are known. Only the number N and the functions $b_j(\phi; x)$ are chosen. Furthermore, we have at our disposal the data $(x_k, y_k), k = 0, \ldots, M-1 \geq 2N-1$. The values y_k are samples of a function $f \in V_{N,\mathrm{a}}$ taken at points $x = x_k$, which are not necessarily equidistant.

Then the objective function is

$$R(A_1, \ldots, A_N; \phi) = \sum_{k=0}^{M-1} \left(y_k - \sum_{j=1}^{N} A_j b_j(\phi; x_k) \right)^2. \tag{4}$$

For given ϕ it is a quadratic N-variate polynomial in the variables $A_1, \ldots, A_N$. This objective function can be rewritten for some $r_1(A_2, \ldots, A_N; \phi)$ and $R_1(A_2, \ldots, A_N; \phi)$ as

$$
\begin{aligned}
R &= \sum_{k=0}^{M-1} \left(y_k - \sum_{j=1}^{N} A_j b_j(\phi; x_k) \right)^2 \\
&= \left(\sum_{k=0}^{M-1} y_k^2 \right) + \alpha_1(\phi) A_1^2 - 2\beta_1(A_2, \ldots, A_N; \phi) A_1 + r_1(A_2, \ldots, A_N; \phi) \\
&= \left(\sum_{k=0}^{M-1} y_k^2 \right) + \alpha_1(\phi) \left(A_1 - \frac{\beta_1(A_2, \ldots, A_N; \phi)}{\alpha_1(\phi)} \right)^2 + R_1(A_2, \ldots, A_N; \phi), \quad (5)
\end{aligned}
$$

with

$$
\begin{aligned}
\alpha_1(\phi) &= \sum_{k=0}^{M-1} b_1^2(\phi; x_k), \\
\beta_1(A_2, , \ldots, A_N; \phi) &= \sum_{k=0}^{M-1} b_1(\phi; x_k) \left(y_k - \sum_{j=2}^{N} A_j b_j(\phi; x_k) \right), \\
R_1(A_2, \ldots, A_N; \phi) &= r_1(A_2, \ldots, A_N; \phi) - \frac{\beta_1^2(A_2, \ldots, A_N; \phi)}{\alpha_1(\phi)}.
\end{aligned}
$$

This step can now be repeated for $R_1(A_2, \ldots, A_N; \phi)$ since it is a quadratic expression in the remaining $A_2, \ldots, A_N$. We thus deliver expressions for $\alpha_2(\phi), \beta_2(A_3, \ldots, A_N; \phi)$ and $R_2(A_3, \ldots, A_N; \phi)$. In the end, the objective function is recursively re-expressed as

$$
R(A_1, \ldots, A_N; \phi) = \sum_{k=0}^{M-1} y_k^2 + \sum_{j=1}^{N} \alpha_j(\phi) \Big(A_j - \beta_j(A_{j+1}, \ldots, A_N; \phi)/\alpha_j(\phi) \Big)^2 + R_N(\phi)\,. \quad (6)
$$

A recursive expression for the $\alpha_j(\phi)$ is easily obtained. Let us denote $B_j := (b_j(\phi; x_0), \ldots, b_j(\phi; x_{M-1})), j = 1, \ldots, N$. With $\alpha_1(\phi) = ||B_1||_2^2$ as given above, we introduce more generally

$$
\alpha_1^{(h,\ell)}(\phi) := \langle B_h, B_\ell \rangle, \qquad h, \ell = 1, \ldots, N
$$

so that $\alpha_1(\phi) = \alpha_1^{(1,1)}(\phi)$. From the computation of R_1 we find that

$$
\alpha_2(\phi) = \alpha_1^{(2,2)}(\phi) - \frac{\left(\alpha_1^{(1,2)}(\phi) \right)^2}{\alpha_1^{(1,1)}(\phi)}.
$$

Again we generalize the definition of $\alpha_2(\phi)$ to

$$
\alpha_2^{(j,h)}(\phi) := \alpha_1^{(j,h)}(\phi) - \frac{\alpha_1^{(1,j)}(\phi) \alpha_1^{(1,h)}(\phi)}{\alpha_1^{(1,1)}(\phi)}
$$

so that $\alpha_2(\phi) = \alpha_2^{(2,2)}(\phi)$. It is easy to see that the recursive definition of the $\alpha_j(\phi)$ continues in the same style.

Lemma Let

$$\alpha_1^{(h,\ell)}(\phi) := \langle B_h, B_\ell \rangle, \qquad h, \ell = 1, \ldots, N$$

and

$$\alpha_{j+1}^{(h,\ell)}(\phi) = \alpha_j^{(h,\ell)}(\phi) - \frac{\alpha_j^{(j,h)}(\phi)\alpha_j^{(j,\ell)}(\phi)}{\alpha_j^{(j,j)}(\phi)}, \qquad j = 1, \ldots, , N-1. \tag{7}$$

Then

$$\alpha_j(\phi) = \alpha_j^{(j,j)}(\phi), \qquad j = 1, \ldots, N.$$

Proof The proof is by induction. We have seen that (7) holds for $j = 1$. For $j = 1, \ldots, N-1$, the coefficient $\alpha_{j+1}(\phi)$ of A_{j+1}^2 in $R_j(A_{j+1}, \ldots, A_N; \phi)$ is the sum of the coefficient of A_{j+1}^2 in $r_j(A_{j+1}, \ldots, A_N; \phi)$ and that in $-\beta_j^2(A_{j+1}, \ldots, A_N; \phi)/\alpha_j(\phi)$, more precisely

$$\alpha_j^{(j+1,j+1)}(\phi) - \frac{\left(\alpha_j^{(j,j+1)}(\phi)\right)^2}{\alpha_j^{(j,j)}(\phi)}.$$

□

It is obvious that all $\alpha_1^{(h,h)}(\phi) \geq 0$. From the Cauchy-Schwarz identity it is also obvious that all $\alpha_2^{(h,h)} \geq 0$. Intensive random experiments using exact arithmetic for various N and $M \geq 2N$ with $B_j \in (\mathbb{Q} + i\mathbb{Q})^M, j = 1, \ldots, N$ as well as all numerical examples in Section 5, only returned positive $\alpha_j(\phi), j = 1, \ldots, N$.

With

$$A_j = \beta_j(A_{j+1}, \ldots A_N; \phi)/\alpha_j(\phi), \qquad j = N, \ldots, 1,$$

in (6), the parameter ϕ is found as

$$\phi = \arg\min_{\phi} R_N(\phi),$$

where $R_N(\phi)$ may take either positive or negative values. In case of small-sized $V_{N,\mathrm{a}}$ the closed-form expression for $R_N(\phi)$ is not a complicated formula. The minimum value of the objective function $R(A_1, \ldots, A_N; \phi)$ is given by

$$\min_{A_1,\ldots,A_N,\phi} R(A_1, \ldots, A_N; \phi) = \sum_{k=0}^{M-1} y_k^2 + \min_{\phi} R_N(\phi).$$

We summarize the algorithm in the following scheme and illustrate its immediate use in the small-size example of Section 5.1. For ease of notation, we omit some dependencies in the algorithmic part, since no ambiguity can arise.

Algorithm A: Alternative separable nonlinear least squares algorithm.

Input: L, n_ℓ for $\ell = 1, \ldots, L$ and $N = n_1 + \cdots n_L$
$(x_k, y_k), k = 0, \ldots, M-1$
$b_j(\phi; x), j = 1, \ldots, N$

Goal: $\min_{A_j,\phi} R(A_1, \ldots, A_N; \phi)$ with

$$R(A_1, \ldots, A_N; \phi) = \sum_{k=0}^{M-1} \left(y_k - \sum_{j=1}^{N} A_j b_j(\phi; x_k) \right)^2$$

Method: rewrite $R(A_1, \ldots, A_N; \phi)$

$$= \sum_{k=0}^{M-1} y_k^2 + \sum_{j=1}^{N} \alpha_j(\phi) \Big(A_j - \beta_j(A_{j+1}, \ldots, A_N; \phi)/\alpha_j(\phi) \Big)^2 + R_N(\phi)$$

Output: $\alpha_j(\phi), \beta_j(A_{j+1}, \ldots, A_N; \phi), j = 1, \ldots, N$
$R_N(\phi)$

Iteration:

$R_0 := R(A_1, \ldots, A_N; \phi) - \sum_{k=0}^{M-1} y_k^2$

FOR $j = 0, \ldots, N-1$

DO expand R_j into quadratic N-variate polynomial in $A_1, \ldots, A_N$

collect $R_j = \alpha_{j+1} A_{j+1}^2 - 2\beta_{j+1} A_{j+1} + r_{j+1}$

$$= \alpha_{j+1} \left(A_{j+1} - \frac{\beta_{j+1}}{\alpha_{j+1}} \right)^2 + \left(r_{j+1} - \frac{\beta_{j+1}^2}{\alpha_{j+1}} \right)$$

$R_{j+1} := r_{j+1} - \dfrac{\beta_{j+1}^2}{\alpha_{j+1}}$

ENDDO

$\min_{A_j,\phi} R(A_1, \ldots, A_N; \phi) = \sum_{k=0}^{M-1} y_k^2 + \min_{\phi} R_N(\phi)$

In Fig. 1 we show the original variable projection expression (3), at the left hand side, and

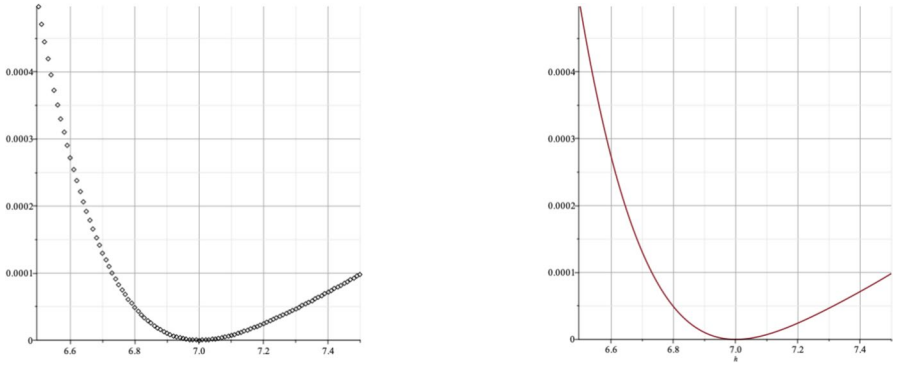


Fig. 1 Graph of (3) at the left and (8) at the right for function (9)

$$\sum_{k=0}^{M-1} y_k^2 + R_N(\phi), \tag{8}$$

at the right hand side, for $M = 40$ randomly selected x_k-values in $[-1.5, 1.5]$ and $y_k = f(x_k)$ with

$$\begin{aligned} f(x) &= 0.0001\big(\exp(7x) - x\exp(-7x)\big) \\ &\in V_{4,(7^4,0,-2\times 7^2,0,1)} = \operatorname{span}\{\exp(7x), x\exp(7x), \exp(-7x), x\exp(-7x)\}. \end{aligned} \tag{9}$$

Formula (3) at the left is computed for discrete values of $\phi \in [6.5, 7.5]$ with a step size of 0.01, while the graph at the right shows the closed-form expression (8).

4 Alternative variable projection programmed

The algorithm in Section 3 can be programmed in a computer algebra environment or, using a down-sized symbolic toolbox, in a numerical programming environment, since it essentially only manipulates a quadratic polynomial in the $A_1, \ldots, A_N$ with coefficients that are parameterized by ϕ. We supply a `Matlab` implementation in the appendix. For the small-sized $V_{N,\mathrm{a}}$ that we consider, a two-step implementation is given in this section.

The challenge of finding the optimal value ϕ, knowing the functions $b_j(\phi; x), j = 1, \ldots, N$, can also be solved solely by inspecting the coefficients of the Prony polynomial $q(z)$. To that end we need a small number of $M_1 \geq 2N - 1$ equidistantly collected samples $y_k = f(k\Delta), k = 0, \ldots, M_1 - 1$. The computation of ϕ is further discussed in Section 4.1.

Subsequently the values for the linear coefficients A_j can be obtained from some easy vector and matrix updates, from the M_1 uniform samples and an additional M_2 non-uniform samples, where usually $M_2 \geq M_1$. This step then replaces the solution of the Vandermonde structured system (1), in the least squares sense when $M > 2N$. The new algorithm is given in Section 4.2 and summarized in a recursive scheme.

The method of Section 4.1 to extract ϕ can be numerically sensitive, as we illustrate in Section 5.2. Therefore this step is only recommended as a replacement for Algorithm A, when the data are sufficiently accurate. It can however also serve to supply a starting point for the optimisation of $R_N(\phi)$.

In the sequel we denote $M = M_1 + M_2$ when we refer to the mixture of the uniform and non-uniform samples.

4.1 Nonlinear parameter ϕ

Let us first discuss how to determine the single nonlinear parameter ϕ for the small-sized models under consideration, from the uniformly collected $y_0, \ldots, y_{M_1-1}$. We deal with the function spaces $V_{N,\mathrm{a}}$ given in 1) and 2) in Section 2.

For $V_{N,\mathrm{a}} = \mathrm{span}\{1, x, \ldots, x^n, \exp(\phi x), \exp(-\phi x)\}$ where $N = n + 3$, the polynomial $q(z)$ is given by

$$q(z) = (z - \exp(\phi\Delta))(z - \exp(-\phi\Delta))(z-1)^{n+1}.$$

Its constant term $b_0 = (-1)^{n+1}$. Its linear term

$$b_1 = (-1)^n\big(n + 1 + \Phi + 1/\Phi\big), \qquad \Phi \in \{\exp(\phi\Delta), \exp(-\phi\Delta)\}.$$

Here we want $A_{n+1}A_{n+2}A_{n+3} \neq 0$ which can be confirmed by computing the rank of $H^{(0)}_{M_1-N-1,N}$. If it is maximal, the rank confirms that the terms $x^n, \exp(\phi x), \exp(-\phi x)$ have nonzero coefficients. We'll deal with non-maximal rank separately in a moment. From the value of b_1 we obtain the quadratic equation

$$\Phi^2 + \Phi\big(n + 1 + (-1)^{n+1}b_1\big) + 1 = 0,$$

with the reciprocal solutions $\exp(\phi\Delta)$ and $\exp(-\phi\Delta)$.

If the rank of $H^{(0)}_N$ is $N-1$, then one of the three coeffients $A_{n+1}, A_{n+2}, A_{n+3}$ is zero. The constant term will tell you which one:

- If $A_{n+1} = 0$ then $b_0 = (-1)^n$ and $b_1 = (-1)^{n-1}\big(n + \Phi + 1/\Phi\big)$.
- If $A_{n+2} = 0$ then $b_0 = (-1)^{n+1}/\Phi$ and similarly if $A_{n+3} = 0$ then $b_0 = (-1)^{n+1}\Phi$. So b_0 tells you which term is absent and identifies Φ at the same time.

If the rank of $H^{(0)}_N$ is $N-2$, then either two of these three coefficients are zero or $A_n = 0 = A_{n+1}$:

- In the latter case, we have $b_0 = (-1)^{n-1}$ and $b_1 = (-1)^{n-2}\big(n - 1 + \Phi + 1/\Phi\big)$.
- In the former case, we either have $A_{n+1} = 0 = A_{n+2}$ or $A_{n+1} = 0 = A_{n+3}$, similar to the second bullet above but with n replaced by $n-1$, or we have $A_{n+2} = 0 = A_{n+3}$ which is similar to the first bullet above with n replaced by $n-1$.

We don't push this discussion any further because it is now clear how to proceed. Also, a user should not overestimate the value of n or N by too much. Note that the computation of the numerical rank of the Hankel matrix, which is traditionally done using SVD, may still be a delicate problem.

4.2 Linear coefficients $A_1, \ldots, A_N$

Now we turn to the identification of the linear coefficients A_j, thereby following a numerical version of the algorithm in Section 3. For this we use all available samples, the uniformly collected $y_k = f(k\Delta), k = 0, \ldots, M_1 - 1$ and the additional non-uniformly collected $y_{M_1+k} = f(x_k), k = 0, \ldots, M_2 - 1$, where $M = M_1 + M_2$. The rank of $H_N^{(0)}$ does not play a role in this derivation.

Remember that the objective function $R(A_1, \ldots A_N; \phi)$, for a given value ϕ, is a quadratic N-variate polynomial in $A_1, \ldots, A_N$:

$$R(A_1, \ldots A_N; \phi) = \sum_{k=0}^{M-1} y_k^2 - 2\sum_{j=1}^{N} c_j(\phi)A_j + \sum_{\substack{j=1 \\ k\geq j}}^{N} d_{jk}(\phi)A_jA_k.$$

In the sequel we omit to denote the dependence of the expressions $c_j(\phi)$ and $d_j(\phi)$ on ϕ, as the optimal value for ϕ has been determined by now and can hence be substituted in the $c_j(\phi)$ and the $d_{jk}(\phi)$.

Let us store the linear coefficients of the A_j in $R(A_1, \ldots, A_N)$ in the vector

$$C := -2\,(c_1, \ldots, c_N)$$

and the quadratic coefficients in the upper triangular matrix

$$D := \begin{pmatrix} d_{11} & d_{12} & \ldots & d_{1N} \\ 0 & d_{22} & \ldots & d_{2N} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & d_{NN} \end{pmatrix}.$$

Then from (5), we find

$$\alpha_1 = d_{11},$$
$$-2\beta_1(A_2, \ldots, A_N) = -2c_1 + \sum_{j=2}^{N} d_{1j}A_j$$

and hence

$$-\frac{\beta_1^2(A_2,\ldots,A_N)}{\alpha_1} = \frac{-1}{4d_{11}}\left(4c_1^2 - 4c_1\sum_{j=2}^{N} d_{1j}A_j + \sum_{j=2}^{N} d_{1j}^2 A_j^2 + 2\sum_{\substack{j=2\\k>j}}^{N} d_{1j}d_{1k}A_jA_k\right).$$

When moving from the treatment of R_0 in algorithm A to that of R_1, the vector $C^{(0)} := C$ and matrix $D^{(0)} := D$ are updated to

$$C^{(1)} := C^{(0)} + \frac{c_1}{d_{11}}(0, d_{12}, \ldots, d_{1N}),$$

$$D^{(1)} := D - \frac{1}{4d_{11}}\begin{pmatrix} 0 & & \cdots & & & 0 \\ & \ddots & & & & \vdots \\ & & d_{12}^2 & 2d_{12}d_{13} & \cdots & 2d_{12}d_{1N} \\ \vdots & & & & & \vdots \\ & & & & \ddots & \\ 0 & & & \cdots & 0 & d_{1N}^2 \end{pmatrix}.$$

When denoting the vector and matrix entries of $C^{(1)}$ and $D^{(1)}$ respectively by $-2c_j^{(1)}, j = 1,\ldots,N$ and $d_{ij}^{(1)}, i,j = 1,\ldots,N$, then

$$\alpha_2 = d_{22}^{(1)},$$

$$-2\beta_2(A_3,\ldots,A_N) = -2c_2^{(1)} + \sum_{j=3}^{N} d_{2j}^{(1)}A_j$$

and the process can be repeated. Let us summarize this in the following scheme, where we don't give an explicit formula for $\beta_j^2(A_{j+1},\ldots,A_N)$, but

- denote the constant term and the coefficients of the linear terms $A_{j+1},\ldots,A_N$ in β_j^2 by $\gamma_j^{(j-1)}$ and $-2\gamma_k^{(j-1)}, k = j+1,\ldots,N$, respectively, and
- denote the coefficients of the quadratic terms A_kA_ℓ in β_j^2 by $\delta_{k\ell}^{(j-1)}$.

Its use is illustrated in the 4-term example of Section 5.2. In the algorithm we also omit to denote the dependence on ϕ for those expressions where the optimal value for ϕ has been substituted.

Algorithm L: Linear coefficients of separable nonlinear least squares problem.

Input: $(x_k, y_k), k = 0, \ldots, M-1$
optimal ϕ from Section 3 or Section 4.1
L, n_ℓ for $\ell = 1, \ldots, L$ and $N = n_1 + \cdots n_L$
$b_j(x_k), j = 1, \ldots, N$

Goal: compute $A_1, \ldots, A_N$ in model $\displaystyle\sum_{j=1}^{N} A_j b_j(\phi; x)$ for optimal ϕ

Method: rewrite $\min_{A_j,\phi} R(A_1, \ldots, A_N; \phi)$

$$= \sum_{k=0}^{M-1} y_k^2 + \sum_{j=1}^{N} \alpha_j \left(A_j - \beta_j(A_{j+1}, \ldots, A_N)/\alpha_j \right)^2 + \min_{\phi} R_N(\phi)$$

Output: $A_1, , \ldots, A_N$

Iteration:

$$R(A_1, \ldots, A_N) = \sum_{k=0}^{M-1} y_k^2 - 2\sum_{j=1}^{N} c_j^{(0)} A_j + \sum_{\substack{j=1 \\ k \geq j}}^{N} d_{jk}^{(0)} A_j A_k$$

$$C^{(0)} := -2(c_1^{(0)}, \ldots, c_N^{(0)})$$

$$D^{(0)} := \left(d_{jk}^{(0)} \right)_{j=1,k\geq j}^{N}$$

$$\alpha_1 := d_{11}^{(0)}$$

$$-2\beta_1(A_2, \ldots, A_N) := -2c_1^{(0)} + \sum_{j=2}^{N} d_{1j}^{(0)} A_j$$

FOR $j = 1, \ldots, N-1$
DO

$$-\beta_j^2(A_{j+1}, \ldots, A_N) = \gamma_j^{(j-1)} - 2\sum_{k=j+1}^{N} \gamma_k^{(j-1)} A_k + \sum_{\substack{k=j+1 \\ \ell \geq k}}^{N} \delta_{kl}^{(j-1)} A_k A_\ell$$

$$C^{(j)} := C^{(j-1)} - \frac{2}{\alpha_j}(0, \ldots, 0, \gamma_{j+1}^{(j-1)}, \ldots, \gamma_N^{(j-1)})$$

$$D^{(j)} := D^{(j-1)} + \frac{1}{\alpha_j} \begin{pmatrix} 0 & \cdots & & & 0 \\ & \ddots & & & \vdots \\ \vdots & & \delta_{j+1,j+1}^{(j-1)} & \cdots & \delta_{j+1,N}^{(j-1)} \\ & & & \ddots & \vdots \\ 0 & & \cdots & 0 & \delta_{n,n}^{(j-1)} \end{pmatrix}$$

$$\alpha_{j+1} := d_{j+1,j+1}^{(j)}$$

$$-2\beta_{j+1}(A_{j+2}, \ldots, A_N) := -2c_{j+1}^{(j)} + \sum_{k=j+2}^{N} d_{j+1,k}^{(j)} A_k$$

ENDDO

FOR $j = N, \ldots, 1$
DO $A_j := \beta_j(A_{j+1}, \ldots, A_N)/\alpha_j$
ENDDO

5 Various examples

A simple numerical illustration for $N = 2$ which does not need the general recursive schemes of the Algorithms A and L can already be found in [23]. A similar model is now symbolically worked out in Section 5.1

Here we further illustrate the new approach on various function spaces $V_{N,\mathrm{a}}$. We use Algorithm A to identify the nonlinear parameter ϕ and we obtain the linear coefficients from Algorithm L. All subsequent examples are illustrated numerically on noisy data, and code for Algorithm A to reproduce the output is given in the appendix. The numeric optimisation and coefficient computations are performed in double precision arithmetic. The examples in the Sections 5.2 and 5.3 are using synthetic data, while the examples in Section 5.4 are using real-life datasets.

5.1 An easy start with $V_{N,\mathrm{a}} = \mathbf{span}\{\exp(\phi x), \exp(-\phi x)\}$

Here we have $N = 2$ and for $f \in V_{N,\mathrm{a}}$,

$$f(x) = A_1 \exp(\phi x) + A_2 \exp(-\phi x).$$

With $y_k = f(x_k), k = 0, \ldots, M-1$ the objective function is

$$R(A_1, A_2; \phi) = \sum_{k=0}^{M-1} \left(y_k - A_1 \exp(\phi x_k) - A_2 \exp(-\phi x_k)\right)^2.$$

After expanding the expression for $R(A_1, A_2; \phi)$, we find that the coefficient of A_1^2 is given by

$$\alpha_1(\phi) = \sum_{k=0}^{M-1} \exp(2\phi x_k),$$

and the coefficient of $-2A_1$ equals

$$\beta_1(A_2; \phi) = \sum_{k=0}^{M-1} y_k \exp(\phi x_k) - MA_2.$$

From expanding the expression $R_1(A_2; \phi)$ in (5) we obtain

$$\alpha_2(\phi) = \sum_{k=0}^{M-1} \exp(-2\phi x_k) - \frac{M^2}{\sum_{k=0}^{M-1} \exp(2\phi x_k)},$$

$$\beta_2(\phi) = \sum_{k=0}^{M-1} y_k \exp(-\phi x_k) - \frac{M \sum_{k=0}^{M-1} y_k \exp(\phi x_k)}{\sum_{k=0}^{M-1} \exp(2\phi x_k)}$$

and

$$R_2(\phi) = -\frac{\left(\sum_{k=0}^{M-1} y_k \exp(\phi x_k)\right)^2}{\sum_{k=0}^{M-1} \exp(2\phi x_k)} - \frac{\beta_2^2(\phi)}{\alpha_2(\phi)}.$$

The nonlinear parameter ϕ is obtained as $\arg\min_\phi R_2(\phi)$ and the A_1 and A_2 can be obtained either by substituting the computed ϕ in the expressions α_j and β_j above with $A_2 = \beta_2(\phi)/\alpha_2(\phi)$ and $A_1 = \beta_1(A_2;\phi)/\alpha_1(\phi)$.

5.2 Higher multiplicity in $V_{N,a} = \mathbf{span}\{\exp(\pm\phi x), x\exp(\pm\phi x)\}$

We consider the function

$$f(x) = 0.01\Big(3x\exp(5x) + (25x+35)\exp(-5x)\Big)$$

which belongs to $V_{4,(5^4,0,-2\times5^2,0,1)}$. We sample the function at 10 equidistant points in $[-1.5, 1.5]$ and at another 30 random sample points in the interval, taken from a uniform distribution. The values $y_k = f(x_k)$ are then contaminated with additive noise taken from a uniform distribution in $[-0.001, 0.001]$. This results in $||y-\tilde{y}||_2/||y||_2 \approx 2.98\times10^{-5}$, where the vector y denotes the vector of the $y_k, k = 0,\ldots,39$ and $\tilde{y}$ denotes the vector of the 40 noisy datapoints.

From Algorithm A we obtain the estimate $\tilde{\phi} = 5.00006$ for $\phi = 5$. From Algorithm L we subsequently get the model

$$\tilde{f}(x) = 0.0299954\exp(\tilde{\phi}x) + (0.249977x + 0.349968)\exp(-\tilde{\phi}x), \qquad (10)$$

with $||A-\tilde{A}||_2/||A||_2 \approx 9.19\times10^{-5}$ where $A = (A_1, A_2, A_3, A_4)$ and $\tilde{A}$ is the estimate obtained for the coefficient vector A. In Fig. 2 we overlay the input samples $\tilde{y}_k$, the exact function f in red and the computed model (10) in black. Because the noise is not large, both the original f and the reconstructed model overlap.

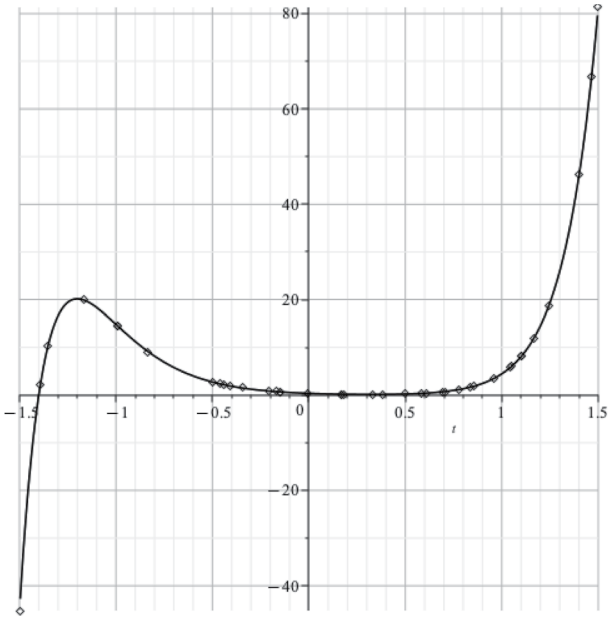


Fig. 2 Input data with functions f (●) and $\tilde{f}$ (●)

Since we have 10 equidistantly collected and rather accurate samples at our disposal, we can also consider the computation of the Prony polynomial $q(z)$ as detailed in Section 4.1. We consider $H_4^{(1)}$ discarding the first and the last sample, as this Hankel matrix is the better conditioned one of size $N = 4$. We thus obtain the estimate $\bar{\phi} = 4.99727$ which is clearly less accurate than the estimate $\tilde{\phi}$. Note that the numerical rank of the noisy matrix is 4 while the mathematical rank of the exact matrix is 3. Switching to an overdetermined system for the coefficients of $q(z)$, with coefficient matrix $H_{6,4}^{(0)}$, does not improve the result. In the sequel we therefore do not use the Prony polynomial anymore to estimate the nonlinear parameter ϕ. We only use Algorithm A instead.

Let us further explore the use of the Algorithms A and L on samples $\hat{y}$ that are much noisier. We first increase the additive noise to be uniformly distributed in $[-1.75, 1.75]$ leading to $||y - \hat{y}||_2/||y||_2 \approx 0.0521$.

Algorithm A now delivers $\hat{\phi} = 5.0499$ as an estimate for $\phi = 5$ with a relative error of 0.00998, and Algorithm L computes the coefficients of the reconstruction

$$\hat{f}(x) = (0.016 + 0.027x)\exp(\hat{\phi}x) + (0.32 + 0.23x)\exp(-\hat{\phi}x), \tag{11}$$

with $||A - \hat{A}||_2/||A||_2 \approx 0.0861$. Again the noisy data, the exact f in red and the computed reconstruction (11) in black are overlaid in Fig. 3 where we zoom in on the critical region of data.

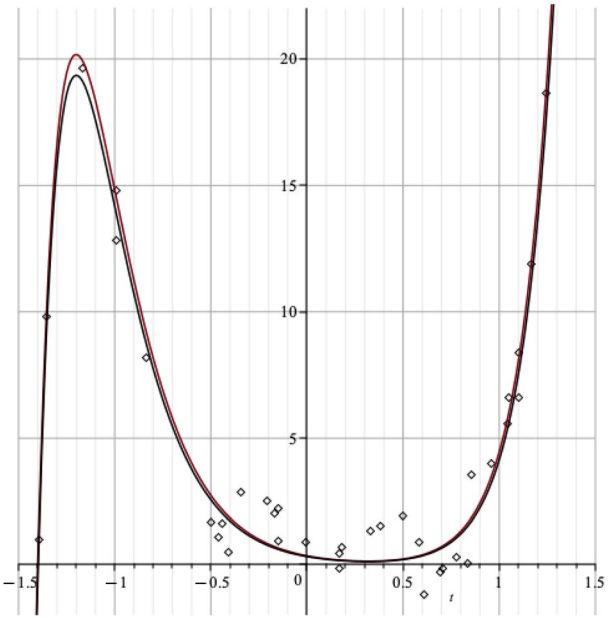


Fig. 3 Data with f (●) and $\hat{f}$ (●)

Let us increase the noise even further, to a uniform distribution in $[-7.5, 7.5]$ with noisy samples $\bar{y}$ where $||y - \bar{y}||_2/||y||_2 \approx 0.223$. Algorithm A has been quite stable in all our experiments, and also here we find $\bar{\phi} = 5.2408$ with a relative error of approximately 0.0482 compared to the exact $\phi = 5$. Algorithm L delivers the coefficients in the reconstruction

$$\bar{f}(x) = (0.0061 + 0.016x)\exp(\bar{\phi}x) + (0.23 + 0.16x)\exp(-\bar{\phi}x) \tag{12}$$

with a relative error $||A - \bar{A}||_2/||A||_2 \approx 0.349$.

A superposition of the noisy data, the exact f in red and the reconstruction (12) in black are given in Fig. 4. Although the error on the data has mainly been suffered by the coefficients, it is clear from Fig. 4 that the graph of the reconstruction is very well following the trend of the given f. We came to the same conclusion even when the relative errors on ϕ and on the coefficient vector A were twice as large. It seems that an accurate computation of $\tilde{\phi}$ is the most important step in the process.

For the sake of completeness we also computed the coefficient vector A from the system of $M = 40$ interpolation conditions (1) in the least squares sense using QR factorisation. The latter solution for A is indistinguishable from the former delivered by Algorithm L. The Euclidean norm of their difference is 2.7×10^{-14}.

We can witness something similar, even in a more pronounced way, in the last example in Section 5.3.

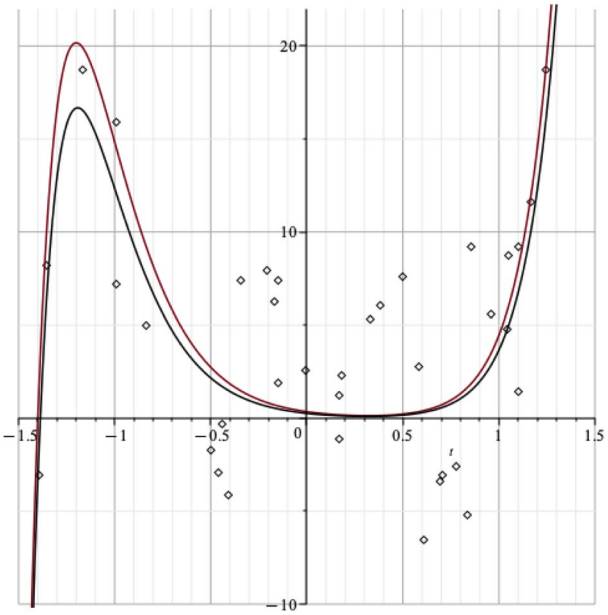


Fig. 4 Data with f (●) and $\bar{f}$ (●)

5.3 Some polynomial flavour in $\mathbf{span}\{1, x, x^\phi, (1-x)^\phi\}$

We consider the functions

$$f_1(x) = 1 + 2.75x - 1.5x^{3.5} + 2(1-x)^{3.5},$$
$$f_2(x) = 1 + 2.75x - 1.5x^{4.1} + 2(1-x)^{4.1},$$

on the interval [0, 1]. These functions are actually out of the scope of Section 2, as they do not belong to a set $V_{N,\mathrm{a}}$, but the Algorithms A and L remain applicable for these Bernstein-like functions $b_j(\phi; x)$. Also, such linear combinations appear in the curves and surfaces literature in the context of shape preservation [7, 24].

We take $\phi > 2$ as the problem statement is singular for $\phi = 0, \phi = 1$ and $\phi = 2$. When running Algorithm A with the functions $b_1(x) = 1, b_2(x) = x, b_3(x) = x^\phi, b_4(x) = (1-x)^\phi$, then the rational expression $R_4(\phi)$ has a multiple pole at $\phi = 0$ and $\phi = 1$ and a pole at $\phi = 2$. In addition, for not very large positive ϕ-values, as in f_1 and f_2, the sign of $R_4(\phi)$ is easily altered in the neighbourhood of the ϕ-parameter, by the noise on the collected data. To see this, we plot in Fig. 5 the denominator of $R_4(\phi)$, computed for some randomly collected noisefree samples of f_2.

We now conduct three experiments to illustrate the use of variable projection in its alternative formulation on samples collected either uniformly or randomly from the functions f_1 and f_2. More experiments than the selected three were conducted, so that we can say that the conclusions hold in general.

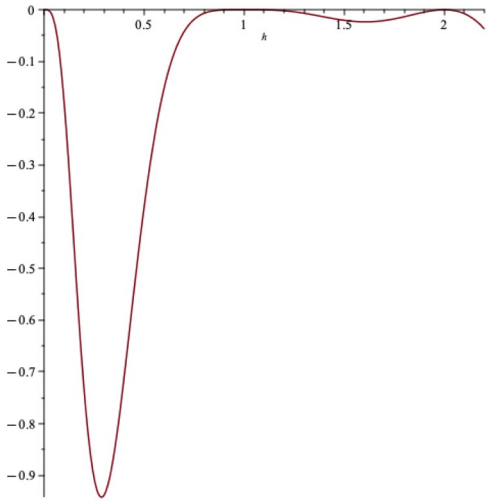


Fig. 5 Noiseless denominator of $R_4(\phi)$ with random data from f_2

First, we equidistantly collect $M = 31$ samples of f_1 with $x \in [0, 1]$ and add uniformly distributed noise in $[-0.2, 0.2]$ to the function values $y_k = f_1(x_k)$. This results in a relative error $||y - \tilde{y}||_2/||y||_2 \approx 0.0443$.

Algorithm A delivers the estimate $\tilde{\phi} = 3.0219$ with a relative error of 0.137 compared to $\phi = 3.5$, and Algorithm L provides the coefficients in the model

$$\tilde{f}_1(x) = -0.452 + 5.63x - 3.01x^{\tilde{\phi}} + 3.59(1 - x)^{\tilde{\phi}}. \tag{13}$$

Although the relative error on the linear coefficients is about 101%, the trend of the considered f_1 is neatly picked up by the model, despite the noisy data. This is illustrated in Fig. 6. Again, for comparison, we also compute the coefficient vector A from the $M = 31$ interpolation conditions (1). In this example the coefficient matrix is a rectangular Vandermonde matrix. The Euclidean norm of the difference between the two computed vectors A is 2.1×10^{-12}.

The fact that distinct sets of parameters $\{\phi, A_1, \ldots, A_N\}$ can display a very similar behaviour, is exactly the reason that the inverse problem of extracting these parameters from the data $y_1, \ldots, y_{M-1}$ can be ill-posed at times [25]. This is particularly well illustrated in what follows, where we consider two experiments with $M = 31$ randomly collected samples of f_2, both experiments using a different random set of x_k-values. The difference between the two examples is that the right limit value $\lim_{\phi \to 2, \phi > 2} R_4(\phi)$ at the pole near $\phi = 2$ (see Fig. 5), switches sign between the two random runs. It equals $-\infty$ in the first random run and $+\infty$ in the second. Note that for the noiseless computation in Fig. 5, $\lim_{\phi \to 2, \phi > 2} R_4(\phi) = +\infty$.

For Fig. 7, the data $\tilde{y}$ differ from y by a relative error of 0.0514 and for Fig. 8 the data $\hat{y}$ differ from y by 0.0454, whereby each uses different uniformly distributed noise terms in $[-0.2, 0.2]$ on distinct samples of f_2. In the first case, Algorithm A delivers $\tilde{\phi} = 8.7963$ while in the second one it delivers $\hat{\phi} = 2.5836$, both clearly being off compared to the correct value $\phi = 4.1$. Algorithm L respectively delivers the models

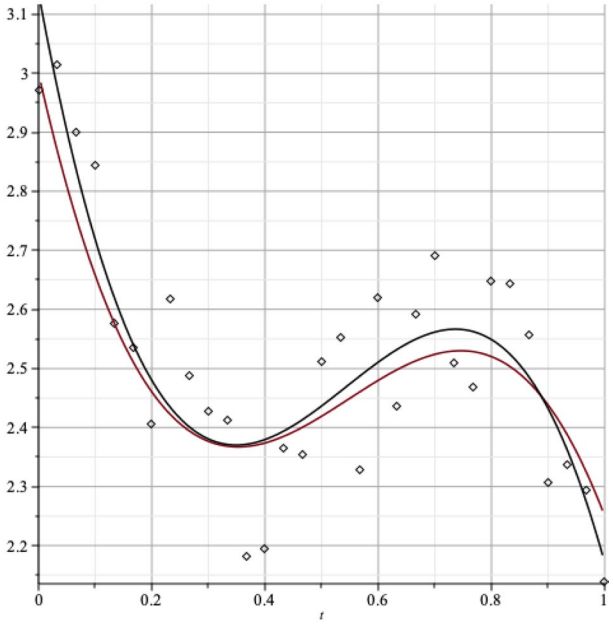


Fig. 6 Input data with functions f_1 (●) and $\tilde{f}_1$ (●)

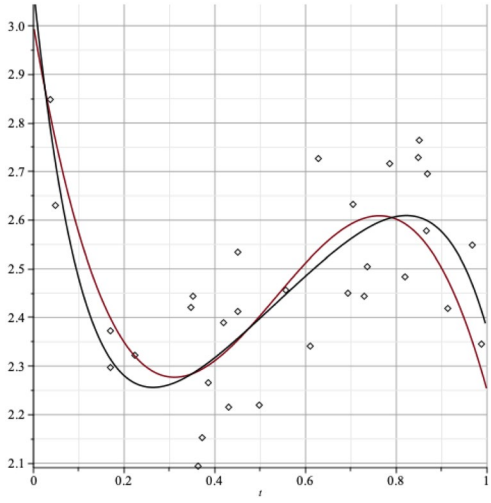
$$\tilde{f}_2(x) = 1.93 + 0.926x - 0.485x^{\tilde{\phi}} + 1.14(1-x)^{\tilde{\phi}}, \tag{14}$$

and

$$\hat{f}_2(x) = -2.64 + 9.92x - 4.89x^{\hat{\phi}} + 5.52(1-x)^{\hat{\phi}}, \tag{15}$$

being off by $||A - \tilde{A}||_2/||A||_2 \approx 63\%$ and $||A - \hat{A}||_2/||A||_2 \approx 244\%$ respectively!

Fig. 7 Data with f_2 (●) and $\tilde{f}_2$ (●)



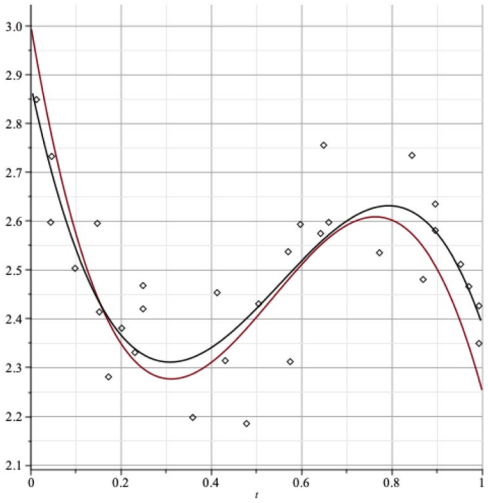


Fig. 8 Data with f_2 (•) and $\hat{f}_2$ (•)

Nevertheless, the models are not meaningless as they pick up the trend of f_2 quite well. It suffices to look at the relative difference (measured in $||\cdot||_2$ norm) between the model (in black) and the truth (in red), evaluated at the sample points x_k used in the Figs. 7 and 8 respectively. This relative difference equals 0.0192 for Fig. 7 and 0.0235 for Fig. 8, about half the relative error introduced by the noise on the y_k before starting the computation.

To conclude and reassure the reader: when running the last two experiments with only some arithmetic noise, then Algorithm A neatly recovers $\phi = 4.1$ up to single precision accuracy.

5.4 Illustration on two real-life datasets

The first dataset is the benchmark `motorcycle` dataset which is given in [26]. This set of $M = 94$ non-uniformly collected samples is also modelled in [27].

In [27] the authors mention that the use of generalized splines with spline pieces belonging to $V_{N,\mathrm{a}} = \mathrm{span}\{\exp(\pm\phi x), x\exp(\pm\phi x)\}$ from Section 5.2, does not add much when compared to using polynomial splines. This is now confirmed by Algorithm A, which returns the optimum frequency $\phi = 0$ when used with the given samples.

The second set consists of the `Nigerian coal production` data over the period 1916–1966, which can be found in [28]. Here $M = 26$ and the data are equidistant, but we do not exploit that, as already indicated at the beginning of Section 5. In the literature the use of $V_{N,\mathrm{a}}$ of Section 5.2 is proposed and so we follow that suggestion. Algorithm A adds a straightforward computation of the unknown parameter ϕ, which mostly remained an open problem. For the coefficients another method is used instead of Algorithm L.

So we model the data with a generalized spline where each spline piece belongs to $V_{N,\mathrm{a}} = \mathrm{span}\{\exp(\pm\phi x), x\exp(\pm\phi x)\}$. After determining the frequency parameter using Algorithm A, the coefficients in the exponential spline are obtained from the minimisation of an objective function consisting of a weighted least squares part, measuring the closeness of the linear combination of the generalized exponential B-spline basis functions to the given data, and a penalization term based on the integral of the squared second derivative of the spline, as in [27].

The resulting generalized spline is shown in Fig. 9 together with the input data (shown as $\diamond$). For the frequency parameter Algorithm A delivers $\phi = 0.07654$. Here the exponential-based model follows the data much more accurately than a traditional penalized cubic spline, shown in turquoise in Fig. 10. The latter excessively smooths the convex and monotone behaviour present in the data.

6 Conclusion

This paper introduces an alternative method for variable projection, especially tailored for small-size separable data fitting problems that depend on one nonlinear real or complex parameter ϕ. The method can be used on either a mixture of M_1 uniformly and M_2 non-uniformly collected samples or solely non-uniformly col-

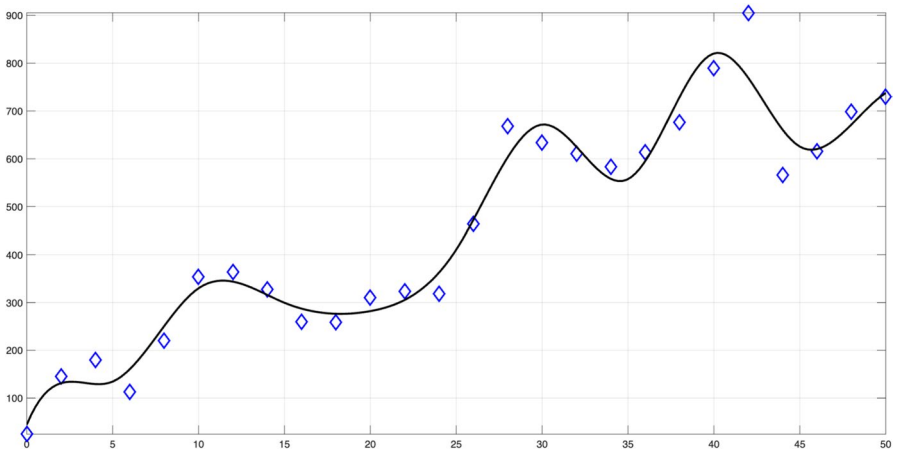


Fig. 9 A penalized exponential spline for Nigerian coal production data

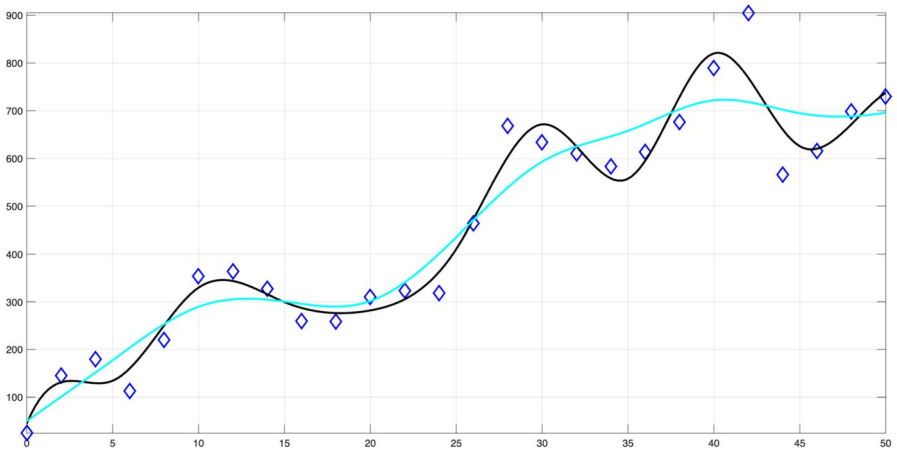


Fig. 10 Comparison to a penalized cubic spline (cyan) for Nigerian coal production data

lected data. We focus in particular on data models with a small but known number $N \leq (M_1 + M_2)/2$ of linear parameters.

The new algorithms A and L respectively estimate the nonlinear parameter ϕ and the linear coefficients in the model. An extensive numerical investigation shows the effectiveness of the novel approach, even under high noise on the data. The applicability of the method is also illustrated on real-life datasets.

Appendix

```
% Symbolic math toolbox Matlab code for Algorithm A
% 5.2 higher multiplicities ...

syms f(x)
f(x) = 0.01*(3*x*exp(5*x)+(25*x+35)*exp(-5*x));

syms R0 phi C00 C11 C12 C13 C14 C22 C23 C24 C33 C34...
C44 C10 C20 C30 C40

N = 4;
R = sym("R_%d",[1 N+1]);
A = sym("A_%d",[1 N]);
alfa = sym("alfa_%d", [1 N+1]);
beta = sym("beta_%d", [1 N+1]);

R(1) = A(1)^2*C11 + 2*A(1)*A(2)*C12 + 2*A(1)*A(3)*C13 + 2*A(1)*A(4)*C14...
+ A(2)^2*C22 + 2*A(2)*A(3)*C23 + 2*A(2)*A(4)*C24 + A(3)^2*C33...
+ 2*A(3)*A(4)*C34 + A(4)^2*C44 - 2*A(1)*C10 - 2*A(2)*C20 - 2*A(3)*C30...
- 2*A(4)*C40;

for j=1:N
CC = coeffs(R(j),A(j));
alpha(j+1)=CC(3);
beta(j+1)=-CC(2)/2;
R(j+1)=simplify(R(j)-alpha(j+1)*A(j)^2+2*beta(j+1)*A(j)-beta(j+1)^2/alpha(j+1));
end

disp(R(N+1));

syms x y z
M1=10;
M2=30;
M=M1+M2;
x=sym(zeros(1,M));
for k=1:M1
x(k)=-3/2+(3/(M1-1))*(k-1);
end
x(M1+1:M1+M2)=3*rand(1,M2)-3/2;
for k=1:M
y(k)=f(x(k));
end
```

```
% noise
L=0.001;
z=2*L*rand(1,M)-L;
y = y+z;

C00=0;
C11=0;
C12=0;
C13=0;
C14=0;
C22=0;
C23=0;
C24=0;
C33=0;
C34=0;
C44=0;
C10=0;
C20=0;
C30=0;
C40=0;

for j=1:M
C00=C00+y(j)^2;
C11=C11+x(j)^2*exp(2*phi*x(j));
C12=C12+x(j)*exp(2*phi*x(j));
C13=C13+x(j)^2;
C14=C14+x(j);
C22=C22+exp(2*phi*x(j));
C23=C23+x(j);
C24=C24+1;
C33=C33+x(j)^2*exp(-2*phi*x(j));
C34=C34+x(j)*exp(-2*phi*x(j));
C44=C44+exp(-2*phi*x(j));
C10=C10+x(j)*exp(phi*x(j))*y(j);
C20=C20+exp(phi*x(j))*y(j);
C30=C30+x(j)*exp(-phi*x(j))*y(j);
C40=C40+exp(-phi*x(j))*y(j);
end

syms Q(phi)
Q(phi) = subs(C00+R(N+1));

figure
fplot(Q(phi),[1 7]) Fig. 11
```

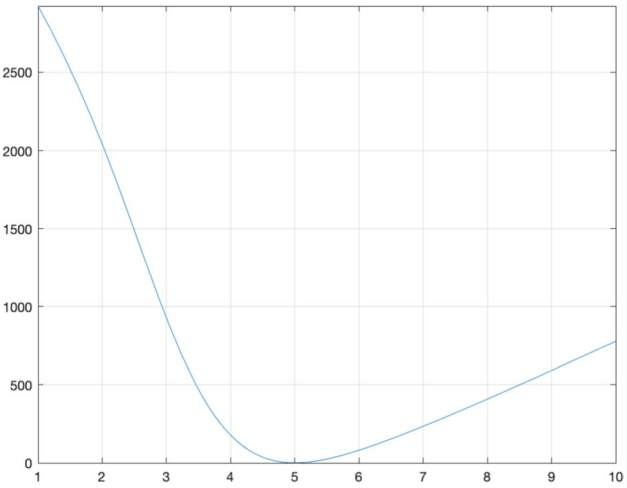


Fig. 11 Code figure for $\min_{A_j,\phi} R(A_1,\ldots,A_4;\phi) = \sum_{k=0}^{M-1} y_k^2 + \min_\phi R_4(\phi)$

Acknowledgements The authors would like to thank Rosanna Campagna of the University of Campania in Italy, for making the code available that computes penalized exponential splines.

Author Contributions C. C., A. C., and W.-s. L. contributed equally to this work and reviewed the manuscript.

Funding Costanza Conti is a member of the Italian INdAM-GNCS research group, which partially supported this work with Project CUP E53C23001670001.
Annie Cuyt and Wen-shin Lee received funding from the European Union Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement 101008231 (EXPOWER).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Prony, G.R.: Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. J. Ec. Poly. **1**(22), 24–76 (1795)
2. Hildebrand, F.B.: Introduction to Numerical Analysis. Mc Graw Hill, New York (1956)
3. Multiscale matrix pencils for separable reconstruction problems: Cuyt, A., Lee, W.-s. Numerical Algorithms **95**, 31–72 (2024). https://doi.org/10.1007/s11075-023-01564-3. (Published online June 22, 2023.)

4. Conti, C., Romani, L.: Algebraic conditions on non-stationary subdivision symbols for exponential polynomial reproduction. J. Comput. Appl. Math. **236**, 543–556 (2011)
5. Albrecht, G., Mainar, E., Peña, J.M., Rubio, B.: A new class of trigonometric B-spline curves. Symmetry **15**(8), (2023). https://doi.org/10.3390/sym15081551
6. Lü, Y., Wang, G., Yang, X.: Uniform hyperbolic polynomial B-spline curves. Computer Aided Geometric Design **19**(6), 379–393 (2002). https://doi.org/10.1016/S0167-8396(02)00092-4
7. Costantini, P., Kaklis, P.D., Manni, C.: Polynomial cubic splines with tension properties. Computer Aided Geometric Design **27**(8), 592–610 (2010). https://doi.org/10.1016/j.cagd.2010.06.007. Advances in Applied Geometry
8. McCartin, B.J.: Theory of exponential splines. J. Approx. Theory **66**, 1–23 (1991)
9. Unser, M., Blu, T.: Cardinal Exponential Splines: Part I—Theory and Filtering Algorithms. IEEE Trans. Signal Process. **53**(4), 1425–1428 (2005). https://doi.org/10.1109/TSP.2005.843700
10. Schumaker, L.L.: On Tchebycheffian spline functions. J. Approx. Theory **18**(3), 278–303 (1976). https://doi.org/10.1016/0021-9045(76)90021-6
11. Lyche, T.: A recurrence relation for Chebyshevian B-splines. Constr. Approx. **1**(1), 155–173 (1985). https://doi.org/10.1007/BF01890028
12. Mazure, M.: Piecewise Chebyshevian splines: interpolation versus design. Numerical Algorithms **77**, 1213–1247 (2018). https://doi.org/10.1007/s11075-017-0360-7
13. Uhlmann, V., Delgado-Gonzalo, R., Conti, C., Romani, L., Unser, M.: Exponential Hermite splines for the analysis of biomedical images. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1631–1634 (2014). https://doi.org/10.1109/ICASSP.2014.6853874
14. Andersen, L.: Discount curve construction with tension splines. Rev. Deriv. Res. **10**, 227–267 (2007). https://doi.org/10.1007/s11147-008-9021-2
15. Hua, Y., Sarkar, T.K.: Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. IEEE Trans. Acoust., Speech, Signal Process. **38**, 814–824 (1990). https://doi.org/10.1109/29.56027
16. Golub, G., Pereyra, V.: Separable nonlinear least squares: the variable projection method and its applications. Inverse Prob. **19**(2), 1–26 (2003). https://doi.org/10.1088/0266-5611/19/2/201
17. Conti, C., López-Ureña, S., Romani, L.: Annihilation operators for exponential spaces in subdivision. Appl. Math. Comput. **418**, 126796 (2022). https://doi.org/10.1016/j.amc.2021.126796
18. Campagna, R., Conti, C., Cuomo, S.: A linear algebra approach to HP-splines frequency parameter selection. Appl. Math. Comput. **458**, 128241 (2023). https://doi.org/10.1016/j.amc.2023.128241
19. Sidi, A.: Interpolation at equidistant points by a sum of exponential functions. J. Approx. Theory **34**, 194–210 (1982)
20. Braess, D.: Nonlinear Approximation Theory. Springer series in computational mathematics. Springer, Berlin, New York (1986)
21. Greene, D.H., Knuth, D.E.: Mathematics For the Analysis of Algorithms, p. 107. Birkhäuser Verlag, Boston (1981)
22. Kaltofen, E., Lee, W.-s.: Early termination in sparse interpolation algorithms. J. Symbolic Comput. **36**(3-4), 365–400 (2003). https://doi.org/10.1016/S0747-7171(03)00088-9 . International Symposiu m on Symbolic and Algebraic Computation (ISSAC 2002) (Lille)
23. Zhang, Y., Cuyt, A., Lee, W.-s., Lo Bianco, G., Wu, G., Chen, Y., Li, D.D.-U.: Towards unsupervised fluorescence lifetime imaging using low dimensional variable projection. Opt. Express **24**(23), 26777–26791 (2016)
24. Bosner, T., Rogina, M.: Variable degree polynomial splines are Chebyshev splines. Adv. Comput. Math. **38**, 383–400 (2011)
25. Lanczos, C.: Applied Analysis. Prentice Hall Inc, Englewood Cliffs, NJ (1956)
26. Wang, Y.: Smoothing Splines : Methods and Applications. Monographs on statistics and applied probability (Series) ; 121. CRC Press, Boca Raton, FL (2011)
27. Campagna, R., Conti, C.: Penalized hyperbolic-polynomial splines. Appl. Math. Lett. **118**, 107159 (2021). https://doi.org/10.1016/j.aml.2021.107159
28. Oguejiofor, G.C.: Modeling of linear and exponential growth and decay equations and testing them on pre- and post-war-coal production in Nigeria: An operations research approach. Energy Sources Part B **5**(2), 116–125 (2010). https://doi.org/10.1080/15567240802053459