Generalized Prony Method

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Dissertation zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

Doctor rerum naturalium

der Georg-August-Universität Göttingen

vorgelegt von Thomas Peter aus Rostock

Göttingen 2013

Betreuungsausschuss

- Prof. Dr. Gerlind Plonka-Hoch, Institut für numerische und angewandte Mathematik, Georg-Agust-Universität Göttingen
- Prof. Dr. Russell Luke, Institut für numerische und angewandte Mathematik, Georg-Agust-Universität Göttingen

Mitglieder der Prüfungskommission

- Referentin:
  Prof. Dr. Gerlind Plonka-Hoch, Institut für numerische und angewandte Mathematik, Georg-Agust-Universität Göttingen
- Korreferent:
  Prof. Dr. Daniel Potts, Fakultät für Mathematik, Technische Universität Chemnitz

Mitglieder der Prüfungskommission

- Prof. Dr. Russell Luke, Institut für numerische und angewandte Mathematik, Georg-Agust-Universität Göttingen
- Prof. Dr. Robert Schaback, Institut für numerische und angewandte Mathematik, Georg-Agust-Universität Göttingen
- Prof. Dr. Stephan Huckemann, Institut für Mathematische Stochastik, Georg-Agust-Universität Göttingen
- Prof. Dr. Thomas Schick, Mathematisches Institut, Georg-Agust-Universität Göttingen

Tag der mündlichen Prüfung: 19.09.2013

"Surf on the zenith of vacuousness!" H.G., 1998

Contents

Acknowledgments

This dissertation was written at the Georg-August-University in Göttingen. I am grateful to the Research Training Group 1023: *Identification in Mathematical Models* for awarding me the Ph.D. fellowship that enabled me to write this thesis in 2010 - 2013, in a very inspiring environment. I would like first and foremost to thank my adviser, Gerlind Plonka-Hoch, for her support and faith in this project and for always making time for enlightening discussions. I want to express my gratitude to Daniel Potts for the interesting discussions through the years and for his effort as a referee for this thesis. My thanks also go to Manfred Tasche for introducing me to the field of signal processing and for our continued exchange of ideas around topics about the Prony Method. Parts of the thesis were presented at various conferences, also in the course of the Research Training Group 1023 and I want to thank my colleagues and all the audiences at those presentations for the offered comments and encouragements.

I also want to thank my parents and friends for their everlasting support throughout my studies and for tolerating my neglect of them towards the end of my thesis. I am particularly thankful for having Joyce at my side, who is always there for me.

0 Introduction

The intention of this thesis is the unification and essential generalization of nonlinear reconstruction methods based on the Prony Method. For that purpose we use the new perception that the extensively used classical Prony Method for parameter identification in exponential sums can also be considered as a reconstruction technique for M-term classification of eigenfunctions of special linear operators. This new insight in the method enables us to derive new generalized reconstruction methods for structured functions that can be depicted for example as trigonometric functions, orthogonal polynomials or finite dimensional vectors. Thus, it establishes a broader field of applications in, e.g. signal analysis or approximation theory.

1 A Generalization of Prony's Method

In recent years, the Prony Method has been successfully applied to different inverse problems, as e.g. for approximation of Green functions in quantum chemistry [55] or fluid dynamics [6], for localization of particles in inverse scattering [27], for parameter estimation of dispersion curves of guided waves [50], and for analysis of ultrasonic signals [7]. The renaissance of Prony's method originates from some modifications of the original algorithm that considerably stabilize the original approach, as e.g. the ESPRIT method, the Matrix Pencil Method or the Approximate Prony Method (APM) [28, 40, 48]. We will see that the Prony Method works under the precondition that the signal at hand is M-sparse in a certain function space. But the techniques mentioned above can also be applied if the sparsity number M is not known beforehand, provided that a sufficiently large number of measurements is given. And the applications in practice show that they work well even in case of noisy measurements. Error estimates for the performance of Prony-like methods with noisy measurements are derived in [2, 19, 40].

In this thesis, we want to present a new very general approach for the reconstruction of sparse expansions of eigenfunctions of suitable linear operators. This new insight provides us with a tool to unify all Prony-like methods on the one hand and to essentially generalize the Prony approach on the other hand. Thus it will establish a much broader field of applications of the method. In particular, we will show that all well-known Prony-like reconstruction methods for exponentials and polynomials known so far, can be seen as special cases of this approach. Moreover, the new insight into Prony-like methods enables us to derive new reconstruction algorithms for orthogonal polynomial expansions including Jacobi, Laguerre, and Hermite polynomials. The approach also applies to finite dimensional vector spaces, and we derive a deterministic reconstruction method for M-sparse vectors from only $2M$ measurements.

This dissertation is organized as follows. In the first chapter the original Prony Method is introduced, followed by a generalization in terms of eigenfunctions of linear operators and generalized eigenfunctions. In chapter two we want to disengage the method from the restriction of a priori needed information of the sparsity M that the signal is supposed to inherit. The third chapter treats numerical realizations of our generalized Prony Method. This new approach and its range will be enlighted in chapter four with the help of examples. Chapter five is dedicated to further extensions of the generalized Prony Method and chapter six rises attention to applications. The last chapter is a collection of open problems in this topic.

1.1 Original Prony Method

In 1795, Gaspard Riche de Prony studied expansion characteristics of gases and stated that they can be well described via low order exponential functions [45].

This means he wanted to solve the problem

$$\min_{c_j, T_j} \|u(k) - f(k)\|_2, \quad k = 0, \dots, 2M, \tag{1.1}$$

with

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j x}, \tag{1.2}$$

for a small number $2M + 1$ of given equidistant measurements $u(k)$, $k = 0, \dots, 2M$. Instead of solving the approximation problem (1.1), he considered the measurements $u(k)$ to be described exactly via exponential functions and introduced a method to solve the exact problem of finding c_j, T_j, $j = 1, \dots, M$ such that $u(k) = f(k)$, $k = 0, \dots, 2M$. His method relies on techniques from the field of finite difference equations. Therefore, we start with a small excursion into that field, which is mainly based on [21], chapter 5.

An $(M + 1)$-valued function F of the form

$$F(f(x), \Delta f(x), \dots, \Delta^M f(x)) = 0 \tag{1.3}$$

is called homogeneous difference equation. Here, $\Delta^M f(x)$ is recursively defined by

$$\begin{aligned} \Delta^M f(x) &:= \Delta^{M-1} f(x+1) - \Delta^{M-1} f(x), \quad M \geq 1, \\ \Delta^0 f(x) &:= f(x). \end{aligned}$$

If F in (1.3) is only linearly dependent on $\Delta^k f(x)$, $k = 0, \dots, M$, we can rewrite (1.3) as

$$\sum_{k=0}^{M} a_k \Delta^k f(x) = 0, \tag{1.4}$$

If, moreover, all a_k, $k = 0, \dots, M$ are independent of x we call (1.4) a *linear homogeneous difference equation with constant coefficients*. One can show that

$$\Delta^k f(x) = \sum_{\ell=0}^{k} (-1)^{k-\ell} \binom{k}{\ell} f(x + \ell),$$

thus we can rewrite (1.4) in terms of translations,

$$\sum_{k=0}^{M} p_k f(x + k) = 0, \tag{1.5}$$

with certain coefficients p_k, $k = 0, \dots, M$. To solve this linear homogeneous difference equation, we use the ansatz $f(x) = z^x$, for some $z \neq 0$ and insert it

in (1.5),

$$\sum_{k=0}^{M} p_k z^{x+k} = 0,$$

$$\sum_{k=0}^{M} p_k z^k = 0. \tag{1.6}$$

The algebraic polynomial of order M in (1.6) is called the *characteristic polynomial* associated with the linear difference equation (1.5). If all roots z_j of the polynomial (1.6) are simple, the general solution of (1.5) will be a linear combination of the functions z_j^x, $j = 1, \ldots, M$. We can represent the possibly complex roots of the polynomial $P(z) := \sum_{k=0}^{M} p_k z^k$ as $z_j = \mathrm{e}^{T_j}$, with the restriction $\mathrm{Im}(T_j) \in [-\pi, \pi)$.

If the parameters T_j, $j = 1, \ldots, M$ of the function f in (1.2) are known, we can compute the coefficients c_j in a subsequent step. In order to determine the coefficients c_j uniquely we need M initial values, e.g. the given values $f(1), \ldots, f(M)$. This leads to the problem of solving the linear Vandermonde system

$$\mathbf{Vc} = \mathbf{f}$$

$$\begin{pmatrix} z_1^1 & z_2^1 & \cdots & z_M^1 \\ \vdots & \vdots & & \vdots \\ z_1^M & z_2^M & \cdots & z_M^M \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \end{pmatrix} = \begin{pmatrix} f(1) \\ \vdots \\ f(M) \end{pmatrix}.$$

The only problem left is to determine the a priori unknown characteristic function $P(z)$, or equivalently, the coefficients p_k in (1.5) out of the given data. Equation (1.5) is satisfied for any x. So, by inserting $x = 1, \ldots, M$ in (1.5) we can provide ourselves with enough equations to determine the coefficients p_j that define the characteristic function. This leads to the problem of solving a homogeneous linear system, where the system matrix is a Hankel matrix,

$$\mathbf{Hp} = \mathbf{0}, \tag{1.7}$$

$$\begin{pmatrix} f(0) & f(1) & \cdots & f(M) \\ f(1) & f(2) & \cdots & f(M+1) \\ \vdots & \vdots & & \vdots \\ f(M) & f(M+1) & \cdots & f(2M) \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_M \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Remark 1.1 *Since multiplying a polynomial $P(z)$ by a constant does not change its roots, we can define $p_M := 1$ and reduce the number of needed input values by one if we rewrite (1.5) to*

$$\sum_{k=0}^{M-1} p_k f(x+k) = -f(x+M)$$

and solve the inhomogeneous system

$$\begin{pmatrix} f(0) & f(1) & \cdots & f(M-1) \\ f(1) & f(2) & \cdots & f(M) \\ \vdots & \vdots & & \vdots \\ f(M-1) & f(M) & \cdots & f(2M-2) \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{M-1} \end{pmatrix} = \begin{pmatrix} f(M) \\ f(M+1) \\ \vdots \\ f(2M-1) \end{pmatrix}.$$

A proof that that the system matrix in the equation above has indeed rank M and thus leads to a unique solution, is done more generally in Theorem 2.1.

Let us summarize this method in an algorithm.

1.1.1 Algorithm

Reconstruction of sparse expansions of exponentials

Let $f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j}$, with $c_j \in \mathbb{C}\backslash\{0\}$, $T_j \in \mathbb{C}$, $\mathrm{Im}(T_j) \in [-\pi, \pi)$.
Input: M, $f(0), \ldots, f(2M-1)$.

1. Form the Hankel-matrix $\mathbf{H} := (f(k+\ell))_{k,\ell=0}^{M-1}$ and solve the system $\mathbf{Hp} = -\mathbf{f}$ with $\mathbf{p} = (p_k)_{k=0}^{M-1}$ and $\mathbf{f} = (f(M+k))_{k=0}^{M-1}$.
2. Define $p_M = 1$ and find all roots e^{T_j}, $j = 1, \ldots, M$ of the polynomial $P(z) = \sum_{k=0}^{M} p_k z^k$.
3. Determine the unknowns c_j, $j = 1, \ldots, M$ as the solution of the Vandermonde-system $\mathbf{Vc} = \mathbf{f}_1$, with $\mathbf{V} := (\mathrm{e}^{T_j k})_{k=0,j=1}^{M-1,M}$, $\mathbf{c} := (c_j)_{j=1}^{M}$, $\mathbf{f}_1 := (f(k))_{k=0}^{M-1}$.

Output: $c_j, T_j = \ln(\mathrm{e}^{T_j})$, $j = 1, \ldots, M$.

The algorithm, as it was introduced above, is just a plain version in order to explain the idea. For real applications one has to consider erroneous data, unknown sparsity and numerical issues that follow from them. This will be the topic in chapters 2 and 3.

In the literature there can be found various ideas to solve this non-linear inverse problem, where [28, 39, 45, 48, 49, 54] is just an excerpt of the list of methods, but the oldest known goes back to Baron de Prony.

1.2 The Generalized Prony Method

Now that we understand how the Prony Method works for sparse linear combinations of exponential functions, we might ask ourselves how this method can be generalized for reconstructing sparse expansions of other function systems.

Recently, we considered in [35] the function reconstruction problem for sparse Legendre expansions of order N of the form

$$f(x) = \sum_{j=1}^{M} c_j P_{n_j}(x)$$

with $0 \leq n_1 < n_2 \ldots < n_M = N$, where $M \ll N$, aiming at a generalization of Prony's method for this case. We derived a reconstruction algorithm involving the function and derivative values $f^{(\ell)}(1)$, $\ell = 0, \ldots, 2M-1$. The reconstruction in [35] is based on special properties of Legendre polynomials, but it does not provide a direct approach for further generalization of the method to other sparse orthogonal polynomial expansions or to other function systems apart from exponentials and monomials.

In [16], the idea of efficient sparse polynomial interpolation has been transferred to the more general case of M-term sums of characters of abelian monoids. This approach has also been used in [26] for the reconstruction of functions being linear combinations of eigenfunctions of linear operators on suitable algebras on integral domains. This last paper can be seen as one starting point for our considerations in our work [34] and therefore also in this thesis.

Let V be a normed vector space over $\mathbb{C}$, and let $\mathcal{A} : V \to V$ be a linear operator.

Assume that $\mathcal{A}$ possesses eigenvalues, and let $\Lambda := \{\lambda_j : j \in I\}$ be a (sub)set of pairwise distinct eigenvalues of $\mathcal{A}$, where I is a suitable index set. We consider the eigenspaces $\mathcal{V}_j = \{v : \mathcal{A}v = \lambda_j v\}$ to the eigenvalues λ_j, and for each $j \in I$, we predetermine a one-dimensional subspace $\tilde{\mathcal{V}}_j$ of $\mathcal{V}_j$ that is spanned by the normalized eigenfunction v_j. In particular, we assume that there is a unique correspondence between λ_j and v_j, $j \in I$.

An expansion f of eigenfunctions of the operator $\mathcal{A}$ is called M-sparse if its representation consists of only M non-vanishing terms, i.e. if

$$f = \sum_{j \in J} c_j v_j, \quad \text{with } J \subset I \text{ and } |J| = M. \tag{1.8}$$

Due to the linearity of the operator $\mathcal{A}$, the k-fold application of $\mathcal{A}$ to f yields

$$\mathcal{A}^k f = \sum_{j \in J} c_j \mathcal{A}^k v_j = \sum_{j \in J} c_j \lambda_j^k v_j. \tag{1.9}$$

Further, let $F : V \to \mathbb{C}$ be a linear functional with the property $Fv_j \neq 0$ for all $j \in I$. We show that the expansion f in (1.8) can be reconstructed by using only the $2M$ values $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2M-1$.

Theorem 1.2 *With the above assumptions, the expansion f in* (1.8) *of eigenfunctions $v_j \in \tilde{\mathcal{V}}_j$, $j \in J \subset I$, of the linear operator $\mathcal{A}$ can be uniquely reconstructed from the values $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2M-1$, i.e., the "active" eigenfunctions v_j as well as the coefficients $c_j \in \mathbb{C}$, $j \in J$, in* (1.8) *can be determined uniquely.*

Proof. We give a constructive proof.

1. We define the so-called Prony polynomial

$$P(z) := \prod_{j \in J} (z - \lambda_j),$$

where the roots λ_j, $j \in J$, are the eigenvalues corresponding to the (unknown) active eigenfunctions v_j in the representation of f. Further, let $P(z) = \sum_{k=0}^{M} p_k z^k$, with $p_M = 1$, be the monomial representation of the Prony polynomial. Combining the unknown coefficients p_k with the given values $F(\mathcal{A}^k f)$, $k = 0, \dots, 2M-1$, and using (1.9) we observe the following relation for $m = 0, 1, \dots,$

$$\begin{aligned}\sum_{k=0}^{M} p_k F(\mathcal{A}^{k+m} f) &= \sum_{k=0}^{M} p_k F\left(\sum_{j\in J} c_j \lambda_j^{k+m} v_j\right) = \sum_{j\in J} c_j \lambda_j^m \left(\sum_{k=0}^{M} p_k \lambda_j^k\right) F v_j \\ &= \sum_{j\in J} c_j \lambda_j^m \underbrace{P(\lambda_j)}_{=0} F v_j = 0.\end{aligned}$$

Together with $p_M = 1$, the coefficients p_k, $k = 0, \dots, M-1$, of the Prony polynomial can now be determined via the linear system

$$\sum_{k=0}^{M-1} p_k F(\mathcal{A}^{k+m} f) = -F(\mathcal{A}^{M+m} f), \quad m = 0, \dots, M-1. \tag{1.10}$$

Indeed, the coefficient matrix $\mathbf{H} := (F(\mathcal{A}^{k+m} f))_{k,m=0}^{M-1,M-1}$ is an invertible Hankel matrix, since (1.9) yields

$$\mathbf{H} = \mathbf{V}_\lambda \cdot \operatorname{diag}(\mathbf{c}) \cdot \operatorname{diag}(F\mathbf{v}) \cdot \mathbf{V}_\lambda^{\mathrm{T}}$$

with the Vandermonde matrix

$$\mathbf{V}_\lambda := \left(\lambda_j^k\right)_{k=0,j\in J}^{M-1}$$

and with the diagonal matrices $\operatorname{diag}(\mathbf{c}) = \operatorname{diag}(c_j)_{j\in J}$, $\operatorname{diag}(F\mathbf{v}) = \operatorname{diag}(Fv_j)_{j\in J}$, where the indices $j \in J$ are assumed to be given in a fixed order. By assumption, $\mathbf{V}_\lambda$ as well as the diagonal matrices $\operatorname{diag}(\mathbf{c})$ and $\operatorname{diag}(F\mathbf{v})$ have full rank yielding the invertibility of $\mathbf{H}$.

2. Having determined the Prony polynomial

$$P(z) = \sum_{k=0}^{M} p_k z^k = \prod_{j\in J} (z - \lambda_j),$$

we can evaluate the eigenvalues λ_j, $j \in J$, that are the zeros of $P(z)$. Since the eigenspaces $\tilde{\mathcal{V}}_j$ are assumed to be one-dimensional we can uniquely determine the corresponding eigenfunctions v_j, $j \in J$.

3. In the last step we compute the coefficients c_j, $j \in J$, of the expansion (1.8) by solving the overdetermined linear system

$$F(\mathcal{A}^k f) = \sum_{j\in J} c_j \lambda_j^k v_j, \quad k = 0, \dots, 2M-1,$$

using the eigenvalues λ_j and eigenfunctions v_j found in the previous step. ■

This general approach to the Prony Method enables us to derive reconstruction algorithms for a variety of systems of eigenfunctions. We summarize the algorithm as follows.

1.2.1 Algorithm

Reconstruction of the sparse expansion (1.8)

Input: M, $F(\mathcal{A}^k f)$, $k = 0, \dots, 2M-1$.

1. Solve the linear system

$$\sum_{k=0}^{M-1} p_k F(\mathcal{A}^{k+m} f) = -F(\mathcal{A}^{M+m} f), \qquad m = 0, \dots, M-1. \tag{1.11}$$

2. Form the Prony polynomial $P(z) = \sum_{k=0}^{M} p_k z^k$ using the obtained values p_k, $k = 0, \dots, M-1$ from step 1 and $p_M = 1$. Compute the zeros λ_j, $j \in J$, of $P(z)$ and determine the corresponding (normalized) eigenfunctions v_j, $j \in J$.

3. Compute the coefficients c_j by solving the overdetermined system

$$F(\mathcal{A}^k f) = \sum_{j \in J} c_j \lambda_j^k v_j \qquad k = 0, \dots, 2M-1.$$

Output: c_j, v_j, $j \in J$, determining f in (1.8).

A MATLAB code for this algorithm, with $h := (F(\mathcal{A}^k f))_{k=0}^{2M-1}$ and $h_M := h(k)_{k=M}^{2M-1}$ can look as follows. Note that the index in MATLAB for matrices and vectors starts a 1 instead of 0, as we have used it until now.

MATLAB implementation of algorithm 1.2.1

```
% input signal column vector %h%, with 2M values.
% input sparsity number $M$.
H = hankel(h(1:M),h(M:(end-1)));
h_M = h(M+1:end);
Lambda = roots([1; flipud(H\(-h_M))]);
V = construct_V(Lambda);
c = V\h;
```

The found eigenvalues $\lambda_1, \dots, \lambda_M$ are stored in the vector $\mathbf{\Lambda}$ and the corresponding coefficients are stored in $\mathbf{c}$. Note that the function that constructs the Vandermonde-type-matrix $\mathbf{V}$ in the MATLAB implementation shown above is not only dependent on the found eigenvalues $\lambda_1, \dots, \lambda_M$, but also on the operator $\mathcal{A}$ and the functional F, for they define the sampling values. That is why we cannot show a specific implementation of that function. But at the end of this chapter we will present one version of the construction of $\mathbf{V}$ for a special example of the operator $\mathcal{A}$.

In order to enhance the comprehension of the generalization to sparse expansions of eigenfunctions of linear operators, we will now look at an example.

A demonstration of the range of applications of the generalized Prony Method will be undertaken in chapter 4.1, where we will investigate a lot more specific linear operators $\mathcal{A}$ and also the impact of special functionals F to the algorithm 1.2.1.

Let us consider the vector space $C(\mathbb{R})$ of continuous functions, and let $S_a : C(\mathbb{R}) \to C(\mathbb{R})$ with

$$S_a f(x) := f(x+a), \quad a \in \mathbb{R}\backslash\{0\} \tag{1.12}$$

be the shift operator on $C(\mathbb{R})$. We observe that $\{\mathrm{e}^{Ta} : T \in \mathbb{C},\ \mathrm{Im}\, T \in [-\frac{\pi}{a}, \frac{\pi}{a})\}$ is a set of pairwise distinct eigenvalues of S_a, and by

$$S_a \mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})} = \mathrm{e}^{(x+a)(T+\frac{2\pi \mathrm{i} k}{a})} = \mathrm{e}^{Ta}\mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})}, \qquad x \in \mathbb{R},\ k \in \mathbb{Z},$$

we find for each eigenvalue $\lambda_T := \mathrm{e}^{Ta}$, $T \in I := \{T \in \mathbb{C},\ \mathrm{Im}\, T \in [-\frac{\pi}{a}, \frac{\pi}{a})\}$, the eigenspace $\mathcal{V}_T := \mathrm{span}\,\{\mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})} : k \in \mathbb{Z}\}$. In order to obtain a unique correspondence between λ_T and its eigenfunction, we only consider the subeigenspaces $\tilde{\mathcal{V}}_T = \mathrm{span}\,\{\mathrm{e}^{Tx}\}$. Further, let the functional $F : C(\mathbb{R}) \to \mathbb{C}$ be given by

$$F(f) := f(x_0), \quad \forall f \in C(\mathbb{R}), \tag{1.13}$$

with an arbitrarily fixed $x_0 \in \mathbb{R}$. Hence $F(\mathrm{e}^{T\cdot}) = \mathrm{e}^{Tx_0} \neq 0$ for all $T \in I$. Applying Theorem 1.2 yields that the sparse sum of exponentials

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j x} \tag{1.14}$$

with pairwise different $T_j \in \mathbb{C}$ and $\mathrm{Im}(T_j) \in [-\frac{\pi}{a}, \frac{\pi}{a})$ can be uniquely reconstructed from the values

$$F(S_a^k f) = F(f(\cdot + ka)) = f(x_0 + ka), \quad k = 0, \ldots, 2M-1,$$

e.g. from $2M$ equidistant sampling points with sampling distance a, starting at point x_0.

Let us look at a realization of the function that constructs $\mathbf{V}$ in the MATLAB implementation of algorithm 1.2.1. Let $\mathcal{A} := \mathcal{S}_2$ and let F be the point evaluation functional $F(f) := f(x_0)$, with $x_0 = 5$. The construction of $\mathbf{V}$ can then look as follows.

MATLAB implementation of `construct_V`

```
function V = construct_V(Lambda)
M = length(Lambda);
x = (0:1:2*M-1)*2+5;
V = (ones(2*M,M)*diag(Lambda)).^(diag(x)*ones(2*M,M));
end
```

Remark 1.3 *If we choose the linear operator $\mathcal{A}$ in (1.9) to be the translation operator S_a (4.34) with $a = 1$ and if we further choose the point evaluation functional F in theorem 1.2 to be $F(f) := F(0)$, we see that algorithm 1.2.1 reduces to algorithm 1.1.1. In other words, the original Prony Method is a special case of the generalized Prony Method applied to the translation operator S_a. And even if we only consider the translation operator S_1, we get a vast generalization of the original Prony Method, due to the freedom of choice in the functional F.*

1.3 Generalized Eigenfunctions

Let us also consider the case of generalized eigenfunctions. Let $r \geq 1$ be a fixed integer. Analogously as for linear operators in finite-dimensional vector spaces, we say that $\tilde{v}_\ell$, $\ell = 1, \ldots, r$, are generalized eigenfunctions of multiplicity ℓ of a linear operator $\mathcal{A} : V \to V$ to the eigenvalue λ, if

$$(\mathcal{A} - \lambda I)^\ell \tilde{v}_\ell = 0, \qquad \ell = 1, \ldots, r,$$

and

$$\mathcal{A}\tilde{v}_\ell = \lambda \tilde{v}_\ell + \sum_{s=1}^{\ell-1} \alpha_{\ell,s} \tilde{v}_s, \qquad \ell = 1, \ldots, r, \tag{1.15}$$

with some constants $\alpha_{\ell,s} \in \mathbb{C}$. Again, let $\Lambda = \{\lambda_j,\, j \in I\}$ be a (sub)set of pairwise distinct eigenvalues of $\mathcal{A}$, and for each $j \in I$, let $\{\tilde{v}_{j,\ell} : \ell = 1, \ldots, r\}$ be a predetermined set of linearly independent generalized eigenfunctions to the eigenvalue λ_j. Further, let $F : V \to \mathbb{C}$ be a functional with $F(\tilde{v}_{j,\ell}) \neq 0$ for $j \in I, \ell = 1, \ldots, r$.

Theorem 1.4 *With the above assumptions, the expansion*

$$f = \sum_{j \in J} \sum_{\ell=1}^{r} c_{j,\ell} \tilde{v}_{j,\ell}, \qquad J \subset I,\, |J| = M,\, r \geq 1,$$

of generalized eigenfunctions of the linear operator $\mathcal{A}$ to the eigenvalues λ_j, $j \in J \subset I$, can be uniquely reconstructed from the values $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2rM - 1$, supposed that the matrix $(F(\mathcal{A}^{k+m} f))_{k,m=0}^{rM-1,rM-1}$ is invertible.

Proof. Using an induction argument, equation (1.15) implies that

$$\mathcal{A}^k \tilde{v}_{j,\ell} = \sum_{s=0}^{\ell-1} \binom{k}{k-s} \lambda_j^{k-s} \sum_{\mu_1=1}^{\ell-1} \sum_{\mu_2=1}^{\mu_1-1} \cdots \sum_{\mu_s=1}^{\mu_{s-1}-1} \left(\prod_{\nu=1}^{s} \alpha_{j,\ell-\mu_\nu} \right) \tilde{v}_{j,\mu_s},$$

where we set $\binom{k}{k-s} := 0$ if $s > k$. We now consider a generalized Prony polynomial of the form

$$P(z) := \prod_{j \in J} (z - \lambda_j)^r = \sum_{k=0}^{Mr} p_k z^k, \tag{1.16}$$

where again λ_j denote the unknown eigenvalues of $\mathcal{A}$ that determine the active sets of (generalized) eigenfunctions. Then we obtain the following relation for $m = 0, \dots, Mr-1$,

$$\begin{aligned}
&\sum_{k=0}^{Mr} p_k F(\mathcal{A}^{k+m} f) \\
&= \sum_{k=0}^{Mr} p_k F\left(\sum_{j\in J} \sum_{\ell=1}^{r} c_{j,\ell} \mathcal{A}^{k+m} \tilde{v}_{j,\ell} \right) \\
&= \sum_{k=0}^{Mr} p_k F\left(\sum_{j\in J} \sum_{\ell=1}^{r} c_{j,\ell} \sum_{s=0}^{\ell-1} \binom{k+m}{k+m-s} \lambda_j^{k+m-s} \sum_{\mu_1=1}^{\ell-1} \cdots \right. \\
&\qquad \left. \cdots \sum_{\mu_s=1}^{\mu_{s-1}-1} \left(\prod_{\nu=1}^{s} \alpha_{j,\ell-\mu_\nu} \right) \tilde{v}_{j,\mu_s} \right) \\
&= \sum_{j\in J} \sum_{\ell=1}^{r} c_{j,\ell} \sum_{s=0}^{\ell-1} \left(\sum_{k=0}^{Mr} p_k \binom{k+m}{k+m-s} \lambda_j^{k+m-s} \cdots \right. \\
&\qquad \left. \cdots \sum_{\mu_s=1}^{\mu_{s-1}-1} \left(\prod_{\nu=1}^{s} \alpha_{j,\ell-\mu_\nu} \right) \right) F(\tilde{v}_{j,\ell-s}) \\
&= \sum_{j\in J} \sum_{\ell=1}^{r} c_{j,\ell} \sum_{s=0}^{\ell-1} \lambda_j^{m-s} \left(\sum_{\mu_1=1}^{\ell-1} \cdots \sum_{\mu_s=1}^{\mu_{s-1}-1} \prod_{\nu=1}^{s} \alpha_{j,\ell-\nu} \right) \cdot \\
&\qquad \left(\sum_{k=0}^{Mr} p_k \binom{k+m}{k+m-s} \lambda_j^{k} \right) F(\tilde{v}_{j,\ell-s}).
\end{aligned}$$

Now, the term

$$\sum_{k=0}^{Mr} p_k \binom{k+m}{k+m-s} \lambda_j^k$$

can be written as a linear combination of the Prony polynomial $P(z)$ and its first $r-1$ derivatives $P^{(s)}(z) = \sum_{k=s}^{Mr} p_k \frac{k!}{(k-s)!} z^{k-s}$ evaluated at λ_j. For this purpose, we only have to show that there exist coefficients $\beta_{j,s,t}$ such that

$$\sum_{t=0}^{r-1} \beta_{j,s,t} \frac{k!}{(k-t)!} = \binom{k+m}{k+m-s}$$

holds for each $k = 0, \dots, Mr$, where the coefficients $\beta_{j,s,t}$ are independent of k. This is obviously possible, since each polynomial in k of degree up to $r-1$ can be written in the form $\sum_{t=0}^{r-1} \beta_{j,s,t} \frac{k!}{(k-t)!}$. Hence, by $P^{(s)}(\lambda_j) = 0$ for $j \in J, s = 0, \dots, r-1$, it follows that

$$\sum_{k=0}^{Mr} p_k F(\mathcal{A}^{k+m} f) = 0$$

for $m = 0, \dots Mr - 1$. In this way, we obtain again a linear Hankel system of the form

$$\sum_{k=0}^{rM-1} p_k F(\mathcal{A}^{k+m} f) = -F(\mathcal{A}^{rM+m} f), \qquad m = 0, \dots, rM-1$$

in order to determine the coefficients p_k of the Prony polynomial. Having determined the zeros λ_j of the Prony polynomial, we obtain the corresponding eigenfunctions $\tilde{v}_{j,1}, \dots, \tilde{v}_{j,r}$, and afterwards compute the complex coefficients $c_{j,\ell}$ by solving the overdetermined system

$$F(\mathcal{A}^k f) = \sum_{j \in J} \sum_{\ell=1}^{r} c_{j,\ell} \mathcal{A}^k \tilde{v}_{j,\ell}, \qquad k = 0, \dots, 2rM-1.$$

■

Remark 1.5 *Generalized Prony polynomials of the form* (1.16) *have already been used for the reconstruction of spline functions in* [54] *and* [38].

In this chapter we saw how the Prony Method can be derived from considerations about linear difference equations and how we can generalize the method to (generalized) eigenfunctions of linear operators. Until now we have not considered applicational problems that may arise, as for example the actual implementation for an algorithm or the handling of unknown sparsity M. This will be our focus in the next two chapters.

2 Generalized Prony Method for Unknown Order of Sparsity M

The algorithms 1.1.1 and 1.2.1 we saw in the introductory chapter need the exact number M of sparsity in the expansion (1.8) as input data. In practice one is rarely provided with the exact value of M, but only with an upper bound $L \geq M$. In this chapter we will see that this is no major drawback and expound how we can adapt algorithm 1.2.1 to this setting.

Let $\mathcal{A}$ be a linear operator as introduced in chapter 1.2 and

$$f(x) = \sum_{j=1}^{M} c_{n_j} v_{n_j}(x) \tag{2.17}$$

be a sparse expansion of eigenfunctions $v_{n_j}(x)$ to the eigenvalues λ_{n_j} of $\mathcal{A}$. Assume we have $2N \geq 2L \geq 2M$ sampling points of the form

$$h(k) = F(\mathcal{A}^k f), \quad k = 0, \dots, 2N-1,$$

where L is an a priorly known upper bound of the sparsity M in (2.17). There are three ways that come immediately to mind to construct a Hankel matrix to put the oversampled data $h(k)$, $k = 0, \dots, 2N-1$ into the first step of the algorithm. We can use $\mathbf{H}_N := (h(k+\ell))_{k,\ell=0}^{N-1} \in \mathbb{C}^{N \times N}$ which employs all the given information (except $h(2N-1)$), but can be a really big matrix, or we use $\mathbf{H}_L := (h(k+\ell))_{k,\ell=0}^{L-1} \in \mathbb{C}^{L \times L}$ which generally is of convenient size but neglects all information stored in $h(m)$, $m = 2L-1, \dots, 2N-1$. Another way is to discard the square structure of the Hankel matrix $\mathbf{H}$ and to change over to a rectangular matrix $\mathbf{H}_{2N-L+1,L} := (h(k+\ell))_{k,\ell=0}^{2N-L,L-1}$. Of course this changes the problem of solving the linear system in step one of algorithm 1.2.1 into a least square problem, but it uses all given information while the order of the polynomial in step 2 of algorithm 1.2.1 (which is defined by the number of columns of $\mathbf{H}$) still remains to be L.

Before we start our investigation with $\mathbf{H}_L := (h(k+\ell))_{k,\ell=0}^{L-1}$, we will consider a different perspective to algorithm 1.2.1. In (1.10) we used the information that the leading coefficient $p_M = 1$ of the Prony polynomial $P(z)$ is one, to reduce the number of input data by one. Thus we have to solve a linear system in the first step of algorithm 1.2.1. If we neglect the information $p_M = 1$, we can rewrite (1.10) into a homogenous system

$$\begin{aligned} \sum_{k=0}^{M} p_k F(\mathcal{A}^{k+\ell} f) = 0, \quad \ell = 0, \dots, M, \\ \mathbf{H}_{M+1}\mathbf{p} = \mathbf{0}, \end{aligned} \tag{2.18}$$

with $\mathbf{H}_{M+1} := (h(k+\ell))_{k,\ell=0}^{M}$, $\mathbf{p} := (p_k)_{k=0}^{M}$ and $\mathbf{0} \in \mathbb{C}^{M+1}$. Of course, equation (2.18) is still a linear system, but we can also interpret it as the eigenvalue problem of finding an eigenvector $\mathbf{p}$ to the eigenvalue 0 of $\mathbf{H}_{M+1}$.

To make this approach applicable we have to show first, that $\mathbf{H}_{M+1}$ is singular. Indeed, we will show that $\dim(\ker(\mathbf{H}_{L+1})) = L - M + 1$ for $L \geq M$. Furthermore, we have to show that any vector $\mathbf{u}$ from the kernel of $\mathbf{H}_{L+1}$ defines a polynomial $U(z) = \sum_{k=0}^{L} p_k z^k$ that contains (amongst others) all desired roots λ_{n_j}, $j = 1, \ldots, M$.

For the original Prony Method, i.e. $\mathbf{H}_{L+1} = (h(k+\ell))_{k,\ell=0}^{L}$, $h(k) = \sum_{j=1}^{M} c_j \, \mathrm{e}^{T_j k}$, $c_j \in \mathbb{C}\backslash\{0\}, T_j \in [-\pi, \pi)$, it was shown in [39] that $\dim(\ker(\mathbf{H}_{L+1})) = L - M + 1$. We adapt that theorem to the generalized Prony Method. Using the coefficients p_k, $k = 0, \ldots, M$ of the Prony polynomial $P(z) = \prod_{j=1}^{M}(z - \lambda_{n_j}) = \sum_{k=0}^{M} p_k z^k$ we construct the vector $\mathbf{p}$, where $p_k = 0$, $k = M+1, \ldots, L$. By $\mathbf{S} := (\delta_{k-\ell-1})_{k,\ell=0}^{L}$ we denote the forward shift matrix.

Theorem 2.1 *Let $L \geq M$ be given and let $h(k) := F(\mathcal{A}^k f)$, with $f(x)$ as defined in (2.17). Then the Hankel matrix $\mathbf{H}_{L+1} := (h(k+\ell))_{k,\ell=0}^{L}$ has the eigenvalue 0, where the kernel of $\mathbf{H}_{L+1}$ has the form*

$$\ker(\mathbf{H}_{L+1}) = \operatorname{span}\{\mathbf{p}, \mathbf{S}\mathbf{p}, \ldots, \mathbf{S}^{L-M}\mathbf{p}\}.$$

Proof. 1. We show that $\mathbf{S}^m\mathbf{p}$, $m = 0, \ldots, L-M$ is contained in the kernel of $\mathbf{H}_{L+1}$. From setting $(p_{k,m})_{k=0}^{L} := \mathbf{S}^m\mathbf{p}$, $m = 0, \ldots, L-M$ it follows

$$p_{k,m} = \begin{cases} p_{k-m}, & (k-m) = 0, \ldots, M \\ 0, & \text{otherwise.} \end{cases}$$

For $m = 0, \ldots, L-M$ we get for the ℓ-th component of the vector $\mathbf{H}_{L+1}\mathbf{S}^m\mathbf{p}$

$$\begin{aligned}
&(\mathbf{H}_{L+1}\mathbf{S}^m\mathbf{p})_\ell \\
&= \sum_{k=0}^{L} p_{k,m} F\left(\mathcal{A}^{k+\ell} f\right) \\
&= \sum_{k=0}^{m-1} \underbrace{p_{k,m}}_{=0} F\left(\mathcal{A}^{k+\ell} f\right) + \sum_{k=m}^{M+m} \underbrace{p_{k,m}}_{=p_{k-m}} F\left(\mathcal{A}^{k+\ell} f\right) + \sum_{k=M+m+1}^{L} \underbrace{p_k}_{=0} F\left(\mathcal{A}^{k+\ell} f\right) \\
&= \sum_{k=0}^{M} p_k F\left(\mathcal{A}^{k+m+\ell} f\right) = \sum_{k=0}^{M} p_k F\left(\sum_{j=1}^{M} c_{n_j} \lambda_{n_j}^{k+m+\ell} v_{n_j}\right) \\
&= \sum_{j=1}^{M} c_{n_j} \lambda_{n_j}^{\ell+m} \left(\sum_{k=0}^{M} p_k \lambda_{n_j}^{k}\right) F v_{n_j} = \sum_{k=0}^{M} c_{n_j} \lambda_{n_j}^{\ell+m} \underbrace{P(\lambda_{n_j})}_{=0} F v_{n_j}.
\end{aligned}$$

2. Let $\mathbf{u}$ be any vector from the kernel of $\mathbf{H}_{L+1}$, i.e. $\mathbf{H}_{L+1}\mathbf{u} = \mathbf{0}$. We show that

$\mathbf{u} \in \operatorname{span}\{\mathbf{p}, \mathbf{Sp}, \ldots, \mathbf{S}^{L-M}\mathbf{p}\}$. Indeed, we have

$$\begin{aligned}
0 &= \sum_{k=0}^{L} u_k F\left(\mathcal{A}^{k+\ell} f\right) \\
&= \sum_{\ell=0}^{L} \left(\sum_{k=0}^{L} u_k F\left(\mathcal{A}^{k+\ell} f\right) \right) z^{\ell}, \quad z \in \mathbb{C}\backslash\{0\} \\
&= \sum_{j=1}^{M} \sum_{\ell=0}^{L} \sum_{k=0}^{L} u_k F\left(c_{n_j} \lambda_{n_j}^{k+\ell} v_{n_j}\right) z^{\ell} \\
&= \sum_{j=1}^{M} c_{n_j} \sum_{\ell=0}^{L} (\lambda_{n_j} z)^{\ell} \sum_{k=0}^{L} u_k \lambda_{n_j}^{k} F v_{n_j} \\
&= \sum_{j=1}^{M} \underbrace{c_{n_j}}_{\neq 0} Q(\lambda_{n_j} z) U(\lambda_{n_j}) \underbrace{F v_{n_j}}_{\neq 0},
\end{aligned}$$

with $Q(z) := \sum_{k=0}^{L} z^k$ and $U(z) := \sum_{k=0}^{L} u_k z^k$. This means that we obtain a linear combination of the polynomials $Q(\lambda_{n_j} z)$, where the coefficients depend on the values $U(\lambda_{n_j})$. In the next step we show that the set of polynomials $\{Q(\lambda_{n_j} z) | j = 1, \ldots, M\}$ is linearly independent, which means that the linear combination above can only add up to zero if all coefficients $U(\lambda_{n_j})$ are zero. Or in other words, that the polynomial $U(z) = P(z)U_0(z)$ is a product of the Prony polynomial $P(z)$ and a certain polynomial $U_0(z)$ of degree $L - M - 1$.

3. Let q_j $j = 1, \ldots, M$ be coefficients of a linear combination of the polynomials $Q(\lambda_{n_j} z)$, such that this weighted sum adds up to 0,

$$0 = \sum_{j=1}^{M} q_j Q(\lambda_{n_j} z) = \sum_{j=1}^{M} q_j \sum_{k=0}^{L} \lambda_{n_j}^{k} z^k = \sum_{k=0}^{L} z^k \left(\sum_{j=1}^{M} q_j \lambda_{n_j}^{k} \right).$$

Since this equation holds for arbitrary $z \in \mathbb{C}\backslash\{0\}$ we get

$$\begin{aligned}
\sum_{j=1}^{M} q_j \lambda_{n_j}^{k} &= 0, \quad k = 0, \ldots, L \\
\mathbf{Zq} &= \mathbf{0},
\end{aligned}$$

with $\mathbf{q} := (q_j)_{j=1}^{M}$ and a Vandermonde like system matrix $\mathbf{Z} := (\lambda_{n_j}^{k})_{k=0,j=1}^{L,M}$ which has full rank, since the eigenvalues λ_{n_j} are pairwise distinct. This shows that $q_j = 0$ for all $j = 1, \ldots, M$ and thus we have shown the linear independence of the polynomials $Q(\lambda_{n_j} z)$.

4. Now, that we know from step 2 that $U(z) = P(z)U_0(z)$, for a polynomial

$U_0(z) = \sum_{k=0}^{L-M} \mu_k z^k$ we get

$$\begin{aligned} U(z) = P(z)U_0(z) &= \sum_{k=0}^{M} p_k z^k \sum_{\ell=0}^{L-M} \mu_\ell z^\ell \\ &= u_0 + u_1 z + \cdots + u_L z^L \\ &= p_0\mu_0 + z(p_0\mu_1 + p_1\mu_0) + \cdots + z^L(p_M\mu_{L-M}). \end{aligned}$$

By comparing coefficients we get

$$\begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_L \end{pmatrix} = \begin{pmatrix} p_0\mu_0 \\ p_0\mu_1 + p_1\mu_0 \\ \vdots \\ p_M\mu_{L-M} \end{pmatrix}$$

$$= \mu_0 \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ 0 \end{pmatrix} + \mu_1 \begin{pmatrix} 0 \\ p_0 \\ p_1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \mu_{L-M} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ p_M \end{pmatrix}$$

This shows that any vector $\mathbf{u}$ from the kernel of $\mathbf{H}_{L+1}$ is a linear combination of the vectors $\mathbf{S}^k\mathbf{p}$, $k = 0, \ldots, L-M$

$$\mathbf{u} = \sum_{k=0}^{L-M} \mu_k \mathbf{S}^k \mathbf{p}.$$

■

When we have no a priori knowledge of the sparsity M, we need to calculate it during the algorithm. We want to present two ideas here. The first one is to use theorem 2.1, because it tells us that the rank of $\mathbf{H}_{L+1}$ is equal to M. Thus, we can estimate the rank of $\mathbf{H}_{L+1}$ in a first step and use the outcome as the sparsity number M. At this stage let us point out that the outcome $\tilde{M}$ of a rank estimate might vary from the original M in the case of erroneous input data.

The second idea is to calculate all roots ξ_m, $m = 1, \ldots, L$ of the polynomial $U(z) := \sum_{k=0}^{L} u_k z^k$, defined via $\mathbf{H}_{L+1}\mathbf{u} = \mathbf{0}$. Afterwards we find a least square solution $\tilde{\mathbf{c}}$ of the overdetermined Vandermonde type system

$$\left(\xi_m^k\right)_{k=0,m=1}^{2N-1,L} \tilde{\mathbf{c}} = (h_k)_{k=0}^{2N-1},$$

and discard all roots ξ_m for which $|\tilde{c}_m|$ is smaller than a previously defined threshold ε.

Finding the roots of a polynomial of high order is not trivial and therefore the second idea of finding ξ_m, $m = 0, \ldots, N$ might not be applicable to the case of using a large square matrix $\mathbf{H}_{N+1}$. As stated before, we can overcome this

problem by changing the eigenvalue problem $\mathbf{H}_{L+1}\mathbf{u} = \mathbf{0}$ into a singular value problem $\mathbf{H}_{2N-L,L+1}\mathbf{u} = \mathbf{0}$, and end up with a *Prony-related-polynomial* $U(z)$ of lower order $L < N$, while still using all given information about the function f.

Remark 2.2 *As it is done in [39], we can also formulate a version of theorem 2.1 for the rectangle matrix* $\mathbf{H}_{2N-L,L+1}$*, which states*

$$\ker(\mathbf{H}_{2N-L,L+1}) = \operatorname{span}\{\mathbf{p}, \mathbf{S}\mathbf{p}, \ldots, \mathbf{S}^{L-M}\mathbf{p}\},$$

where $\mathbf{p}$ *is a right singular vector to the singular value* 0 *of* $\mathbf{H}_{2N-L,L+1}$*. But since the proof follows similar lines as the proof of theorem 2.1 it is omitted.*

We can now formulate two variants of a generalized Prony algorithm where the sparsity M is not known a priori. Here the symbol $\mathbf{H}$ stands for either $\mathbf{H}_{L+1}, \mathbf{H}_{N+1}$ or $\mathbf{H}_{2N-L,L+1}$.

2.1 Algorithm with Rank Estimate

Input: $h(k) = F(\mathcal{A}^k f)$, $k = 0, \ldots, 2N-1$

1. Estimate the rank M of $\mathbf{H}$.
2. (a) Find all roots λ_{n_j}, $j = 1, \ldots, M$ of $P(z) = \sum_{k=0}^{M} p_k z^k$, where $\mathbf{p} = (p_k)_{k=0}^{M}$ is found as the eigenvector of the eigenproblem

$$(h(k+\ell))_{k,\ell=0}^{M} (p_k)_{k=0}^{M} = (0)_{k=0}^{M}.$$

   Alternatively:

   (b) Find all roots λ_{n_j}, $j = 1, \ldots, M$ of $P(z) = \sum_{k=0}^{M} p_k z^k$, where $\mathbf{p} = (p_k)_{k=0}^{M}$ is found as the singular vector of the singular value problem

$$(h(k+\ell))_{k,\ell=0}^{2N-M-1,M} (p_k)_{k=0}^{M} = (0)_{k=0}^{2N-M-1}.$$

3. Calculate c_{n_j}, $j = 1, \ldots, M$ as least square solution of

$$\left(\lambda_{n_j}^k\right)_{k=0,j=1}^{2N-1,M} (c_{n_j})_{j=1}^{M} = (h(k))_{k=0}^{2N-1}.$$

Output: M, c_{n_j}, v_{n_j}, $j = 1, \ldots, M$.

A MATLAB implementation of this algorithm can look the following way. Here we used the eigenvalue approach. The singular value approach will be shown in the implementation of algorithm 2.2.

MATLAB implementation of algorithm 2.1

```
% input column vector %h%, with more than 2M values.
N = round(length(h)/2);
M = rank(hankel(h(1:N),h(N:end)));
H = hankel(h(1:M+1),h(M+1:2*M+1));
[U,W]= eigs(H,1,0);
Lambda = roots(flipud(U));
V = construct_V(Lambda,s);
c = V\h;
```

If the number of input values in the implementation above is odd, the matrix constructed for the rank calculation is a square matrix in $\mathbb{C}^{N\times N}$. Note that in the case of odd N the call `round(N/2)` returns $(N+1)/2$. If the number of input values is even, the matrix returned by `hankel` is rectangular and an element of $\mathbb{C}^{N\times N+1}$. The function `eigs(H,1,0)` returns the eigenvalue closest to zero and the corresponding eigenvector of $\mathbf{H}$. After running the algorithm, the sparsity of the signal is stored in M, the eigenvalues in $\mathbf{\Lambda}$ and the corresponding coefficients in $\mathbf{c}$. Note that again the implementation of `construct_V` is not only dependent on the found eigenvalues $\mathbf{\Lambda}$ but also on the operator $\mathcal{A}$ and the functional F. That is why we have not shown an implementation of this function here. For an example see the implementation at the end of chapter 1.2.

2.2 Algorithm with Coefficient Threshold

Input: L,ε, $h(k) = F(\mathcal{A}^k f)$, $k = 0, \ldots, 2N-1$

1. Find all roots ξ_m, $m = 1, \ldots, L$, of $U(z) = \sum_{k=0}^{L} u_k z^k$, where $\mathbf{u} = (u_k)_{k=0}^{L}$ is found as the eigenvector of the eigenvalue problem (resp. singular value problem) $\mathbf{Hu} = \mathbf{0}$.

2. Calculate $\tilde{c}_m$, $m = 1, \ldots, L$, as least square solution of

$$\left(\xi_m^k\right)_{k=0,m=1}^{2N-1,L} \left(\tilde{c}_m\right)_{m=1}^{L} = \left(h(k)\right)_{k=0}^{2N-1}.$$

3. Discard all pairs $(\xi_m, \tilde{c}_m)$ with $|\tilde{c}_m| < \varepsilon$ and rename the remaining parameters as λ_{n_j} and c_{n_j}, $j = 1, \ldots, M$. Here, M is the quantity of the remaining pairs λ_{n_j}, c_{n_j}.

Output: M, c_{n_j}, v_{n_j}, $j = 1, \ldots, M$

MATLAB implementation of algorithm 2.2

```
% input signal %h% as a column vector.
% input an upper bound %L% for the sparsity of %f%.
% input threshold parameter %epsilon%.
H = transpose(hankel(h(1:L+1),h(L+1:end)));
[W,D,U]= svd(H);
D = diag(D);
smallest_SV = find(D == min(D));
Lambda = roots(flipud(U(:,smallest_SV)));
V = construct_V(Lambda,s);
c = V\h;
places = find(abs(c) > epsilon);
Lambda_end = Lambda(places);
c_end = c(places);
M = length(Lambda_end);
```

In contrast to the implementation of algorithm 2.1 we used here the function `svd(H)` instead of `svds(H,1,0)`, because MATLAB has often problems to find a singular vector to the smallest singular value of a rank deficient matrix. Furthermore, we use the column of the matrix $\mathbf{U}$ corresponding to the smallest entry in the vector $\mathbf{D}$ instead of the corresponding row in $\mathbf{W}$ to evaluate the roots $\lambda_1, \ldots, \lambda_L$, because $\mathbf{U} \in \mathbb{C}^{(L+1)\times(L+1)}$, whereas $\mathbf{W} \in \mathbb{C}^{(2N-L)\times(2N-L)}$. In this way we have to calculate the roots of a polynomial of order $L+1$ instead of order $2N-L$ where the latter can be considerably larger.

Remark 2.3 *Often, the eigenvalues λ_j, $j \in J$ are well separated or they all lie on a lower dimensional manifold, e.g. on the unit circle. We may use this a priori knowledge to discard all roots ξ_m in step 1 of the last algorithm, whose distance to the set $\Lambda = \{\lambda_j | j \in J\}$ is greater than a suitable threshold δ.*

Example 2.4 *An example gives us an impression of the different behavior of the ideas stated above. For the operator $\mathcal{A}$ we use the translation operator and the function we want to analyze will be*

$$f(x) = \sum_{j=1}^{7} c_j \mathrm{e}^{\lambda_j x},$$

with

j	c_j	λ_j
1	1.0	0.3i
2	-1.0	0.8i
3	2.0	1.1i
4	0.5	1.4i
5	0.2	2.1i
6	-3.0	2.3i
7	-1.0	$(0.8+a)$i.

At this stage we have not taken erroneous input data into account, i.e. we work with correct data in this example. However, the accuracy of the algorithms output is not only dependent on the error of the input data, but also on the smallest distance of two active eigenvalues λ_m and λ_{m+1}, as will be examined in chapter 3. Therefore, we look here at the different behaviors of the algorithms in dependence of the smallest distance z, $|z| < 0.2$, between λ_2 and λ_7, as well as at the speed differences for increasing numbers N of input data.

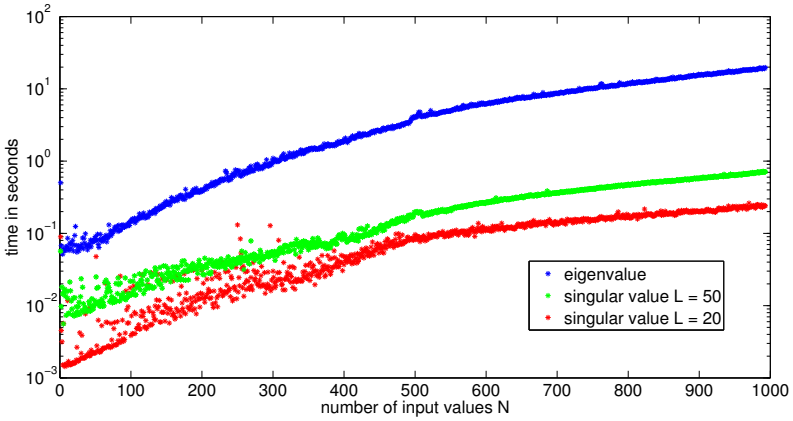

Figure 1: The red stars indicate the total time needed in seconds to run the generalized Prony algorithm with a rectangular Hankel matrix and the svd approach in dependency of the number of rows N of $\mathbf{H}_{2N-L,L+1}$ and for $L = 20$. The green stars show the running time for $L = 50$ whereas the blue stars indicate the time needed for the square matrix $\mathbf{H}_{N+1}$ and the eigenvalue approach.

Since the running time for the svd based algorithm with $L = 20$ is almost always about 100 times faster than the eigenvalue approach, we used a logarithmic scale for the measured time in order to illustrate the time differences throughout largely varying running times. We see in figure 1 that the algorithm based on a singular value decomposition with a good approximate $L(= 20)$ to $M = 7$ is considerably faster than the algorithm based on an eigenvalue decomposition of a square matrix $\mathbf{H}_{N+1}$, when all given data are used in both versions. The total time for all 1000 runs accumulates to 1.5 *minutes for the svd approach with $L = 20$ and to* 4 *minutes for $L = 50$. The eigenvalue decomposition approach on the other hand needs roughly* 1.5 *hours. All tests are made on a standard computer with a* 2.4 GHz *processor and* 4 GB RAM. *This test shows that the svd approach outperforms the eigenvalue decomposition approach vastly in terms of computational time. In the next test, when we will look at accuracy, it will turn out that the eigenvalue approach shows advantages over the svd approach. For that reason both approaches are presented here and none of the two algorithm variants can be regarded as superior in all circumstances.*

When the separation distance

$$q := \min\{\|\lambda_{n_j} - \lambda_{n_s}\|_2 \,|\, j, s \in \{1, \dots, M\}\}$$

becomes very small, it is reasonable to assume that the algorithm runs into difficulties for resolving all active eigenfunctions $v_{n_j}(x)$ in (2.17) separately. On the other hand, if we allow more input data we will generally observe a greater interference difference between two different active eigenvalues at increased time. This effect (called beat in acoustics) will be more dominant the more input data we use. That is why we hope to resolve the active eigenvalues λ_{n_j} better, the more input data we use. In Table 1 we see the performance of the eigenvalue decomposition approach (resp. svd approach) for $q = 0.0001$ and increasing number N of input data. Here, let $\boldsymbol{\lambda} = (\lambda_{n_j})_{j=1}^{M}, \tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_{n_j})_{j=1}^{M}$, with λ_{n_j} denoting the eigenvalue corresponding to the active eigenfunction $v_{n_j}(x)$ of the example signal and $\tilde{\lambda}_{n_j}$ the according computed approximation.

N	$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty^{\text{eig}}$	$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty^{\text{svd}}, L = 8$	$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty^{\text{svd}}, L = 20$	$\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty^{\text{svd}}, L = 50$
8	$1.14 \cdot 10^{-7}$	$1.71 \cdot 10^{-7}$	–	–
16	$1.22 \cdot 10^{-10}$	$4.10 \cdot 10^{-9}$	$7.80 \cdot 10^{-10}$	–
32	$2.58 \cdot 10^{-11}$	$1.31 \cdot 10^{-9}$	$4.53 \cdot 10^{-11}$	$5.55 \cdot 10^{-10}$
64	$9.33 \cdot 10^{-12}$	$8.06 \cdot 10^{-10}$	$4.80 \cdot 10^{-11}$	$9.71 \cdot 10^{-12}$
128	$8.17 \cdot 10^{-13}$	$2.33 \cdot 10^{-10}$	$7.79 \cdot 10^{-12}$	$4.02 \cdot 10^{-12}$
256	$9.05 \cdot 10^{-14}$	$1.77 \cdot 10^{-10}$	$2.17 \cdot 10^{-12}$	$6.80 \cdot 10^{-13}$

Table 1: Accuracy differences of algorithms 2.1 and 2.2.

We observe an immense accuracy improvement for increasing the number N of input data for the eigenvalue approach, but a less strong improvement for the svd approach if L is fixed. If we, however, increase the upper bound L we experience more accurate calculations. This way we can tune the tradeoff between computational time and accuracy with the upper bound L.

Interestingly, the algorithm with the svd approach performs really well even in the case of $L = 20, N = 16$ which leads to an underdetermined Hankel-system with a Hankel-matrix $\mathbf{H}_{32-20,20+1} \in \mathbb{C}^{12\times 21}$. Until now we always claimed $L < N$ and ended up with a rectangular Hankel matrix that has more rows than columns. But if $N - L$ is still larger than M, apparently the algorithm still returns good results. The same holds for $L = 50, N = 32$.

Algorithm 2.1 and 2.2 are generalizations of the Approximate Prony Method (APM) in terms of eigenfunctions of linear operators. This variant of the Prony Method was introduced in [39] for exponential functions. In the next chapter we will examine also other implementations of the Prony Method and generalize them too, to eigenfunctions of special linear operators.

3 Numerical Behavior of the Prony Method

The Prony Method is now more than 200 years old. But it was almost forgotten and undergoes now a revival. One reason for that is, that the method was marked as being unstable. In [33] on pages 276-280, C. Lanczos gives an example of the instability of Prony's method. He uses the function

$$f(x) = 0.0951\mathrm{e}^{-x} + 0.8607\mathrm{e}^{-3x} + 1.5576\mathrm{e}^{-5x},$$

takes 24 equidistant sampling points $f(\xi)$, $\xi = 0, 1/20, \ldots, 23/20$ and rounds them to the nearest cent. Then he uses Prony's method to find the unknwon exponents $-1, -3, -5$ and the corresponding coefficients out of the erroneous input data. The outcome of this test is that the Prony Method fails to recover the three unknown exponents but returns only two exponentials $\mathrm{e}^{-4.45}$ and $\mathrm{e}^{-1.58}$. But nevertheless the input data and the approximation constructed with the two found exponentials differ only in the range of the noise level.

The proposed test for the Prony Method in [33] inherits two hard difficulties. At first there is made a big failure on the measurements by rounding them to two digits after the dot. Thus, the signal strength is in the range of the noise level. Further, the problem introduced above has an intrinsic difficulty due to the sensitivity of exponential functions to noise. C. Lancos states at page 279 of [33]:
"It would be idle to hope that some other modified mathematical procedure could give better results, since the difficulty lies not with the manner of evaluation but with the extraordinary sensitivity of the exponents and amplitudes to very small changes of the data, which no amount of least-square or other form of statistics could remedy. The only remedy would be an increase of accuracy . . . "

This test does not mean that we have to give up the Prony Method because it is unstable, but that we can not hope to resolve the underlying functions with a high accuracy, when the noise level exceeds the signal strength. Indeed Lanczos himself states in the last sentence on page 280 in [33]:
"On the other hand, the picture would have been quit different if our data had had a ten times greater accuracy."

That is to say we have to ensure a certain amount of accuracy and can confidently use Prony's method. And then we can of course ask the question whether there is a numerical variant of Prony's method that solves the problem better than others. In this chapter we want to deal with this question.

3.1 Prony-like Methods

Only recently, it was shown by Potts and Tasche [43] that the ESPRIT Method [48] by Roy and Kailath, the Matrix Pencil Method [28] by Hua and Sarkar and the Approximate Prony Method (APM) [39] by Potts and Tasche can be seen as Prony-like-methods, i.e., all those methods solve the same kind of problems with the same core ideas. This means that those methods can be seen as different (stable) numerical realizations of Prony's method. In the upcoming subchapters 3.1.1, 3.2 on the ESPRIT and Matrix Pencil Method we will follow

the steps in [43] and adapt the calculations to our generalized Prony Method for eigenfunctions of operators.

Let $\mathcal{A}$ be a linear operator with a set of eigenfunctions $V := \{v_\ell | \mathcal{A}v_\ell = \lambda_\ell v_\ell, \ell \in I\}$ and a corresponding set of eigenvalues $\Lambda := \{\lambda_\ell, \ell \in I\}$ for an (possibly infinite) index set I. We consider a finite linear combination of eigenfunctions v_ℓ,

$$f(x) := \sum_{j=1}^{M} c_{n_j} v_{n_j}(x). \tag{3.19}$$

In the following we want to recover all unknown parameters M, c_{n_j}, v_{n_j} out of $N \geq 2M$ sampling points

$$F\left(\mathcal{A}^k f\right) := F\left(\sum_{j=1}^{M} c_{n_j} \lambda_{n_j}^k v_{n_j}\right), \quad k = 0, \ldots, N,$$

according to (1.9). Let us reintroduce the Prony polynomial

$$P(z) := \prod_{j=1}^{M} (z - \lambda_{n_j}) = \sum_{k=0}^{M} p_k z^k,$$

with leading coefficient $p_M = 1$, the corresponding vector $\mathbf{p} := (p_k)_{k=0}^{M-1}$ and the companion matrix

$$\mathbf{C}(P) := \begin{pmatrix} 0 & 0 & \ldots & 0 & -p_0 \\ 1 & 0 & \ldots & 0 & -p_1 \\ 0 & 1 & \cdots & 0 & -p_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -p_{M-1} \end{pmatrix}. \tag{3.20}$$

The eigenvalues of $\mathbf{C}(P)$ are the zeroes λ_{n_j} of the corresponding polynomial $P(z)$, [8]. For simplicity we define $h(k) := F(\mathcal{A}^k f)$ and with

$$\mathbf{H}_M(s) := (h(s + m + \ell))_{m,\ell=0}^{M-1}, \quad s = 0, 1 \tag{3.21}$$

$$\mathbf{H}_M(0) = \begin{pmatrix} h(0) & \ldots & h(M-1) \\ \vdots & \ddots & \vdots \\ h(M-1) & \ldots & h(2M-2) \end{pmatrix},$$

$$\mathbf{H}_M(1) = \begin{pmatrix} h(1) & \ldots & h(M) \\ \vdots & \ddots & \vdots \\ h(M) & \ldots & h(2M-1) \end{pmatrix}$$

we denote two Hankel matrices whose entries are given input data. We saw in (1.10) that $\mathbf{p}$ is a solution of the square Yule-Walker system $\mathbf{H}_M(0)\mathbf{p} =$

$-(h(k))_{k=M}^{2M-1}$. Or in other words, $\mathbf{p}$ is the (unique) coefficient vector for the linear combination of the last column of $\mathbf{H}_M(1)$ out of all columns of $\mathbf{H}_M(0)$. With this algebraic point of view we can extend the vector $-\mathbf{p}$ with a forward shift matrix $(\delta(\ell - m + 1))_{m,\ell=0}^{M-1,M-2}$ to a full $M \times M$ matrix $\mathbf{C}(P)$ as in (3.20), and we can formulate the useful relation

$$\mathbf{H}_M(0)\mathbf{C}(P) = \mathbf{H}_M(1). \tag{3.22}$$

Equation (3.22) is closely related to a square matrix pencil problem [24] pp.251.

Definition 3.1 *Let $\mathbf{A}, \mathbf{B}$ be two $M \times M$ matrices. The set of all matrices of the form $\mathbf{A} - \lambda\mathbf{B}$ with $\lambda \in \mathbb{C}$ is said to be a* matrix pencil. *The eigenvalues of the matrix pencil are elements of the set $\lambda(\mathbf{A}, \mathbf{B})$ defined by*

$$\lambda(\mathbf{A}, \mathbf{B}) := \{\lambda \in \mathbf{C} | \det(\mathbf{A} - \lambda\mathbf{B}) = 0\}.$$

Definition 3.2 *With*

$$\lambda(\mathbf{H}) := \{\lambda_j | \mathbf{H}\mathbf{x}_j = \lambda_j \mathbf{x}_j, j = 1, \ldots, N\}.$$

we denote the set of eigenvalues of a non-defective matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$.

Now we can formulate the lemma that connects the Prony Method with the Matrix Pencil Method.

Lemma 3.3 *The eigenvalues of the matrix pencil $\mathbf{H}(1), \mathbf{H}(0)$ are exactly the eigenvalues of the matrix $\mathbf{C}(P)$.*

Proof. Let $\mathbf{x}$ be an eigenvector of the matrix $\mathbf{C}(P)$ to an eigenvalue λ_{n_j}. We will see, that this assumption is enough to show $\lambda(\mathbf{C}(P)) = \lambda(\mathbf{H}(1), \mathbf{H}(0))$. Consider equation (3.22)

$$\begin{aligned} \mathbf{H}_M(0)\mathbf{C}(P) &= \mathbf{H}_M(1) \\ \mathbf{H}_M(0)\mathbf{C}(P)\mathbf{x} - \mathbf{H}_M(1)\mathbf{x} &= \mathbf{0} \\ \left[\lambda_{n_j}\mathbf{H}_M(0) - \mathbf{H}_M(1)\right]\mathbf{x} &= \mathbf{0}. \end{aligned}$$

■

Lemma 3.3 shows that we can calculate the unknown active eigenvalues λ_{n_j} by solving the matrix pencil problem $\mathbf{H}(1)\mathbf{x} = \lambda\mathbf{H}(0)\mathbf{x}$. Later on we will see the details of algorithms that are either based on QR-, QZ-, or SV-decomposition. But before we go into details we will extend the previous approach to unknown order of sparsity M.

In practice we rarely have exact a priori knowledge of the sparsity M in the linear combination (3.19), but one is often provided with an upper bound L such that $M \leq L \leq \lfloor N/2 \rfloor$. As demonstrated in chapter 2, there are basically three ways to put the given data into the algorithm. The simplest case is to refuse to use the knowledge of an upper bound $L \geq M$ and construct a large square

Hankel matrix $\mathbf{H}_{\lfloor N/2 \rfloor+1} = (h(m+\ell))_{m,\ell=0}^{\lfloor N/2 \rfloor}$ in order to use as many given data as possible. As we saw, this is computationally very expensive. If just a small square Hankel matrix $\mathbf{H}_{L+1} = (h(m+\ell))_{m,\ell=0}^{L}$ is constructed, we are not using all the given data which leads to less accurate results. A way in between is to construct a rectangular matrix $\mathbf{H}_{N-L,L} = (h(m+\ell))_{m,\ell=0}^{N-L-1,L-1}$ that uses all given data and preserves stability, but at the expense of a more complicated analysis.

In the case of $L > M$, let $\mathbf{q}$ denote the vector that solves $\mathbf{H}_{N-L,L+1}\mathbf{q} = \mathbf{f}_{N-L} := (h(L+m))_{m=0}^{N-L-1}$ and whose entries are the coefficients q_k, $k = 0, \ldots, L-1$ of the Prony-related-polynomial

$$Q(z) = \sum_{k=0}^{L} q_k z^k = \prod_{j=1}^{M} (z - \lambda_{n_j}) R(z) = P(z)R(z).$$

Here, $q_L = 1$ and $R(z)$ is a residual polynomial of degree $L-M$, i.e. all zeroes of $P(z)$ are also zeroes of $Q(z)$. Note that the polynomial $Q(z)$ is not uniquely determined, since $\mathbf{H}_{N-L,L}$ is rank deficient. If we define the rectangular analogon of (3.21)

$$\mathbf{H}_{N-L,L}(s) := (h(s+m+\ell))_{m,\ell=0}^{N-L-1,L-1}, \quad s = 0, 1,$$

the relation

$$\mathbf{H}_{N-L,L}(0)\mathbf{C}(Q) = \mathbf{H}_{N-L,L}(1)$$

still holds, with the same argumentation as above. If we replace the eigenvector $\mathbf{x}$ in Lemma 3.3 by a right singular vector, the same lines in the proof can be used to show that the eigenvalues λ_{n_j}, $j = 1, \ldots, L$ of $\mathbf{C}(Q)$ are eigenvalues of the rectangular matrix pencil problem

$$\mathbf{H}_{N-L,L}(1)\mathbf{x} = \lambda \mathbf{H}_{N-L,L}(0)\mathbf{x}. \tag{3.23}$$

If an upper bound $L \leq N$ of the sparsity M is known, we can either construct a square Hankel matrix of convenient size and solve an eigenvalue problem or we construct a rectangular Hankel matrix and solve a singular value problem. The first approach leads to the Matrix Pencil Method, whereas the latter leads to the ESPRIT Method.

3.1.1 ESPRIT

In [43] it was shown that the ESPRIT Method (estimation of signal parameters via rotational techniques) [48] can be seen as a variant of Prony's method for sparse sums of exponentials. The advantage of this numerical realization of Prony's method is the avoidance of calculating the coefficients of the Prony polynomial and its roots. As we saw in the original algorithm 1.1.1, we first have to solve the system $\mathbf{Hp} = \mathbf{0}$ and afterwards find all roots of the Prony polynomial $P(z) = \sum_{j=0}^{M} p_j z^j$. Numerically we find the roots of the Prony

polynomial as the eigenvalues of the companion matrix $\mathbf{C}(P)$ (3.20). In the introduction of this chapter we saw, how these steps can be elegantly combined in the generalized eigenvalue problem $\mathbf{H}_M(0)\mathbf{C}(P) = \mathbf{H}_M(1)$. In this chapter we will exemplarily show for the ESPRIT Method, that this variant of Prony's method does not rely on features of the exponential sums in the original Prony Method and thus can be used for the generalized Prony Method as well.

We follow the lines in [43] and adapt the proofs to the case of sparse representations of eigenfunctions of linear operators.

Let $L \leq N$ be an upper bound on the sparsity M, when $2N$ data of the form $F(\mathcal{A}^\ell f)$, $\ell = 0, \ldots, 2N-1$ are given. The ESPRIT Method is based on a singular value decomposition. Let us start with the factorization

$$\mathbf{H}_{2N-L,L+1} = \mathbf{U}_{2N-L}\mathbf{D}_{2N-L,L+1}\mathbf{W}_{L+1} \tag{3.24}$$

of

$$\mathbf{H}_{2N-L,L+1} := \left(F(\mathcal{A}^{\ell+m}f)\right)_{\ell,m=0}^{2N-L-1,L}, \tag{3.25}$$

where $\mathbf{U}_{2N-L}$ and $\mathbf{W}_{L+1}$ are unitary matrices and where $\mathbf{D}_{2N-L,L+1}$ is a rectangular diagonal matrix. By construction, the singular values of $\mathbf{H}_{2N-L,L+1}$ are the diagonal entries of $\mathbf{D}_{2N-L,L+1}$. Let the rows of $\mathbf{W}_{L+1}$ and the columns of $\mathbf{U}_{2N-L}$ be arranged in such a way that the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{L+1}$ are nonincreasingly ordered. We use the known submatrix notation of $\mathbf{A}(a:b, c:d)$ to select the rows a to b and the columns c to d of $\mathbf{A}$ and we define

$$\begin{aligned}\mathbf{D}_{2N-L,M} &:= \mathbf{D}_{2N-L,L+1}(1:2N-L, 1:M) = \begin{pmatrix}\operatorname{diag}(\sigma_j)_{j=1}^M \\ \mathbf{0}_{2N-L-M,M}\end{pmatrix} \\ \mathbf{W}_{M,L+1} &:= \mathbf{W}_{L+1}(1:M, 1:L+1).\end{aligned}$$

This is reasonable since we know from remark (2.2) of Theorem (2.1) that the rank of $\mathbf{H}_{2N-L,L+1}$ equals the sparsity M in the representation of f in (3.19). For that reason we have $\sigma_{M+1} = \sigma_{M+2} = \cdots = \sigma_{L+1} = 0$ in the noiseless case and we can focus on the range of $\mathbf{H}_{2N-L,L+1}$. Another way to look at it is, to estimate the rank of $\mathbf{H}_{2N-L,L+1}$ first and than apply the ESPRIT Method for known sparsity. The simplified version of (3.24) is

$$\mathbf{H}_{2N-L,L+1} = \mathbf{U}_{2N-L}\mathbf{D}_{2N-L,M}\mathbf{W}_{M,L+1}.$$

Similarly to the notation (3.21) we define also for $\mathbf{W}_{M,L+1}$ the matrix consisting of the first L respectively last L columns as

$$\mathbf{W}_{M,L}(s) := \mathbf{W}_{M,L+1}(1:M, 1+s:L+s), \quad s \in \{0,1\}. \tag{3.26}$$

With this notation we can write

$$\mathbf{H}_{2N-L,L}(s) = \mathbf{U}_{2N-L}\mathbf{D}_{2N-L,M}\mathbf{W}_{M,L}(s),$$

because the selection of columns we make in $\mathbf{W}_{M,L+1}$ in the above representation is the same selection of columns in $\mathbf{H}_{2N-L,L+1}$. Since $\mathbf{U}_{2N-L}$ is unitary the generalized eigenvalue problem of the rectangular matrix pencil $z\mathbf{H}_{2N-L}(0) - \mathbf{H}_{2N-L}(1)$ is equivalent to the generalized eigenvalue problem of the matrix pencil

$$z\mathbf{D}_{2N-L,M}\mathbf{W}_{M,L}(1) - \mathbf{D}_{2N-L,M}\mathbf{W}_{M,L}(0) \tag{3.27}$$

Note that $\mathbf{D}_{2N-L,M}\mathbf{W}_{M,L}(s) = \operatorname{diag}(\sigma_1, \ldots, \sigma_M)\mathbf{W}(1 : M, 1 + s : M + s)$ since the last rows of $\mathbf{D}_{2N-L,M}$ are zero. Therefore, by multiplying (3.27) from the left with $\mathbf{D}_{2N-L,M}^{-\mathrm{T}}$, the matrix pencil (3.27) reduces to

$$z\mathbf{W}(1 : M, 1 : M) - \mathbf{W}(1 : M, 2 : M + 1).$$

But we can also use the trick that was introduced in [43] and multiply the transposed matrix pencil (3.27) from the right with

$$\begin{pmatrix} \operatorname{diag}(\sigma_1^{-1}, \ldots, \sigma_M^{-1}) \\ \mathbf{0}_{2N-L-M,M} \end{pmatrix}$$

and obtain the generalized eigenvalue problem of the matrix pencil

$$z\mathbf{W}_{M,L}(0)^{\mathrm{T}} - \mathbf{W}_{M,L}(1)^{\mathrm{T}},$$

which has the same eigenvalues as the matrix pencil (3.27) except for the zero eigenvalues and the advantage that more input data are used for the construction of this matrix pencil, whenever $L > M$. Finally we determine the eigenvalues λ_j of the active eigenfunctions v_j as eigenvalues of the matrix

$$\mathbf{Z}_M^{SVD} := \left(\mathbf{W}_{M,L}(0)^{\mathrm{T}}\right)^{\dagger} \mathbf{W}_{M,L}(1)^{\mathrm{T}}.$$

Thus the ESPRIT algorithm reads as follows:

3.1.2 ESPRIT Algorithm

Input: $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2N - 1$, $L \geq M$, ε

1. Compute the singular value decomposition (3.24). Determine the rank and construct the matrices $\mathbf{W}_{M,L}(0), \mathbf{W}_{M,L}(1)$ as in (3.26).

2. Compute all eigenvalues $\lambda_1, \ldots, \lambda_M$ of the square matrix

$$\mathbf{Z}_M^{SVD} := \left(\mathbf{W}_{M,L}(0)^{\mathrm{T}}\right)^{\dagger} \mathbf{W}_{M,L}(1)^{\mathrm{T}}.$$

3. Compute the coefficients c_j by solving the overdetermined system

$$F(\mathcal{A}^k f) = \sum_{j=1}^{M} c_j \lambda_j^k v_j, \quad k = 0, \ldots, 2N - 1.$$

Note, that for determining the rank of the matrix $\mathbf{H}_{2N-L,L-1}$ we calculate the quotients $\sigma_1/\sigma_0, \ldots, \sigma_L/\sigma_0$ and set the index of the first σ_M with $\sigma_M/\sigma_0 < \varepsilon$ as the rank of $\mathbf{H}_{2N-L,L+1}$, as it was proposed in [44]. Let us look at a MATLAB implementation.

MATLAB implementation of the ESPRIT algorithm 3.1.2

```
% input signal column vector %h%, with 2N values.
% input upper bound %L% of the sparsity number $M$.
% input threshold %epsilon%.
H = transpose(hankel(h(1:L+1),h(L+1:end)));
[U,V,W] = svd(H);
M1 = find(diag(V)/V(1,1) < epsilon);
M = M1(1)-1;
W1 = W(1:L,1:M);
W2 = W(2:L+1,1:M);
Z = W1\W2;
Lambda = eig(Z);
V = construct_V(Lambda);
c = V\h;
```

Of course the line `M = M1(1)-1;` can become problematic if the threshold ε is not chosen suitably.

3.2 Matrix Pencil

In [43] Potts and Tasche have also shown that the Matrix Pencil Method discussed for example in [25, 28] can be seen as a Prony-like method. The main difference to ESPRIT is that the first step of algorithm 3.1.2 does not rely on a singular value decomposition but on a QR decomposition. The steps to show that also the Matrix Pencil Method is a Prony-like method are similar to those we saw when we treated the ESPRIT Method and are therefore omitted here. For details we refer to [43] and just state the algorithm.

3.2.1 Matrix Pencil Algorithm

Input: $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2N-1$, $L \geq M$, ε

1. (a) Compute a QR decomposition of the matrix $\mathbf{H}_{2N-L,L+1} = (F(\mathcal{A}^{k+\ell}))_{k,\ell=0}^{2N-L-1,L}$ with a permutation matrix $\mathbf{\Pi}_{L+1}$ such that

$$\mathbf{H}_{2N-L,L+1}\mathbf{\Pi}_{L+1} = \mathbf{Q}_{2N-L}\mathbf{R}_{2N-L,L+1},$$

   with non-increasing diagonal entries of $\mathbf{R}_{2N-L,L+1}$.

   (b) Set $\mathbf{D} = \operatorname{diag}(\mathbf{R}_{2N-L,L+1})$ and determine the rank M of $\mathbf{H}_{2N-L,L+1}$.

(c) Set $\mathbf{D}_M = \mathbf{D}(1:M, 1:M)$ and construct the matrices

$$\mathbf{S}_{M,L}(s) = \mathbf{D}_M^{-1}\mathbf{R}_{2N-L,L+1}\mathbf{\Pi}_{L+1}^{\mathrm{T}}(1:M, 1+s:L+s), \quad s = 0, 1.$$

2. Compute all eigenvalues $\lambda_1, \ldots, \lambda_M$ of the square matrix

$$\mathbf{Z}_M^{QR} = ((\mathbf{S}_{M,L}(0)^{\mathrm{T}})^{\dagger}\mathbf{S}_{M,L}(1))^{\mathrm{T}}$$

3. Compute the coefficients c_j by solving the overdetermined system

$$F(\mathcal{A}^k f) = \sum_{j=1}^{M} c_j \lambda_j^k v_j, \quad k = 0, \ldots, 2N.$$

We determine the rank of the matrix $\mathbf{H}_{2N-L,L+1}$ as we did for the ESPRIT algorithm by calculating the quotients $\mathbf{D}(1,1)/\mathbf{D}(0,0), \ldots, \mathbf{D}(L,L)/\mathbf{D}(0,0)$ and setting the row index M of the first diagonal entry $\mathbf{D}(M,M)$ with $\mathbf{D}(M,M)/\mathbf{D}(0,0) < \varepsilon$ as the rank of $\mathbf{H}_{2N-L,L+1}$. Let us look at a MATLAB implementation.

MATLAB implementation of the Matrix Pencil Method 3.2.1

```
% input signal column vector %h%, with 2N values.
% input upper bound %L% of the sparsity number $M$.
% input threshold %epsilon%.
H = transpose(hankel(h(1:L+1),h(L+1:end)));
[Q,R,P] = qr(H); %i.e.  H*P = Q*R
D = diag(R);
M1 = find(D/D(1) < epsilon);
M = M1(1)-1;
DM = inv(diag(D(1:M)));
S = R*P'; %thus H = Q*S
S0 = DM*S(1:M,1:L);
S1 = DM*S(1:M,2:L+1);
Z = pinv(transpose(S0))*transpose(S1);
Lambda = eig(Z);
V = construct_V(Lambda);
c = V\f;
```

3.3 MUSIC

We have seen that the ESPRIT Method is a Prony-like method. When searching the literature about the ESPRIT Method one inevitably comes across the MUSIC Method (multiple signal classification) [49] as another parameter identification method for trigonometric functions. We will first state the classical MUSIC Method for exponentials following the presentation in [53] and see that it also is a Prony-like method.

Let

$$f(x) := \sum_{j=1}^{M} c_j \mathrm{e}^{\omega_j x},$$
$$\tilde{f}(x) := \sum_{j=1}^{M} c_j \mathrm{e}^{\omega_j x} + e(x),$$

be a signal with additional noise $e(x)$ with zero mean and variance σ^2, $c_j \in \mathbb{R}$ and $\mathrm{Re}(\omega_j) = 0$. Let L be an upper bound for the unknown sparsity M. We set

$$\begin{aligned}\mathbf{A} &:= (\mathrm{e}^{\omega_j k})_{k=0,j=1}^{L,M}, \quad \in \mathbb{C}^{(L+1)\times M},\\ \mathbf{y}(t) &:= (c_j \mathrm{e}^{\omega_j t})_{j=1}^{M},\\ \mathbf{e}(t) &:= (e(t+k))_{k=0}^{L},\end{aligned}$$

then

$$\begin{aligned}\mathbf{f}(t) &:= (f(t+k))_{k=0}^{L} = \mathbf{A}\mathbf{y}(t),\\ \tilde{\mathbf{f}}(t) &:= (f(t+k))_{k=0}^{L} = \mathbf{A}\mathbf{y}(t) + \mathbf{e}(t).\end{aligned} \tag{3.28}$$

Since ω_j, ω_ℓ can be considered independent random variables we observe

$$\begin{aligned}\mathrm{E}\{c_j \mathrm{e}^{\omega_j t} c_\ell \mathrm{e}^{-\omega_\ell (t-k)}\} &= c_j^2 e^{\omega_j k}\delta_{j,\ell},\\ \mathrm{E}\{f(t)f^*(t-k)\} &= \sum_{j=1}^{M} c_j^2 \mathrm{e}^{\omega_j k} + \sigma^2\delta_{k,0}.\end{aligned} \tag{3.29}$$

What we like to have in order to perform the MUSIC algorithm is the covariance matrix

$$\mathbf{R} := \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \tilde{\mathbf{f}}(t)\tilde{\mathbf{f}}^*(t),$$

with $\tilde{\mathbf{f}}(t)$ as in (3.28), but that implies that we need access to infinitely many sampling points. According to the available data we estimate the covariance matrix as the truncated sum

$$\hat{\mathbf{R}} := \frac{1}{L}\sum_{t=1}^{L} \mathbf{f}(t)\mathbf{f}^*(t).$$

From (3.29) and (3.28) we get

$$\mathbf{R} = \mathbf{A}\mathbf{P}\mathbf{A}^* + \sigma^2\mathbf{I}$$

for $\mathbf{P} := \mathrm{diag}(c_1^2, \ldots, c_M^2)$. Let us assume that $c_j \gg \sigma^2$, $j = 1, \ldots, M$, then, for the non increasing ordered eigenvalues λ_k, $k = 1, \ldots, L$ of $\mathbf{R}$ holds

$$\begin{aligned}\lambda_k &> \sigma^2, \quad k = 0, \ldots, M-1,\\ \lambda_k &= \sigma^2, \quad k = M, \ldots, L,\end{aligned}$$

since the rank of $\mathbf{APA}^*$ is M.

Now let $\mathbf{g}_m \in \mathbb{C}^{L+1}$, $m = 0, \ldots, L$, be the eigenvectors of $\mathbf{R}$ corresponding to the eigenvalues λ_k, $k = 0, \ldots, L$. From the structure of the set of eigenvalues it follows that the first M eigenvectors $\mathbf{g}_m$, $m = 1, \ldots, M-1$ form the signal subspace, whereas the last $L - M$ eigenvectors $\mathbf{g}_m$, $m = M, \ldots, L$ form the noise only subspace. This means, for $\mathbf{G} := [\mathbf{g}_M, \ldots, \mathbf{g}_L]$ we get $\mathbf{A}^*\mathbf{G} = \mathbf{0}$. In particular this last equation tells us that the noise space is orthogonal to the signal subspace. Schmidt suggested in [49] to search through $\omega \in [-\pi, \pi]$ for steering vectors $\mathbf{v}(\omega) := (\mathrm{e}^{\omega k})_{k=0}^{L}$ that fulfill

$$\mathbf{v}^*(\omega)\mathbf{G}\mathbf{G}^*\mathbf{v}(\omega) = 0$$

and then look for poles of the function

$$T(\omega) = \frac{1}{\mathbf{v}^*(\omega)\mathbf{G}\mathbf{G}^*\mathbf{v}(\omega)}. \tag{3.30}$$

In order to avoid confusions later on, we refer to this original method as the non root MUSIC Method. In [46] Rao suggests the root MUSIC algorithm that finds the unknown parameters ω_j, $j = 1, \ldots, M$, as the roots of a certain polynomial. Rao states:
"In Spectral MUSIC, a primary motivation for computing the null spectra was the fact that

$$\mathbf{A}^*\mathbf{g}_m = 0, \quad m = M+1, \ldots, L,$$

[...]. Therefore, if we define polynomials using the eigenvectors corresponding to the noise subspace, i.e.

$$\mathbf{g}_m(z) = \sum_{k=0}^{L} g_{k,m} z^k, \quad m = M+1, \ldots, L, \tag{3.31}$$

then $z_j = \mathrm{e}^{\omega_j}$, $j = 1, \ldots, M$, the signal zeroes, are roots of each of the above polynomials."

Now we have everything together to see, that the root MUSIC algorithm is nothing but the Prony Method as we have seen it in algorithm 2.2. Note that if we define $\mathbf{H} := (f(k+m))_{k,m=0}^{L}$, as in the Prony Method we get

$$(\mathbf{H}\mathbf{H}^*)_{k,m} := \sum_{\ell=0}^{L} h_{k,\ell}\bar{h}_{\ell,m} = \sum_{\ell=0}^{L} f(\ell+k)f^*(\ell+m) = (\hat{\mathbf{R}})_{k,m}.$$

Thus, the matrix $\tilde{\mathbf{R}}$ that is used in the MUSIC algorithm is basically the squared Hankel-matrix $\mathbf{H}$ in the Prony Method. From $\mathbf{R}\mathbf{g}_m = \mathbf{0}$ we derive

$$\begin{aligned} \mathbf{R}\mathbf{g}_m &= \mathbf{0} \\ \mathbf{g}_m^*\mathbf{R}\mathbf{g}_m &= 0 \\ \mathbf{g}_m^*\mathbf{H}\mathbf{H}^*\mathbf{g}_m &= 0 \\ (\mathbf{H}^*\mathbf{g}_m)^*\mathbf{H}^*\mathbf{g}_m &= 0 \\ \Leftrightarrow \mathbf{H}\mathbf{g}_m &= \mathbf{0}. \end{aligned}$$

In other words, the eigenvectors corresponding to the eigenvalue zero of $\mathbf{R}$ coincide with the eigenvectors of $\mathbf{H}^*$ corresponding to the eigenvalue zero in the absence of noise. Therefore, the polynomials defined in (3.31) are Prony related Polynomials $Q(z)^*$ as in algorithm 2.2, with zeroes $\bar{\lambda}_j$, $j = 1, \ldots, M$. Note that in the presence of noise, the eigenvectors of $\mathbf{H}^*$ corresponding to the smallest eigenvalues do not necessarily have to coincide with the eigenvectors corresponding to the smallest eigenvalues of the estimation $\hat{\mathbf{R}}$ to the matrix $\mathbf{R}$. Therefore we can expect different results for the MUSIC Method and the APM.

Even though we saw that the MUSIC Method is a Prony-like method, it turns out that it does not perform as well as the methods we have investigated before. Apparently, squaring the input data prior to using them in a Prony-like method affects the output quality. For that reason we will not generalize the MUSIC algorithm to our generalized Prony Method.

Let us look at a comparative test example between the MUSIC Method and the APM.

Example 3.4 *We have seen in example 2.4 that it is a hard problem to detect a small minimal separation distance*

$$q := \{|\lambda_j - \lambda_s| \,|\, j, k \in \{1, \ldots, M\}\}$$

between eigenvalues λ_j of active eigenfunctions v_j, $j = 1 \ldots, M$ in the signal $f = \sum_{j=1}^{M} c_j v_j$. Consider the function

$$f(x) = \sin(2.3x) - 2\cos(0.1x) + \mathrm{e}^{2.301\mathrm{i}x},$$

with $q = 0.001$. We oversample the signal f and use 23 input values $f(0)$, $\ldots, f(22)$ without an a priori knowledge about an upper bound L for the actual sparsity $M = 5$. Therefore, the two matrices we use in the MUSIC Method and in the APM are elements of $\mathbb{C}^{12\times 12}$. In Table 2 we compare the computed eigenvalues of the root MUSIC algorithm with those computed by the APM.

j	λ_j	$\tilde{\lambda}_j$ rootMUSIC	$\tilde{\lambda}_j$ APM
1	−2.3i	−2.299999918157941i	−2.300000000000000i
2	−1i	−1.000089264421881i	−0.999999999999997i
3	1i	1.000000228773602i	1.000000000000011i
4	2.3i	−	2.300000000010592i
5	2.3001i	2.300803605521613i	2.300999999986370i

Table 2: Performance differences of the MUSIC Method and the APM.

We see that the root MUSIC Method has problems in separating eigenvalues with a small separation distance where the APM at the same time gives very accurate results. Let us also have a look at the function (3.30) of the non root MUSIC Method that is supposed to have singularities at 2.3 and 2.3001. In Figure 2 we see an excerpt of $T(\omega)$ in the neighborhood of $\omega = 2.3$.

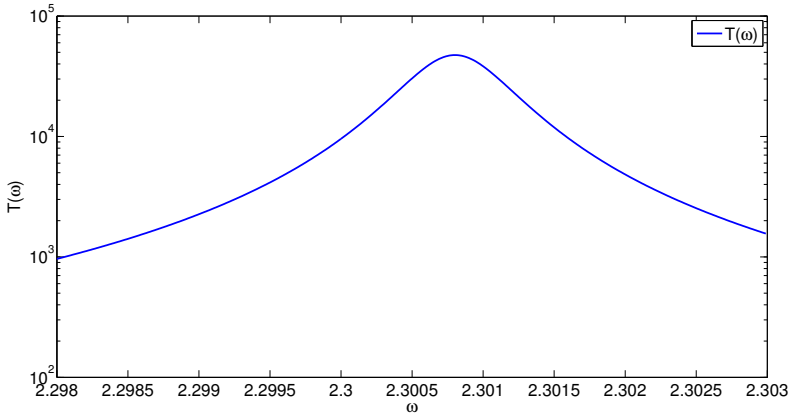

Figure 2: The blue line shows the characteristic of the function T_ω as introduced in 3.30 at the critical points 2.3 and 2.301.

The function does have a peak in the region around $\lambda_4 = 2.3$ and $\lambda_5 = 2.3001$, but it should have two singularities. Note, that the vertical axis is in logarithmic scale, such that we can observe more easily the missing second peak. This test shows that also the non root MUSIC Method does not perform better.

Remark 3.5 *The Pisarenko Method that was introduced in [37] is a special case of the MUSIC Method when the sparsity M is known beforehand, as it was shown for example in [53] .*

The huge performance differences between the MUSIC Method and the APM discards the MUSIC Method as an alternative method to the Prony-like methods we have studied before. We will have comparative tests between the APM, ESPRIT and the Matrix Pencil Method in subchapter 3.5, where we will see that these three numerical variants perform somewhat similar. Therefore, we have just tested the MUSIC Method against the APM.

3.4 Stability of the Approximate Prony Method

There are just a few stability results known for numerical realizations of the original Prony and the existing ones are usually not transferrable to our generalized Prony Method. In this section we want to list some known stability results for the case of the translation operator $\mathcal{A}f(x) := f(x+1)$ with the point evaluation functional $F(f) := f(0)$ and elicit their incapacity to perform as well for other linear operators.

In [39] the stability of the Approximate Prony Method (APM) is investigated. In other words, we are now looking at the special case were the signal f

has the form

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{\mathrm{i}T_j x},$$

with $c_j \in \mathbb{R}\backslash\{0\}$ and $T_j \in [-\pi, \pi)$, $j = 1, \ldots, M$. Let $h_k := f(k)$ denote the exact input data and let only perturbed input values $\tilde{h}_k = h_k + e_k$, $k = 0, \ldots, 2N-1$ be known. Here we assume that $|e_k| \leq \varepsilon$ for a certain accuracy ε. Furthermore we assume to know an upper bound L with $M \leq L \leq N$ of the sparsity M of f. Note that the failure matrix

$$\mathbf{E} = (e_{k+m})_{k,m=0}^{2N-L-1,L}$$

is also a rectangular Hankel matrix and we can denote the Hankel matrix $\tilde{\mathbf{H}}_{2N-L,L+1}$ that we use in the first step of algorithm 2.2 as $\tilde{\mathbf{H}}_{2N-L,L+1} = \mathbf{H}_{2N-L,L+1} + \mathbf{E}_{2N-L,L+1}$.

The analysis in [39] starts backwards with the accuracy of the calculations of the coefficients c_j in step 2 of algorithm 2.2. Let $\mathbf{V}$ be the Vandermonde-type matrix

$$\mathbf{V} = \left(\mathrm{e}^{\mathrm{i}kT_j}\right)_{k=0,j=1}^{2N,M}. \tag{3.32}$$

The quality of the calculated coefficients c_j is not only dependent on the accuracy of the input values but also on the minimal separation distance

$$q := \min\{|T_j - T_k| \,|\, j, k \in \{1, \ldots, M\}\},$$

as we saw in example 2.4. This observation can easily be justified by observing that a Vandermonde-type matrix is rank deficient iff at least two entries in one row are the same. This means, the condition number of $\mathbf{V}$ increases when the separation distance q decreases. When enough input data are available we can state an upper bound for the squared spectral norm of $\mathbf{V}$ and specify a left inverse $\mathbf{L}$ depending on N and q. Thereby also the condition number $\mathrm{cond}(\mathbf{V}) := \|\mathbf{L}\|_2 \|\mathbf{V}\|_2$ is bounded.

Lemma 3.6 *Let $N > \pi^2/q$ and $\mathbf{D} := \mathrm{diag}(1 - |k|/(N+1))_{k=-N}^{N}$ be a diagonal matrix, then*

$$\mathbf{L} := (\mathbf{V}^{\mathrm{H}}\mathbf{D}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{H}}\mathbf{D}$$

is a left inverse of $\mathbf{V}$ and the squared spectral norm of L can be estimated by

$$\|\mathbf{L}\|_2^2 \leq \frac{3}{2N+2}.$$

The squared spectral norm of $\mathbf{V}$ can be bounded by

$$\|\mathbf{V}\|_2^2 \leq 2N + 1 + \frac{2\pi}{q}\left(1 + \ln\frac{\pi}{q}\right).$$

Thus the condition cond($\mathbf{V}$) is bound by $\mathcal{O}(1/(Nq))$. For a proof see [39] chapter 4. The proof heavily relies on the fact that all entries of $\mathbf{V}$ have norm 1 and thus their magnitude is not exponentially increasing (respectively decreasing) with k in (3.32). That is the reason for the moderate increase of the condition number of this special Vandermonde-type matrix. Even for the translation operator and the neglect of the restriction of real valued parameters $T_j \in [-\pi, \pi)$, Lemma 3.6 does not hold anymore, even less for arbitrary linear operators $\mathcal{A}$ and eigenfunctions differing from the exponential functions.

Now let us look at a sensitivity result for the calculations of the coefficients c_j when only perturbed input data are given. Again we formulate the Lemma as it is stated in [39] and refer to the proof in that work.

Lemma 3.7 *Assume that* $|h_k - \tilde{h}_k| \leq \varepsilon$, $k = 0, \ldots, 2N-1$ *and* $T_j - \tilde{T}_j \leq \delta$, $j = 1, \ldots, M$. *Let* $\mathbf{V}$ *be given by 3.32 and* $\tilde{\mathbf{V}}$ *the perturbed version of* $\mathbf{V}$ *determined by approximations* $\tilde{T}_j$ *of* T_j. *Further let* $\mathbf{V}\boldsymbol{\rho} = \mathbf{h}$ *and* $\tilde{\boldsymbol{\rho}} = \tilde{\mathbf{L}}\tilde{\mathbf{h}}$, *where* $\tilde{\mathbf{L}} := (\tilde{\mathbf{V}}^{\mathrm{H}}\mathbf{D}\tilde{\mathbf{V}})^{-1}\tilde{\mathbf{V}}^{\mathrm{H}}\mathbf{D}$ *is a left inverse of* $\tilde{\mathbf{V}}$. *If the assumptions of Lemma 3.6 are fulfilled, then for each* $N \in \mathbb{N}$ *with*

$$N > \pi^2 \max\{q^{-1}, \tilde{q}^{-1}\}$$

the following estimate

$$\|\boldsymbol{\rho} - \tilde{\boldsymbol{\rho}}\| \leq \sqrt{6(N+1)(2M+1)}\|\mathbf{h}\|\delta + \sqrt{3}\varepsilon$$

is fulfilled.

Special properties of the Dirichlet Kernel applied to differences of exponential functions are used in the proof. Therefore, this result does not hold for other eigenfunctions of linear operators than exponentials with purely imaginary parameters T_j.

In practice, the condition $N > \pi^2/q$ is often an immense exaggeration of the actually need number of input values, but up until now it is an open question, if the bounds in Lemma 3.6 and 3.7 are tight.

Let $U(z) = \sum_{k=0}^{L} u_k z^k$ be the Prony-related polynomial which coefficients are calculated in step 1 of algorithm 2.2 and let $\tilde{U}(z)$ be the corresponding polynomial when only erroneous input data are available. In [39] Potts and Tasche also give a result which states that the perturbed polynomial $\tilde{U}(z)$ evaluated at the correct active eigenvalues e^{T_j} is always smaller than a certain bound. Which indicates that the roots of the Prony polynmial $\tilde{P}(z)$ are close to those corresponding ones of $\tilde{U(z)}$. But the determination of the bound relies on $\|\mathbf{V}\|$ and $\|\mathbf{L}\|$, for which we argued that they are not easily calculated for eigenfunctions of arbitrary linear operators $\mathcal{A}$. Therefore this result is omitted here.

3.5 Numerical Tests for Different Methods

Now that we have considered different numerical realizations for our generalized Prony Method, let us test if we can recognize performance differences. Consider

the function

$$f(x) = \mathrm{e}^{0.3\mathrm{i}x} - \mathrm{e}^{0.7\mathrm{i}x} + 2\cos(x) - 2\mathrm{e}^{2.3\mathrm{i}x} + 5\mathrm{e}^{2.9\mathrm{i}x}.$$

In this test scenario we use the translation operator $\mathcal{A}f(x) := f(x+1)$ and the point evaluation functional $F(f) := f(0)$ in order to analyze the signal f. We set $N = 40$, i.e. we have the input values $f(0), \ldots, f(79)$ and we set $L = 20$ as an upper bound for the actual sparsity $M = 6$. For the ESPRIT Method, the Matrix Pencil Method and the APM based on singular value decomposition we use the rectangular Hankel matrix $\mathbf{H}_{2N-L,L+1} = (f(k+m))_{k,m=0}^{2N-L-1,L}$ and for the APM based on eigenvalue decomposition we use the square matrix $\mathbf{H}_N = (f(k,m))_{k,m=0}^{N-1}$.

In our first test, we assume given perturbed values

$$\tilde{f}(k) = f(k) + e(k).$$

Here, $e(k)$ is additive white gaussian noise $e(k)$, with mean 0 and variance σ^2. We perform tests for 13 different variances $\sigma^2 = 10^{-14}, 10^{-13}, \ldots, 10^{-2}$ and for each choice of σ^2 we perturb the signal f with 500 different instances of additional noise e and average the results. More precisely, we compute the largest difference between the actual eigenvalues λ_j and the computed eigenvalues $\tilde{\lambda}_j$, $j = 1, \ldots, 6$ and average these values over the 500 calculations. Set $\boldsymbol{\lambda} = (\lambda_j)_{j=1}^6$ and $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_j)_{j=1}^6$, then we define for each method the reconstruction failure *err* as

$$err := \frac{1}{500} \sum_{k=1}^{500} \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty.$$

The results of all $500 \cdot 13 \cdot 4 = 26000$ calculations are presented in Table 3.

σ^2	*err* ESPRIT	*err* Matrix Pencil	*err* svd-Prony	*err* eig-Prony
10^{-14}	$1.53 \cdot 10^{-15}$	$1.47 \cdot 10^{-15}$	$3.63 \cdot 10^{-15}$	$3.04 \cdot 10^{-15}$
10^{-13}	$1.58 \cdot 10^{-15}$	$1.64 \cdot 10^{-15}$	$3.47 \cdot 10^{-15}$	$3.20 \cdot 10^{-15}$
10^{-12}	$5.38 \cdot 10^{-15}$	$8.14 \cdot 10^{-15}$	$1.72 \cdot 10^{-14}$	$1.51 \cdot 10^{-14}$
10^{-11}	$5.25 \cdot 10^{-14}$	$7.96 \cdot 10^{-14}$	$1.59 \cdot 10^{-13}$	$1.49 \cdot 10^{-13}$
10^{-10}	$5.24 \cdot 10^{-13}$	$8.02 \cdot 10^{-13}$	$2.72 \cdot 10^{-12}$	$1.56 \cdot 10^{-12}$
10^{-9}	$5.18 \cdot 10^{-12}$	$7.89 \cdot 10^{-12}$	$1.56 \cdot 10^{-11}$	$1.55 \cdot 10^{-11}$
10^{-8}	$5.27 \cdot 10^{-11}$	$8.05 \cdot 10^{-11}$	$1.42 \cdot 10^{-10}$	$1.53 \cdot 10^{-10}$
10^{-7}	$5.29 \cdot 10^{-10}$	$7.95 \cdot 10^{-10}$	$1.53 \cdot 10^{-09}$	$1.89 \cdot 10^{-09}$
10^{-6}	$5.15 \cdot 10^{-09}$	$8.03 \cdot 10^{-09}$	$1.67 \cdot 10^{-08}$	$1.68 \cdot 10^{-08}$
10^{-5}	$5.17 \cdot 10^{-08}$	$8.02 \cdot 10^{-08}$	$1.52 \cdot 10^{-07}$	$1.67 \cdot 10^{-07}$
10^{-4}	$5.25 \cdot 10^{-07}$	$8.00 \cdot 10^{-07}$	$1.71 \cdot 10^{-06}$	$1.52 \cdot 10^{-06}$
10^{-3}	$5.24 \cdot 10^{-06}$	$8.18 \cdot 10^{-06}$	$1.66 \cdot 10^{-05}$	$1.71 \cdot 10^{-05}$
10^{-2}	$5.25 \cdot 10^{-05}$	$7.93 \cdot 10^{-05}$	$1.49 \cdot 10^{-04}$	$1.64 \cdot 10^{-04}$
10^{-1}	$5.36 \cdot 10^{-04}$	$8.00 \cdot 10^{-04}$	$1.50 \cdot 10^{-03}$	$1.57 \cdot 10^{-03}$
10^{-0}	$5.49 \cdot 10^{-03}$	$1.25 \cdot 10^{-02}$	$1.36 \cdot 10^{-02}$	$1.27 \cdot 10^{-02}$

Table 3: Performance differences of ESPRIT, MUSIC and the APM.

We observe that both APM approaches give similar results. Note, that the singular value decomposition approach though is faster, as we saw in example 2.4. But we see that the Matrix Pencil Method gives twice as good results as the APM and the ESPRIT Method is even three times more accurate. However, in the presence of high noise, there were occasionally some test, were neither the ESPRIT nor the Matrix Pencil Method could recover all 6 active eigenfunctions in f, due to a falsely computed sparsity M. In the last row of Table 3 we cheated at the ESPRIT Method and MUSIC Method by setting $\tilde{M} = 6$ and not letting the algorithms find it during the computation. In this sense, the APM is more reliable when it comes to heavily corrupted data. Note that all methods give very accurate results in the presence of noise.

As we have already seen in example 2.4, a small separation distance $q := \{|\lambda_j - \lambda_s| \,|\, j,k \in \{1,\dots,M\}\}$ of the eigenvalues corresponding to the active eigenfunctions in the signal f can cause numerical difficulties. For that reason we consider now the function

$$f(x) = \mathrm{e}^{0.3\mathrm{i}x} - \mathrm{e}^{0.7\mathrm{i}x} + 2\cos(x) - 2\mathrm{e}^{2.3\mathrm{i}x} - \mathrm{e}^{(2.3+z)\mathrm{i}x} + 5\mathrm{e}^{2.9\mathrm{i}x}. \tag{3.33}$$

for some small $z \in \mathbb{R}$ and test, how well the different methods can compute the eigenvalues λ_j, $j = 1,\dots,7$. Again we set $\boldsymbol{\lambda} = (\lambda_j)_{j=1}^7$, $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_j)_{j=1}^7$ and $err = \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty$. The performance tests are made for $z = 10^{-\ell/2}$, $\ell = 1,\dots,11$. For a minimal separation distance $q = 10^{-6}$ non of the methods is able to distinguish between the eigenvalues 2.3i and 2.300001i. Table 4 shows the accuracy results for the different methods.

z	*err* ESPRIT	*err* Matrix Pencil	*err* svd-Prony	*err* eig-Prony
$10^{-0.5}$	$2.13 \cdot 10^{-15}$	$1.83 \cdot 10^{-15}$	$2.04 \cdot 10^{-15}$	$1.77 \cdot 10^{-15}$
10^{-1}	$2.48 \cdot 10^{-15}$	$1.61 \cdot 10^{-15}$	$1.08 \cdot 10^{-14}$	$3.44 \cdot 10^{-15}$
$10^{-1.5}$	$5.06 \cdot 10^{-15}$	$2.66 \cdot 10^{-15}$	$2.00 \cdot 10^{-15}$	$2.31 \cdot 10^{-15}$
10^{-2}	$1.95 \cdot 10^{-14}$	$3.21 \cdot 10^{-15}$	$1.78 \cdot 10^{-14}$	$1.90 \cdot 10^{-14}$
$10^{-2.5}$	$2.48 \cdot 10^{-13}$	$4.53 \cdot 10^{-14}$	$9.57 \cdot 10^{-13}$	$3.21 \cdot 10^{-14}$
10^{-3}	$2.30 \cdot 10^{-12}$	$2.48 \cdot 10^{-13}$	$2.60 \cdot 10^{-12}$	$1.81 \cdot 10^{-12}$
$10^{-3.5}$	$2.43 \cdot 10^{-11}$	$1.76 \cdot 10^{-11}$	$1.41 \cdot 10^{-11}$	$3.26 \cdot 10^{-11}$
10^{-4}	$2.22 \cdot 10^{-10}$	$1.34 \cdot 10^{-10}$	$6.80 \cdot 10^{-11}$	$1.23 \cdot 10^{-10}$
$10^{-4.5}$	$3.42 \cdot 10^{-09}$	$4.49 \cdot 10^{-10}$	$1.09 \cdot 10^{-09}$	$1.06 \cdot 10^{-09}$
10^{-5}	$1.75 \cdot 10^{-08}$	$1.90 \cdot 10^{-08}$	$2.74 \cdot 10^{-09}$	$1.89 \cdot 10^{-08}$
$10^{-5.5}$	$1.42 \cdot 10^{-07}$	$1.04 \cdot 10^{-07}$	$2.44 \cdot 10^{-07}$	$1.90 \cdot 10^{-07}$

Table 4: Accuracy differences of ESPRIT, MUSIC and the APM.

Note that we established here a hard problem in distinguishing between eigenvalues of active eigenfunctions with so small a separation distance as $q \approx 3.16 \cdot 10^{-6}$. Because, if we consider a function f_2 were the summand $-2\mathrm{e}^{2.3\mathrm{i}x}$ in (3.33) is replaced by $-3\mathrm{e}^{2.3\mathrm{i}x}$ and the summand $-\mathrm{e}^{(2.3+z)ix}$ vanishes, then the maximal distance

$$\max_{k=0,\dots,79}(f(k) - f_2(k)) = 2.49 \cdot 10^{-4}$$

of both functions at the sampling points $k = 0, \ldots, 79$ is rather small.

In summary we can say that the Prony-like methods are high resolution methods that can distinguish between eigenvalues that are very close to one another in the absence of noise, but they still are powerful tools for analyzing noisy measurements, when the minimal separation distance is only reasonably small. Note that the difference between f and $\tilde{f}$ increases in time, since the effect of the beat frequency q intensifies over longer observation intervals. Therefore, we can expect better results if more input data are used. Indeed, if we set for example $q = 10^{-5.5}$, as in the last row of Table 4 and $N = 400$, $L = 200$ we get the following better results.

z	*err* ESPRIT	*err* Matrix Pencil	*err* svd-Prony	*err* eig-Prony
$10^{-5.5}$	$1.97 \cdot 10^{-10}$	$3.88 \cdot 10^{-10}$	$1.09 \cdot 10^{-09}$	$2.94 \cdot 10^{-10}$

All four variants of the Prony Method that we investigated in this chapter perform about the same, whereas the ESPRIT Method turns out to be slightly favorable over the others. In [53] Stoica and Moses elucidate the Pisarenko [37], the MUSIC [49] and the ESPRIT [48] method and conclude:
"All the high-resolution methods presented [...] provide very accurate frequency estimates, with only small differences in their performances. Furthermore, the computational burdens associated with these methods are rather similar. Hence, selecting one of the high-resolution methods for frequency estimation is essentially a **matter of taste***."*
We observed better results with the Prony-like methods compared to the (non root) MUSIC algorithm and therefore also the Pisarenko Method. And indeed the authors in [53] append after the passage cited above that the ESPRIT Method holds some advantages over the other methods. But the message that the choice (in one of the Prony-like methods) is principally a matter of taste, transfers still to our case.

4 Reconstruction of Sparse Expansions of Eigenfunctions of Linear Operators

The introductory chapter comprised the original Prony Method for exponential functions and a generalization to eigenfunctions of linear operators. In the second chapter we saw, how the generalized Prony algorithm can be adapted to the case when the sparsity M in (2.17) is not known a priori. Now, having algorithms that are also applicable if the sparsity M is unknown beforehand we want to present the range of possible applications of our generalized Prony Method. Therefore we will look at various examples of linear operators and their eigenfunctions. There are several further linear operators that may be used if we know their spectral structure, i.e. if we can give a suitable set of eigenfunctions that correspond to pairwise different eigenvalues.

4.1 Prony Method for Eigenfunctions of the Translation Operator

At the end of chapter one we have already considered the translation operator. But for completeness and since it inherits the original Prony Method we will revisit it here again.

Let us consider the vector space $C(\mathbb{R})$ of continuous functions, and let $S_a : C(\mathbb{R}) \to C(\mathbb{R})$ with

$$S_a f(x) := f(x+a), \quad a \in \mathbb{R}\backslash\{0\} \tag{4.34}$$

be the shift operator on $C(\mathbb{R})$. We observe that $\{\mathrm{e}^{Ta} : T \in \mathbb{C},\ \mathrm{Im}\, T \in [-\frac{\pi}{a}, \frac{\pi}{a})\}$ is a set of pairwise distinct eigenvalues of S_a, and by

$$S_a \mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})} = \mathrm{e}^{(x+a)(T+\frac{2\pi \mathrm{i} k}{a})} = \mathrm{e}^{Ta}\mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})}, \qquad x \in \mathbb{R},\ k \in \mathbb{Z},$$

we find for each eigenvalue $\lambda_T := \mathrm{e}^{Ta}$, $T \in I := \{T \in \mathbb{C},\ \mathrm{Im}\, T \in [-\frac{\pi}{a}, \frac{\pi}{a})\}$, the eigenspace $\mathcal{V}_T := \mathrm{span}\,\{\mathrm{e}^{x(T+\frac{2\pi \mathrm{i} k}{a})} : k \in \mathbb{Z}\}$. In order to obtain a unique correspondence between λ_T and its eigenfunction, we consider only the subeigenspaces $\tilde{\mathcal{V}}_T = \mathrm{span}\,\{\mathrm{e}^{Tx}\}$. Further, let the functional $F : C(\mathbb{R}) \to \mathbb{C}$ be given by

$$F(f) := f(x_0), \quad \forall f \in C(\mathbb{R}), \tag{4.35}$$

with an arbitrarily fixed $x_0 \in \mathbb{R}$. Hence $F(\mathrm{e}^{T\cdot}) = \mathrm{e}^{Tx_0} \neq 0$ for all $T \in I$. Applying Theorem 1.2 yields that the sparse sum of exponentials

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j x} \tag{4.36}$$

with pairwise different $T_j \in \mathbb{C}$ and $\mathrm{Im}\ T_j \in [-\frac{\pi}{a}, \frac{\pi}{a})$ can be uniquely reconstructed from the values

$$F(S_a^k f) = F(f(\cdot + ka)) = f(x_0 + ka), \quad k = 0, \ldots, 2M-1,$$

e.g., from $2M$ equidistant sampling points with sampling distance a, starting at point x_0. Let us specify an algorithm for the translation operator.

Algorithm 4.1 (Reconstruction of f in (4.36))

Input: Translation operator with $h \neq 0$, as well as the data $F(f)(x+hk)$, $k = 0, \ldots, 2N-1$. Optionally an upper bound L for the sparsity M, else $L = N$.

1. (a) Construct the Hankel matrix
$$\mathbf{H}_{2N-L,L+1} = (F(f)(x+h(k+m)))_{k,m=0}^{2N-L-1,L}$$
and use the ESPRIT Method 3.1.2, the Matrix Pencil Method 3.2.1 or use 2.1 respectively 2.2 with an svd approach to find the eigenvalues $\lambda_j = \mathrm{e}^{T_j}$, $j = 1, \ldots, M$ of the active eigenfunctions $\mathrm{e}^{T_j x}$.
Alternatively:
   (b) Construct the Hankel matrix $\mathbf{H}_{N-1} = (F(f)(x+h(k+m)))_{k,m=0}^{N-1}$ and use 2.1 or 2.2 with an eigenvalue decomposition approach to find the eigenvalues $\lambda_j = \mathrm{e}^{T_j}$, $j = 1, \ldots, M$ of the active eigenfunctions $\mathrm{e}^{T_j x}$.

2. Compute the coefficients c_j by solving the overdetermined system
$$F(f)(x+hk) = \sum_{j=1}^{M} c_j F(\mathrm{e}^{T_j(x+hk)}) \qquad k = 0, \ldots, 2N-1.$$

Output: M, c_j, T_j, $j = 1, \ldots, M$.

In radar, sonar, geophysical seismology and likewise areas the signals dealt with can often be well described by

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j x}, \quad T_j, c_j \in \mathbb{C}\backslash\{0\}, \tag{4.37}$$

see [53](p.139). Because it has numerous applications, the translation operator is an exceptional case and we will examine some examples in detail to see the method at work.

Assume $f(x)$ to be a real valued linear combination of sines and cosines

$$f(x) = \sum_{\ell=0}^{M} a_\ell \cos(\omega_\ell x) + \sum_{\ell=1}^{M} b_\ell \sin(\omega_\ell x), \tag{4.38}$$

with $\omega_\ell \in [-\pi, \pi)\backslash\{0\}, a_\ell, b_\ell \in \mathbb{R}$, $\ell = 1, \ldots, M$, $\omega_0 = 0$, $a_0 \in \mathbb{R}$. A few calculations show, that (4.38) is a special case of (4.37). Due to Euler's formula

$$\sin(x) = \frac{\mathrm{e}^{\mathrm{i}x} - \mathrm{e}^{-\mathrm{i}x}}{2\mathrm{i}}$$
$$\cos(x) = \frac{\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x}}{2},$$

we get

$$\begin{aligned} f(x) &= \sum_{\ell=0}^{M} a_\ell \frac{\mathrm{e}^{\mathrm{i}\omega_\ell x} + \mathrm{e}^{-\mathrm{i}\omega_\ell x}}{2} + \sum_{\ell=1}^{M} b_\ell \frac{\mathrm{e}^{\mathrm{i}\omega_\ell x} - \mathrm{e}^{-\mathrm{i}\omega_\ell x}}{2\mathrm{i}} \\ &= a_0 + \sum_{\ell=1}^{M} \left[\frac{a_\ell - \mathrm{i}b_\ell}{2} \mathrm{e}^{\mathrm{i}\omega_\ell x} + \frac{a_\ell + \mathrm{i}b_\ell}{2} \mathrm{e}^{-i\omega_\ell x} \right] \\ &= \sum_{j=-M}^{M} c_j \mathrm{e}^{T_j x}, \end{aligned}$$

with

$$c_j = \begin{cases} \frac{a_j - \mathrm{i}b_j}{2}, & j > 0 \\ \frac{a_j + \mathrm{i}b_j}{2}, & j < 0 \\ 0, & j = 0 \end{cases}, \quad T_j = \begin{cases} \omega_j \mathrm{i}, & j > 0 \\ -\omega_j \mathrm{i}, & j < 0 \\ 0, & j = 0 \end{cases}.$$

For all $j \neq 0$ we get $T_j = -T_{-j}$ and $c_j = \bar{c}_{-j}$. Let us look at an example.

Example 4.2 *Let $f(x)$ be a real valued function with four active terms ω_ℓ in the notation of (4.38) or eight active terms in the exponential notation (4.37)*

$$f(x) = \sum_{\ell=1}^{4} a_\ell \cos(\omega_\ell x) + \sum_{\ell=1}^{4} b_\ell \sin(\omega_\ell x), \tag{4.39}$$

with

j	ω_j	a_j	b_j
1	1	2	1
2	2.6	1	−1
3	2.7	1.4	−1
4	3.1	−2	2

Let us choose $F(f) := f(0)$, and use equidistant sampling points $f(k)$, $k = 0, \ldots, 30$ as input data. Furthermore we use an upper bound $L = 10 > 8$ for the sparsity M of $f(x)$. Note, that since we have an 8-sparse signal f, it would be sufficient to use 16 input data. But in practice M is often not known a priori. To resemble that scenario we have oversampled the signal. Furthermore, a higher number of input data can be used for a more stable calculation of the unknowns a_j, b_j, ω_j, if we are faced with numerical difficulties as a small separation distance q (see example 2.4) or erroneous input data, see subchapter 3.5. The function and the sampling points are shown in figure 3.

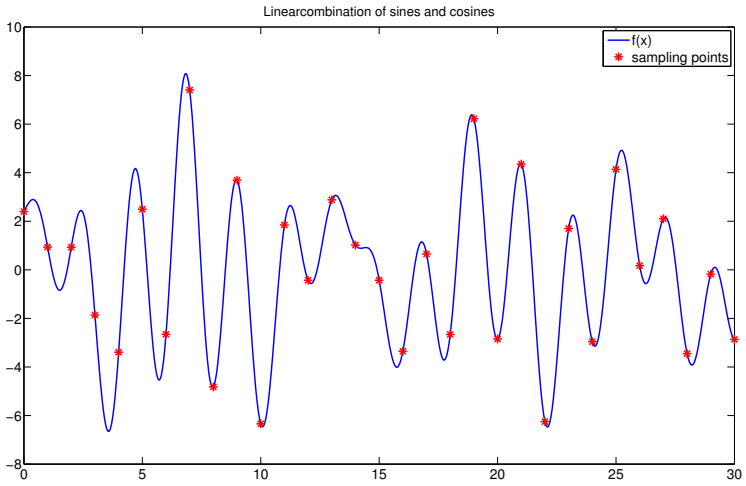

Figure 3: The blue line indicates the function $f(x)$, whereas the red stars show the sampling points.

Since we know an upper bound $L = 10$ of the sparsity $M = 8$ we can construct a rectangular Hankel matrix $\mathbf{H} = (f(k+m))_{k,m=0}^{21,9}$ and search for the right singular vector $\mathbf{p} = (p_k)_{k=0}^{9}$ corresponding to the smallest singular value as described in algorithm 2.1.

Let us stress again that the function $f(x)$ in (4.39) is a special case, where every T_j in 4.36 is purely imaginary and has an associated T_{-j} which is its negative. Therefore the roots of the Prony-polynomial have to lie on the unit circle and they come in complex conjugate pairs. This characteristic is demonstrated in Figure 4. Also, in this example we have to evaluate the roots of a polynomial $Q(z) = \sum_{k=0}^{9} p_k z^k$ which has one root more than we need. For that reason we see a ninth, unwanted root at approximately 0.95 in Figure 4. In the last step of our algorithm 2.2 of choice, the absolut value of the corresponding coefficient $\tilde{c}_9$ of $\tilde{z}_9 \approx 0.95$ will be smaller than the threshold $\varepsilon = 10^{-8}$ and therefor $\mathrm{e}^{0.95x}$ is not considered as an active eigenfunction of the translation operator for the present signal (4.39).

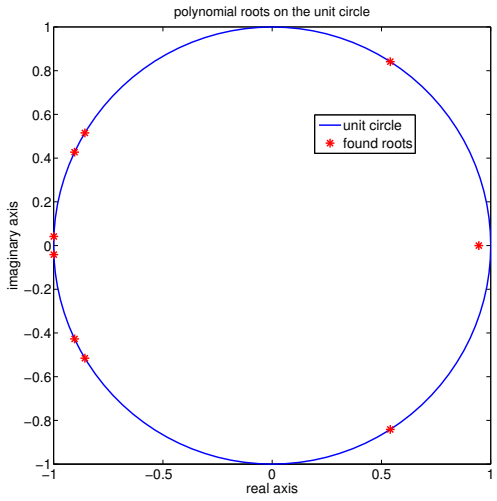

Figure 4: The red stars indicate the found roots of the Prony algorithm.

Indeed, the fact that all roots should lie on the unit circle can be used to stabilize the method for noisy input data. In the case of noisy data we usually consider only those roots z_j that lie close enough to the unit circle, i.e. $||z_j| - 1| < \epsilon$. Moreover, we can also use the information, that the roots come in complex conjugate pairs. In order to do so, we make a short excursion into symmetric polynomials.

4.1.1 Symmetric Prony Polynomials

Let $P_a(x) = a_0 x^{N-1} + a_1 x^{N-2} + \cdots + a_{N-1}$ be an algebraic polynomial of degree $N-1$. We say the polynomial P_a is a *symmetric polynomial* if the condition $a_k = a_{N-1-k}$ holds for all k of the *coefficient-vector* $\mathbf{a} = (a_0, \ldots, a_{N-1})$.

Definition 4.3 *Let* $\mathbf{a} \in \mathbb{R}^N$, $\mathbf{b} \in \mathbb{R}^M$ *with* $M \geq N \geq 1$. *The* discrete convolution $\mathbf{c} = \mathbf{a} * \mathbf{b}$ *is defined as*

$$c_n := \sum_{k=\max(0,n+1-M)}^{\min(n,N)} a_k b_{n-k}. \tag{4.40}$$

Let $\mathbf{a}, \mathbf{b}$ be the coefficient vectors of the polynomials $P_a(x), P_b(x)$ respectively and $\mathbf{c}$ is the coefficient vector of the product $P_c(x)$ of the two polynomials $P_a(x)$ and $P_b(x)$. It is well-known that the coefficient-vector $\mathbf{c}$ of the polynomial $P_c = P_a P_b$ can be evaluated via the discrete convolution of the coefficient-vectors $\mathbf{c} = \mathbf{a} * \mathbf{b}$, see e.g. [13], p. 823-825.

We now introduce the *flip matrix* $\mathbf{T}_N$ of order N which has ones on the counter diagonal and zeroes elsewise by

$$\mathbf{T}_N = (f_{k,l})_{k,l=1}^N, \quad f_{k,l} = \delta_{k,(N+1-l)},$$
$$\mathbf{T}_N = \begin{pmatrix} 0 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 0 \end{pmatrix}.$$

It is easy to see that a multiplication of $\mathbf{A}$ from the left by $\mathbf{T}_N$ flips the rows of the Matrix $\mathbf{A}$ and that a multiplication $\mathbf{A}\mathbf{T}_N$ from the right flips the columns of $\mathbf{A}$.

Remark 4.4 *Due to the commutativity of the discrete convolution the condition $M \geq N$ is no constriction.*
In matrix-vector-notation we can reformulate equation (4.40) as

$$\mathbf{B}\mathbf{a} = \mathbf{c}$$

with the matrix

$$\mathbf{B} = \begin{pmatrix} b_0 & 0 & \cdots & 0 \\ b_1 & b_0 & \cdots & 0 \\ \vdots & & & \vdots \\ b_{M-1} & b_{M-2} & \cdots & b_{M-N} \\ 0 & b_{M-1} & \cdots & b_{M-N+1} \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & b_{M-1} \end{pmatrix} \in \mathbb{R}^{(M+N-1)\times N}. \tag{4.41}$$

Now we can formulate the following Lemma.

Lemma 4.5 *Let $\mathbf{a} \in \mathbb{R}^N$, $\mathbf{b} \in \mathbb{R}^M$ with $M \geq N \geq 1$, $a_0, b_0 \neq 0$ and $\mathbf{T}_N$, $\mathbf{T}_M$, $\mathbf{T}_{N+M-1}$ be flip matrices of order N, M, $N+M-1$ respectively, then*

$$\mathbf{T}_{M+N-1}\mathbf{c} = \mathbf{T}_N\mathbf{a} * \mathbf{T}_M\mathbf{b} \tag{4.42}$$

holds.

Proof. From linear algebra we know, that the order of the polynomial $P_c = P_a P_b$ for non-vanishing leading coefficients a_0 and b_0 equals $N + M - 1$. We start with the equation $\mathbf{c} = \mathbf{a} * \mathbf{b}$. With the notation from equation (4.41) we have

$$\mathbf{c} = \mathbf{B}\mathbf{a} \tag{4.43}$$

We now multiply equation (4.43) with the flip-matrix $\mathbf{T}_{N+M-1}$ from the left hand side and use, that every flip-matrix is selfinverse

$$\mathbf{T}_{M+N-1}\mathbf{c} = \mathbf{T}_{M+N-1}\mathbf{B}\mathbf{a} = \mathbf{T}_{M+N-1}\mathbf{B}\mathbf{I}\mathbf{a} = \mathbf{T}_{M+N-1}\mathbf{B}\mathbf{F}_N\mathbf{T}_N\mathbf{a}.$$

We now consider the matrix $\mathbf{T}_{M+N-1}\mathbf{B}\mathbf{T}_N$. As we observed before, the left-multiplication of any matrix $\mathbf{B}$ by a flip matrix flips the rows of $\mathbf{B}$ and the right-multiplication flips the columns of $\mathbf{B}$. Thus we get

$$\mathbf{T}_{M+N-1}\mathbf{B}\mathbf{T}_N = \begin{pmatrix} b_{M-1} & 0 & \cdots & 0 \\ b_{M-2} & b_{M-1} & \cdots & 0 \\ \vdots & & & \vdots \\ b_0 & b_1 & \cdots & b_{M-1} \\ 0 & b_0 & \cdots & b_{M-2} \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & b_0 \end{pmatrix} \in \mathbb{R}^{(M+N-1)\times N}.$$

With entry-comparison we see that this is exactly the convolution matrix of the convolution $\mathbf{T}_M\mathbf{b} * \mathbf{a}$ for any $\mathbf{a} \in \mathbb{R}^N$. Using the commutativity of the convolution, i.e. $\mathbf{T}_M\mathbf{b} * \mathbf{T}_N\mathbf{a} = \mathbf{T}_N\mathbf{a} * \mathbf{T}_M\mathbf{b}$, the proposition holds. ■

Lemma 4.5 shows, that it does not matter, whether we first flip the coefficient-vectors $\mathbf{a}, \mathbf{b}$ and then evaluate their convolution or flip the result of the convolution of $\mathbf{a}$ and $\mathbf{b}$. This can be used to formulate the following Lemma.

Lemma 4.6 *Let P_a, P_b be two symmetric polynomials of order $N-1, M-1$ respectively, i.e. $\mathbf{a} \in \mathbb{R}^N$, $\mathbf{b} \in \mathbb{R}^M$. Let w.l.o.g. $M \geq N$, then the product $P_c = P_a P_b$ is itself a symmetric polynomial.*

Proof. If $M < N$ we can rename the polynomials due to the commutative multiplication of polynomials. As we said before, the coefficients of the product of two polynomials P_a, P_b can be evaluated via the convolution $\mathbf{c} = \mathbf{a} * \mathbf{b}$ of the corresponding coefficient-vectors $\mathbf{a}, \mathbf{b}$. From Lemma 4.5 we know that

$$\mathbf{T}_{M+N-1}\mathbf{c} = \mathbf{T}_{M+N-1}\mathbf{a} * \mathbf{b} = \mathbf{T}_N\mathbf{a} * \mathbf{T}_M\mathbf{b} = \mathbf{a} * \mathbf{b}. \tag{4.44}$$

holds. Due to the symmetry of the coefficient-vectors $\mathbf{a}, \mathbf{b}$ the last equal-sign in equation (4.44) holds. ■

Let us formulate a Corollary.

Corollary 4.7 *The product of any number of polynomials of the type $(x-z)(x-\bar{z})$ with $z = \mathrm{e}^{\mathrm{i}\omega}$ $\omega \in [0, \pi]$ is symmetric.*

Proof. It is easy to evaluate that $(x-z)(x-\bar{z}) = x^2 - 2ax + 1$ for the representation $z = a + \mathrm{i}b$ of the complex number z. Thus the coefficient-vector of this polynomial reads as $(1, -2a, 1)$ and is symmetric. Using Lemma 4.6 the proposition holds. ■

We apply these results to Prony's method for sparse linear combinations of exponentials, when only an upper bound L to the sparsity M of f is known.

By definition (4.38) of the real valued input signal $f(x)$ we know that for every frequency T_k in f there is another frequency $T_{-k} = -T_k$ in f. Thus, the Prony-polynomial P_p of order $2M+1$ with roots $\lambda_j = \mathrm{e}^{T_j}$, $j = -M, \ldots, M$ and coefficient vector $\mathbf{p} \in \mathbb{R}^{2M+2}$ is a symmetric polynomial. And therefore, we can represent the polynomial P_q that we construct out of the found eigenvector $\mathbf{q} \in \mathbb{R}^N$ in algorithm 2.1 (singular vector in algorithm 2.2) as $P_q = P_p P_r$. Hereby N determines now the number $N+1$ of input values $f(k)$, $k = 0, \ldots, N$ and P_r is some residual polynomial of degree $N - (2M+1)$. From Corollary 4.7 we know that P_p is a symmetric polynomial. If we multiply the coefficient-vector $\mathbf{q}$ of P_q by $\mathbf{T}_{N+1}$ from the left and apply Lemma 4.5 we get

$$\mathbf{T}_{N+1}\mathbf{q} = \mathbf{T}_{2M+2}\mathbf{p} * \mathbf{T}_{N-2M}\mathbf{r} = \mathbf{p} * \mathbf{T}_{N-2M}\mathbf{r}.$$

Hereby the last equality holds because of the symmetry of P_p. So, if we flip the coefficients of the polynomial P_q we get a polynomial which still has the desired zeroes that correspond to the frequencies of f. With this step we can compare the two zero-sets and delete all zeroes which have a distance greater than some ε of any zero of the un-flipped polynomial P_q, because those zeroes surely do not belong to the Prony-polynomial P_p.

This procedure can be helpful to distinguish between the desired zeroes and the unrequested ones. Let $P_{T_{N+1}q}$ be the polynomial associated with the vector $\mathbf{T}_{N+1}\mathbf{q}$, i.e. it is the flipped version of P_q. We saw that both $P_{T_{N+1}q}$ and P_q share the same roots as the Prony-polynomial P_p in the absence of noise. Thus also the polynomial $Q = (1-\alpha)P_q + \alpha P_{T_{N+1}q}$ is a polynomial that has the same roots as P_p. Since the roots of a polynomial are continuously dependent on the coefficients of that polynomial, see e.g. theorem of Bauer-Fike [24] p.200, we can assume that the roots of P_p differ only slightly from the corresponding roots of $\tilde{P}_{T_{N+1}q}$ and $\tilde{P}_q$ if small perturbations in the coefficients of $P_{T_{N+1}q}$ resp. P_q are present. In other words, the common roots of the set of polynomials $P_{q,\alpha} = \{(1-\alpha)P_{T_{N+1}q} + \alpha P_q | \alpha \in \mathbb{R}\}$ should stay in a small neighborhood around the actual roots of P_p independently of α, whereas the remaining roots can vary a lot.

Example 4.8 *Consider the function*

$$f(x) = \sin(1.2x) - 2\cos(0.3x),$$

with given sampling points $f(k)$, $k = 0, \ldots, 40$. Here we use the point evaluation functional $F(f) := f(0)$ and the matrix $\mathbf{H}_{21} = (f(k+m))_{k,m=0}^{20}$ in algorithm 2.2. This leads to a polynomial P_q of order 20, whereas the sparsity M equals 4. Thus we have to discard 16 roots of P_q. In Figure 5 we see the unit circle in blue, the correct locations of the four roots of the Prony-polynomial as red stars and the black dots show the location of the roots of the polynomial $(1-\alpha)P_q + \alpha P_{T_{21}q}$, when we let α go from -2 to 3 in steps of $1/100$. As expected, we see that the common roots stay in a close neighborhood for all α, whereas the other roots differ vastly. Note that only one polynomial P_q is necessary to construct the set of polynomials $P_{q,\alpha} = \{(1-\alpha)P_{T_{21}q} + \alpha P_q | \alpha \in \mathbb{R}\}$.

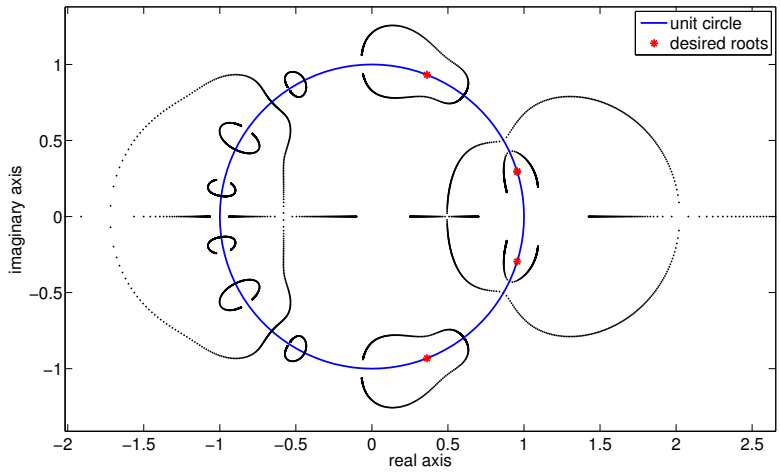

Figure 5: The blue line indicates the unit circle, the red stars show the location of the roots of the Prony-polynomial P_p and the black dots show the locations of the roots of $(1-\alpha)P_q + \alpha P_{T_{21}q}$ for $\alpha = -2, -2+1/100, \ldots, 3$.

Remark 4.9 *1. There are other operators that also possess eigenfunctions of the form* e^{Tx}*, see [34]. For example, the shift operator* S_a *in the above considerations can be replaced by the difference operator* $\Delta_a f(x) = f(x+a) - f(x)$ *or even by an* m*-fold difference operator* $\Delta_a^m f := \Delta_a^{m-1}(\Delta_a f)$*,* $(m \in \mathbb{N})$*. Using again* $Ff := f(x_0)$*, the reconstruction then involves the values*

$$F(\Delta_a^{mk} f) = \Delta_a^{mk} f(x_0) = \sum_{\ell=0}^{mk} \binom{mk}{\ell} (-1)^\ell f(x_0 + \ell a).$$

2. Instead of using the functional $Ff = f(x_0)$ *for some fixed* $x_0 \in \mathbb{R}$*, one can also use a different functional. The functional* $Ff := \int_{x_0}^{x_0+a} f(x)\mathrm{d}x$ *leads to a reconstruction method, where* f *in* (4.36) *can be reconstructed from the values*

$$\begin{aligned}\int_{x_0}^{x_0+a} S_a^k f(x)\,\mathrm{d}x &= \int_{x_0}^{x_0+a} f(x+ka)\,\mathrm{d}x \\ &= \int_{x_0+ka}^{x_0+(k+1)a} f(x)\,\mathrm{d}x, \qquad k = 0, \ldots, 2M-1.\end{aligned}$$

We will have a closer look to the impact of the functional F *to our generalized Prony Method in subchapter 5.1.*

3. The reconstruction method also applies to the multivariate case, as demonstrated in subchapter 5.2.

Moreover, Theorem 1.4 also admits efficient recovery of sums of the form

$$f(x) = \sum_{j=1}^{M} \sum_{\ell=0}^{r} c_{j,\ell} w_\ell(x) \mathrm{e}^{T_j x} \tag{4.45}$$

with $r \geq 0$, where w_ℓ, $\ell = 0, \ldots, r$ denote algebraic polynomials of exact degree ℓ. Indeed we easily observe, that $w_\ell(x)\mathrm{e}^{Tx}$, $\ell = 0, \ldots, r$ are linearly independent generalized eigenfunctions of multiplicity $\ell + 1$ of the shift operator S_a to the eigenvalue e^{Ta}. Therefore, we can apply Theorem 1.4 for the reconstruction of f using the $2M(r+1)$ values $F(\mathcal{A}^k f)$, $k = 0, \ldots, 2M(r+1)-1$. Let us look at an example.

Example 4.10 *Consider the real valued signal*

$$f(x) = (-x^2 + 30x - 56)\mathrm{e}^{-0.4x} + (\sin(x) - 2\sin(x/10))\mathrm{e}^{x/5},$$

where the unknown function $\mathrm{e}^{-0.4x}$ *is multiplicatively combined with a second order polynomial and the sum of sines* $(\mathrm{e}^{\mathrm{i}x} - \mathrm{e}^{-\mathrm{i}x})/(2\mathrm{i}) + \mathrm{i}(\mathrm{e}^{-\mathrm{i}x/10} - \mathrm{e}^{\mathrm{i}x/10})$ *is damped by* $\mathrm{e}^{x/5}$*. For simplicity we choose again the point evaluation functional* $F(f) := f(0)$ *and use the input data* $f(k)$, $k = 0, \ldots, 32$*. In figure 6 the function* f *is depicted in blue, the red stars indicate the* 33 *function values.*

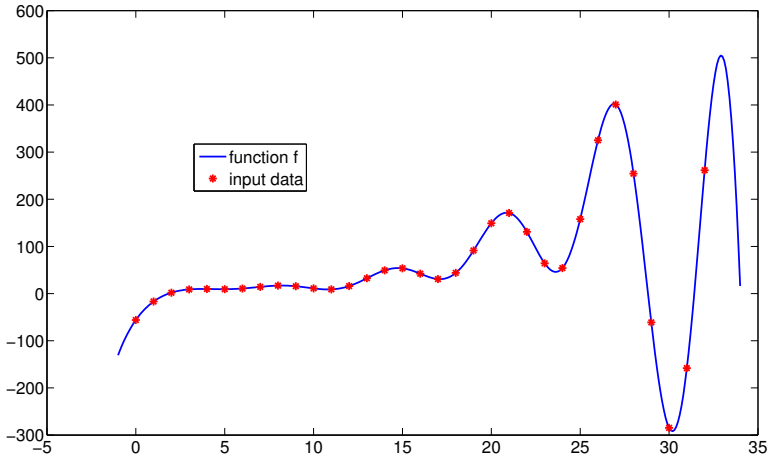

Figure 6: The function $(-x^2 + 30x - 56)\mathrm{e}^{-0.4x} + (\sin(x) - 2\sin(x/10))\mathrm{e}^{x/5}$ in blue and 33 sampling points as red stars.

Theorem 1.4 tells us that the Prony-polynomial has to have a triple root at $\mathrm{e}^{-0.4}$*. Furthermore we expect four roots with real part* 0.2 *and the imaginary parts* $\mathrm{i}, -\mathrm{i}, \mathrm{i}/10, -\mathrm{i}/10$*. Indeed we calculate (among other roots)*

j	$\ln \tilde{\lambda}_j$
1	-0.4000169
2	$-0.3999915 + 0.0000146\mathrm{i}$
3	$-0.3999915 - 0.0000146\mathrm{i}$
4	$0.2000000000000006 + 0.9999999999999996\mathrm{i}$
5	$0.2000000000000006 - 0.9999999999999996\mathrm{i}$
6	$0.1999999999999982 + 0.0999999999999991\mathrm{i}$
7	$0.1999999999999982 - 0.0999999999999991\mathrm{i}$

Table 5: The found roots of the damped sines and a triple root of the Prony-related-polynomial $P(q)$.

We find three roots located in a small neighborhood around $\mathrm{e}^{-0.4}$ *and if we sum them up we get* $\mathrm{e}^{-0.4000000000002}$*, which is in the range of the accuracy of the other desired roots. Nevertheless we observe, that the evaluation of multiple roots is numerically challenging. Note that in the case of present generalized eigenfunctions we first have to classify the clustered roots, calculate the mean* $\lambda_\varnothing$ *and make the ansatz* $\sum_{\ell=0}^{r} x^\ell \mathrm{e}^{\lambda_\varnothing}$ *in the construction of the Vandermonde-type-matrix* $\mathbf{V}$ *in order to calculate the coefficients* c_j*. We set* $\xi = -0.4000000000002$*, construct the matrix*

$$\mathbf{V} = \left(\mathrm{e}^{\xi}, k\mathrm{e}^{\xi}, k^2\mathrm{e}^{\xi}, \mathrm{e}^{\lambda_4 k}, \ldots, \mathrm{e}^{\lambda_{16} k}\right)_{k=0}^{32}$$

and solve the system $\mathbf{V}\tilde{\mathbf{c}} = \mathbf{f}$*, with* $\mathbf{f} = (f(k))_{k=0}^{32}$ *for the vector of coefficients* $\tilde{\mathbf{c}} = (\tilde{c}_j)_{j=1}^{16}$*. Now we discard all coefficients and corresponding roots which absolute value falls below a suitable threshold* δ*. Here we choose* $\delta = 10^{-10}$*. The remaining pairs* $\tilde{c}_j, \lambda_j$ *are the found coefficients and corresponding eigenvalues. We list the coefficients in Table 6.*

j	$\tilde{c}_j$	c_j
1	-0.99999999999489	-1
2	29.99999999999123	30
3	-56.00000000000098	-56
4	$1.00000000000003\mathrm{i}$	1
5	$-1.00000000000003\mathrm{i}$	-1
6	$0.49999999999999\mathrm{i}$	$1/2$
7	$-0.49999999999999\mathrm{i}$	$-1/2$

Table 6: The computed coefficients of the signal f.

Remark 4.11 *The standard differential and integral operators also posses exponential functions as eigenfunctions. We will investigate those operators in subchapter 4.5.*

4.2 Prony Method for Eigenfunctions of the Dilatation Operator

Let us consider the vector space $C(\mathbb{C})$ of continuous functions and let $D_a : C(\mathbb{C}) \to C(\mathbb{C})$ with

$$D_a f(x) := f(ax), \quad a \in \mathbb{C}, |a| \neq 1$$

be the dilatation operator on $C(\mathbb{R})$. By

$$D_a x^p = (ax)^p = a^p x^p$$

we observe that $\{x^p : p \in \mathbb{C}\}$ is a set of eigenfunctions to D_a with corresponding eigenvalues a^p. In order to ensure that the eigenvalues a^p are pairwise distinct, we assume that $\operatorname{Im} p \in [-\frac{\pi}{\ln a}, \frac{\pi}{\ln a})$. Therefore, we consider only the "admissible" set of eigenfunctions $\{x^p : p \in \mathbb{C}, \operatorname{Im} p \in [-\frac{\pi}{\ln a}, \frac{\pi}{\ln a})\}$. Further, let the functional $F : C(\mathbb{C}) \to \mathbb{C}$ be given, e.g. $F(f) := f(x_0)$, where $x_0 \in \mathbb{C}\backslash\{0\}$ is arbitrary but fixed.

With Theorem 1.2, we can uniquely reconstruct the sparse sum of generalized monomials

$$f(x) = \sum_{j=1}^{M} c_j x^{p_j} \tag{4.46}$$

with $c_j \in \mathbb{C}\backslash\{0\}$ and pairwise different $p_j \in \mathbb{C}$ which satisfy $\operatorname{Im} p_j \in [-\frac{\pi}{\ln a}, \frac{\pi}{\ln a})$, using the $2M$ values $F(D_a^k f)$, $k = 0, \ldots, 2M-1$. Furthermore, in practice a needs to be taken, such that $a^k x$, $k = 0, \ldots, 2N$ are pairwise different values in order to circumvent redundant input data. This means, once the number $2N+1$ of input data is fixed we can relax the condition $|a| \neq 1$ in the definition of the dilatation operator to $a^k \neq a^\ell$ for all $k, \ell = 0, \ldots, 2N+1$. Let $a = \alpha \mathrm{e}^{\beta \mathrm{i}}$. The sampling points are lying on a decreasing spiral for $|\alpha| < 1$, on an increasing spiral for $|\alpha| > 1$ and on the unit circle for $\alpha = 1$. For $\alpha = 1$ the value β has to be chosen in such a way, that $a^\ell x \neq a^j x$ for all $\ell, j \in \{0, \ldots, 2N\}$.

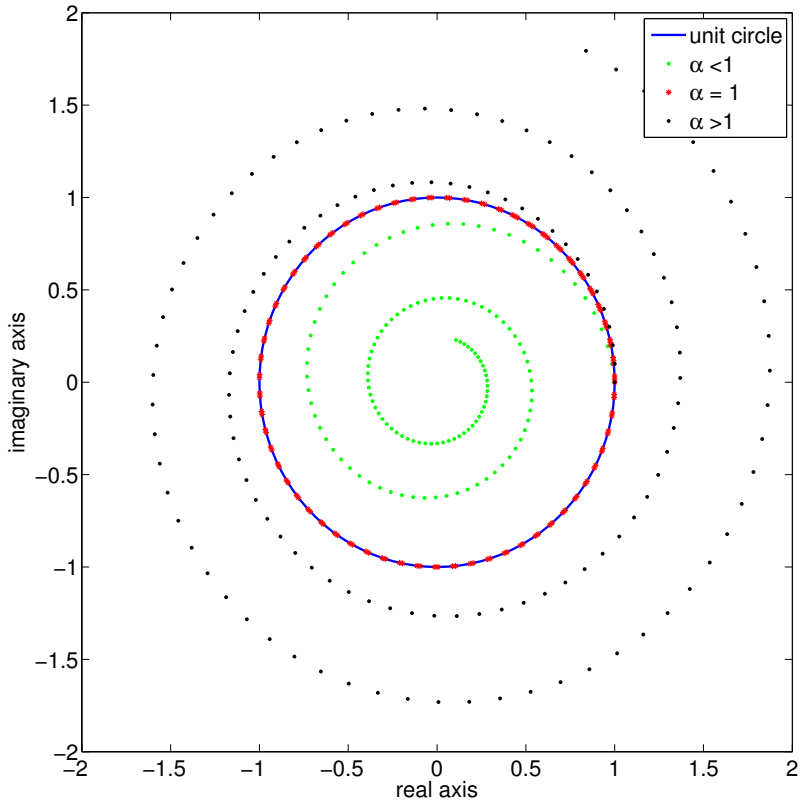

Figure 7: As an example, the needed sets of sampling points are color coded for different values of α.

Although the sets of sampling values might be somewhat unusual, the benefit is that we can use a Prony Method to analyze this function. And this gives the great advantage of needing only $2M$ input data to determine the $2M$ unknowns c_j, n_j, $j = 1, \ldots, M$. In other words, the number of input data is only dependent on the sparsity of the underlying function and not on the order of the generalized polynomial f, i.e. a polynomial of arbitrary dimension is uniquely determined by only $2M$ function values, as long as it is M-sparse in the monomial basis. Let demonstrate this method in an algorithm.

Algorithm 4.12 (Reconstruction of f in (4.46))

Input: Dilatation operator with parameter h satisfying $h^k \neq h^m$, $k, m \in \{0, \ldots, 2N-1\}$ as well as the data $F(f)(xh^k)$, $k = 0, \ldots, 2N-1$. Optionally an upper bound L for the sparsity M, else $L = N$.

1. (a) Construct the Hankel matrix

$$\mathbf{H}_{2N-L,L+1} = (F(f)(xh^{k+m}))_{k,m=0}^{2N-L-1,L}$$

and use the ESPRIT Method 3.1.2, the Matrix Pencil Method 3.2.1 or use 2.1 respectively 2.2 with an svd approach to find the eigenvalues $\lambda_j = h^{p_j}$, $j = 1, \ldots, M$ of the active eigenfunctions x^{p_j}.

Alternatively:

(b) Construct the Hankel matrix $\mathbf{H}_{N-1} = (F(f)(xh^{k+m}))_{k,m=0}^{N-1}$ and use 2.1 or 2.2 with an eigenvalue decomposition approach to find the eigenvalues $\lambda_j = h^{p_j}$, $j = 1, \ldots, M$ of the active eigenfunctions x^{p_j}.

2. Compute the coefficients c_j by solving the overdetermined system

$$F(f)(xh^k) = \sum_{j=1}^{M} c_j F(x^{p_j} h^{p_j k}) \qquad k = 0, \ldots, 2N-1.$$

Output: M, c_j, p_j, $j = 1, \ldots, M$.

Example 4.13 *Let us look at an example with the underlying function*

$$f(x) = -x^{30} + 1.3x^{18} - 2x^9 + 6x^5. \tag{4.47}$$

We choose $a = \mathrm{e}^{\mathrm{i}/10}$ where $\alpha = 1, \beta = 0.1$ and use $f(a^k)$, $k = 0, \ldots, 10$ as input data, where $F(f) = f(x_0)$ is the functional that evaluates the function f at the point x_0 with $x_0 = 1$. Thus, we sample the function (4.47) on the unit circle, as indicated by the green dots in Figure (8). Since we sample the polynomial f in the complex plane we obtain also complex function values. The unit circle is represented by the blue continuous line and the green dots mark the sampling points a^k, $k = 0, \ldots, 10$. The black line shows how the function f behaves on that desired part of the unit circle and the red dots indicate the input data, i.e. the function values at the sampling points. The blue dashed line shows which function value belongs to which sampling point.

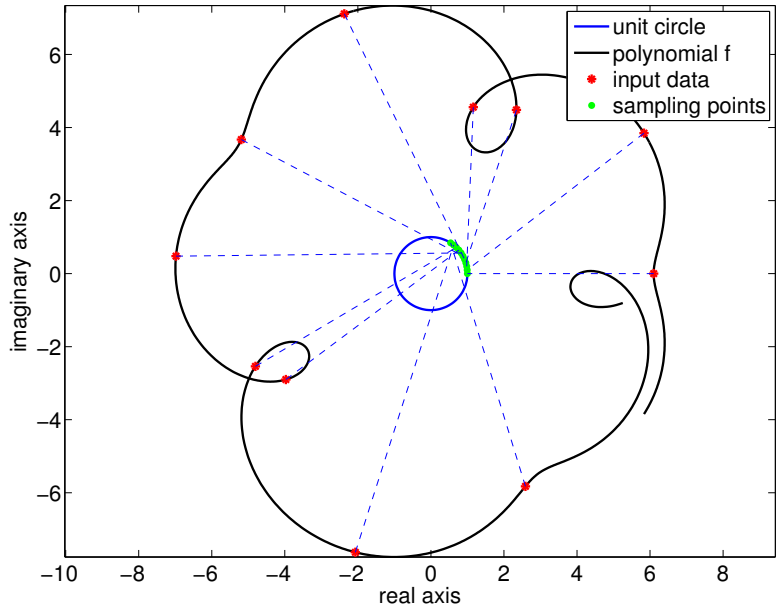

Figure 8: Illustration of the complex valued polynomial in (4.47).

Note that even though we oversampled the function slightly and that no upper bound L to the sparsity $M = 4$ is fixed, we still have very few input data. For that reason we chose algorithm 2.2 with a square Hankel matrix $\mathbf{H} = (F(D_a^{\ell+s} f)(1))_{\ell,s=0}^{5}$.

The output $\tilde{n}_j, \tilde{c}_j$, $j = 1, \ldots, 4$ for the parameters $n_1, \ldots, n_4$ and the corresponding coefficients $c_1, \ldots, c_4$ is shown in Table 7.

j	c_j	n_j	$\tilde{c}_j$	$\tilde{n}_j$
1	6	5	5.999999999999998	4.99999999999996
2	−2	9	−2.000000000000001	8.99999999999757
3	1.3	18	1.300000000000000	17.99999999999967
4	−1	30	−0.999999999999999	29.99999999999999

Table 7: We observe accurate reconstruction of the polynomial degrees n_j and the corresponding coefficients c_j.

Since we know that the degrees n_j are integers we round the obtained degrees $\tilde{n}_j$ prior to solving the linear system in order to compute the corresponding coefficients c_j. For that reason the coefficients $\tilde{c}_j$ are computed more accurately, although they technically rely on the accuracy of $\tilde{n}_j$.

Of course this procedure does not apply if we do not have this information about the parameters n_j in advance. Indeed, n_j can be real or even complex

valued, and we are able to analyze suchlike functions nonetheless.

Example 4.14 *Let f be a sparse linear combination of rational functions*

$$f(x) = \frac{6}{x^9} + \frac{\sqrt{x}}{5} + 1.3\,x, \tag{4.48}$$

and let $x_0 = -0.7-0.7\mathrm{i}$, $a = 1.1\,\mathrm{e}^{\mathrm{i}/5}$ with $\alpha = 1.1, \beta = 0.2$. We can apply exactly the same algorithm to the problem of finding the parameters $n_1 = -9, n_2 = 0.5, n_3 = 1$ and the corresponding coefficients out of given sampling points. For simplicity let us again choose the point evaluation functional $F(f) := f(x_0)$ and let us oversample the function slightly, e.g. we take $f(a^k x_0)$, $k = 0, \ldots, 14$ as input data. This way we see a bit more of the function's behavior in Figure 9. Using algorithm 2.2 we get the following good result:

j	c_j	n_j	$\tilde{c}_j$	$\tilde{n}_j$
1	6	−9	6.000000000000195	−8.999999999999991
2	0.2	0.5	−1.999999999999844	0.499999999999968
3	1.3	1	1.299999999999984	1.000000000000002

In Figure 9 we see a visualization of the behavior of the function, and the sampling points in the same manner as in Figure 8.

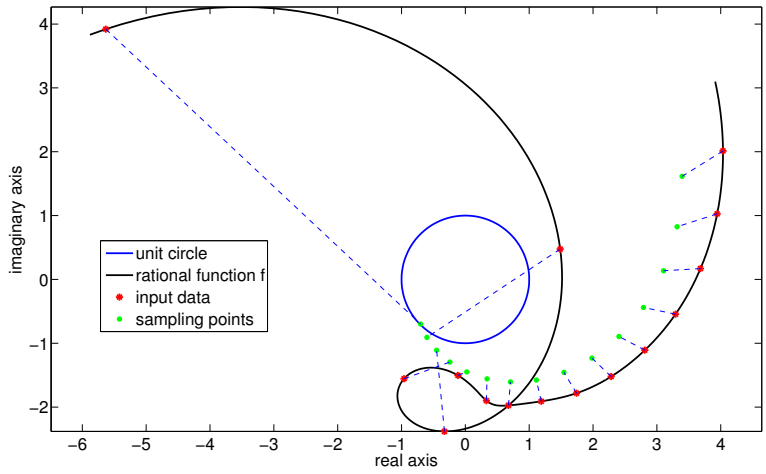

Figure 9: Illustration of the complex valued rational function in (4.48).

Remark 4.15 *In 1988, M. Ben-Or and P. Tiwari [3] introduced an algorithm for parameter identification of sparse polynomials that is a special case of the Prony algorithm for the dilatation operator explained above. Apparently, Ben-Or and Tiwari were not aware that their algorithm is a Prony-type algorithm, but this was discovered, e.g. in 2002 by M. Giesbrecht, G. Labahn and W. Lee [22].*

It should be noted, that in [3] the function f can also be a multidimensional polynomial, but the sampling points have to be very special vectors which entries are pairwise distinct prime numbers. The multidimensional case of the generalized Prony Method will be discussed in chapter 5.2.

Our generalized Prony Method applied to the dilatation operator presented above generalizes the Ben-Or and Tiwari algorithm [3] for interpolating sparse polynomials in the sense that we are not restricted to integer exponents p_j. It can be applied to multivariate sums, where some restrictions to the set of admissible eigenfunctions are needed in order to ensure an injective mapping from the eigenvalues to the eigenfunctions, see Remarks 4.9 for the similar case of multivariate exponential sums. By using a different functional F, a further generalization of the Ben-Or and Tiwari reconstruction method is possible. As before, for example $Ff := \int_0^1 f(x)\mathrm{d}x$ leads to a reconstruction method using the values

$$F(D_a^k f) = \int_0^1 f(a^k x)\,\mathrm{d}x = \frac{1}{a^k}\int_0^{a^k} f(x)\,\mathrm{d}x, \quad k = 0, \ldots, 2M-1.$$

We may admit also generalized eigenfunctions of the dilation operator $D_a : C((0,\infty)) \to C((0,\infty))$. Assuming that $a > 0$, $a \neq 1$, we observe that functions of the form $(\ln x)^\ell x^p$, $\ell = 0, \ldots, r$ are generalized eigenfunctions of multiplicity $\ell + 1$ of D_a (as defined in Chapter 1.3), since

$$\begin{aligned} D_a\left((\ln x)^\ell x^p\right) &= (\ln ax)^\ell (ax)^p = (\ln a + \ln x)^\ell a^p x^p \\ &= a^p (\ln x)^\ell x^p + \sum_{s=0}^{\ell-1} \binom{\ell}{s} (\ln a)^{\ell-s} a^p (\ln x)^s x^p. \end{aligned}$$

Thus, by Theorem 1.4, we are able to recover also expansions of the form

$$f(x) = \sum_{j=1}^{M} \sum_{\ell=0}^{r} c_{j,\ell} (\ln x)^\ell x^{p_j}$$

from the measurements $f(a^k x_0)$, $k = 0, \ldots, 2(r+1)M - 1$ for $r \geq 0$, where $x_0 > 0$ is fixed.

Remark 4.16 *Eigenfunctions of monomial form can also be obtained using suitable differential operators, as we will see in subchapter 4.5.*

4.3 Prony Method for the Sturm-Liouville Operator

Besides the exact Ben-Or/Tiwari interpolation algorithm for sparse multivariate black-box polynomials [3], there have been several attempts to modify the scheme in order to improve its stability and to reduce the computational costs, see e.g. [23], [30]. The modifications include also stochastic approaches, as e.g. the methods by Zippel [56].

However, those reconstruction ideas can not easily be transferred to other polynomial bases $(P_k)_{k=0}^{\infty}$, since they require the property

$$P_k \cdot P_l = P_{k+l}$$

(which is of course satisfied for monomials). An equivalent argument is, that other polynomials than monomials are not eigenfunctions of the dilatation operator.

In [32], Lakshman and Saunders firstly succeeded to achieve reconstruction formulas also for sparse expansions in the Pochhammer basis $(u_n)_{n=0}^{\infty}$ with $u_0(x) = 1$, $u_n(x) = x(x+1)\dots(x+n-1)$ and in the basis of Chebyshev polynomials of first kind, given by $T_0(x) = 1$, $T_1(x) = x$ and $T_n(x) = 2x\,T_{n-1}(x) - T_{n-2}(x)$ for $n \geq 2$. These generalizations rely on very special properties of these two bases, and there is no straightforward method for generalization to other bases of orthogonal polynomials.

Let us consider the Banach space $C^{\infty}(\mathbb{R})$ of infinitely differentiable functions. Let $L_{p,q} : C^{\infty}(\mathbb{R}) \to C^{\infty}(\mathbb{R})$ be the Sturm-Liouville differential operator of the form

$$L_{p,q}f(x) := p(x)f''(x) + q(x)f'(x), \quad f \in C^{\infty}(\mathbb{R}), \tag{4.49}$$

where $p(x)$ and $q(x)$ are polynomials of degree 2 and 1, respectively. It is well-known, that suitably defined orthogonal polynomials Q_n are eigenfunctions of this differential operator for special sets of eigenvalues $\lambda_n, n \in \mathbb{N}_0$, i.e., $L_{p,q}Q_n = \lambda_n Q_n$. For convenience, we list the most prominent orthogonal polynomials with their corresponding $p(x), q(x)$ and their eigenvalues λ_n, $n \in \mathbb{N}$ in Table 8.

$p(x)$	$q(x)$	λ_n	name	symbol
$(1-x^2)$	$(\beta-\alpha-(\alpha+\beta+2)x)$	$-n(n+\alpha+\beta+1)$	Jacobi	$P_n^{(\alpha,\beta)}$
$(1-x^2)$	$-(2\alpha+1)x$	$-n(n+2\alpha)$	Gegenbauer	$C_n^{(\alpha)}$
$(1-x^2)$	$-2x$	$-n(n+1)$	Legendre	P_n
$(1-x^2)$	$-x$	$-n^2$	Chebyshev 1. kind	T_n
$(1-x^2)$	$-3x$	$-n(n+2)$	Chebyshev 2. kind	U_n
1	$-2x$	$-2n$	Hermite	H_n
x	$(\alpha+1-x)$	$-n$	Laguerre	$L_n^{(\alpha)}$

Table 8: Polynomials $p(x)$ and $q(x)$ defining the Sturm-Liouville operator, corresponding eigenvalues λ_n and eigenfunctions.

Obviously, Gegenbauer, Legendre, and Chebyshev polynomials are special cases of Jacobi polynomials, where we have $C_n^{(\alpha)} := P_n^{(\alpha-1/2,\alpha-1/2)}$, $P_n := P_n^{(0,0)}$, $T_n := P_n^{(-\frac{1}{2},-\frac{1}{2})}$ and $U_n := P_n^{(\frac{1}{2},\frac{1}{2})}$.

We easily observe that for a set of eigenfunctions $\{Q_n : n \in \mathbb{N}_0\}$, the corresponding eigenvalues are pairwise different and well separated, i.e. $\lambda_n \neq \lambda_m$ for $n \neq m$. Further, we choose the functional $F : C^{\infty}(\mathbb{R}) \to \mathbb{C}$ that returns f at a fixed value $x_0 \in \mathbb{R}$, i.e., $F(f) := f(x_0)$ with the condition that $Q_n(x_0) \neq 0$ for all $n \in \mathbb{N}_0$.

Let $f(x)$ be an M-sparse expansion of orthogonal polynomials Q_n, $n \geq 0$,

$$f(x) = \sum_{j=1}^{M} c_{n_j} Q_{n_j}(x), \tag{4.50}$$

where $c_{n_j} \in \mathbb{C}\backslash\{0\}$, $0 \leq n_1 < \cdots < n_M = N$ are the indices of the "active" basis polynomials Q_{n_j} in the expansion, and $n_M = N \gg M$ is the polynomial degree of f.

Now Theorem 1.2 yields that $f(x)$ can be uniquely recovered using the values

$$F(L_{p,q}^k f) = L_{p,q}^k f(x_0) = \sum_{j=1}^{M} c_{n_j} \lambda_{n_j}^k Q_{n_j}(x_0), \quad k = 0, \ldots, 2M-1, \tag{4.51}$$

where the functional F is again chosen as $F(f) = f(x_0)$. Of course we will hardly find an application that provides us with input data of the form (4.51), but we will show that the values $L_{p,q}^k f(x_0)$, $(k = 0, \ldots, 2M-1)$ can be determined uniquely by the derivative values $f^{(m)}(x_0)$ for $m = 0, \ldots, 4M-2$, and this assertion holds not only for sparse but for all expansions of orthogonal polynomials $f(x)$.

Theorem 4.17 *Let $f \in C^\infty(\mathbb{R})$ be an arbitrary polynomial of degree $N \in \mathbb{N}$ and let $L_{p,q} : C^\infty(\mathbb{R}) \to C^\infty(\mathbb{R})$ be the Sturm-Liouville differential operator as given in* (4.49)*. Then, for each fixed $x \in \mathbb{R}$, the values $L_{p,q}^k f(x)$, $k = 0, \ldots, 2M-1$, can be determined uniquely by the derivative values $f^{(m)}(x)$, $m = 0, \ldots, 4M-2$, and we have*

$$L_{p,q}^k f(x) = \sum_{\ell=1}^{2k} g_{\ell,k}(x)\, f^{(\ell)}(x)$$

for $k \geq 1$. Here $g_{1,1}(x) = q(x)$, $g_{2,1}(x) = p(x)$, and for $k \geq 2$, $g_{\ell,k}(x)$ satisfies the recursion

$$\begin{aligned} g_{\ell,k}(x) = {} & \ell \left(\frac{\ell-1}{2} p''(x) + q'(x) \right) g_{\ell,k-1}(x) \\ & + ((\ell-1)p'(x) + q(x)) g_{\ell-1,k-1}(x) + p(x) g_{\ell-2,k-1}(x), \quad \ell = 1, \ldots, 2k, \end{aligned} \tag{4.52}$$

with the convention $g_{\ell,k}(x) = 0$ for $k \geq 1$, $\ell \notin \{1, \ldots, 2k\}$.

Proof. 1. For $k = 0$, we observe that $L_{p,q}^0 f(x) = f(x)$, i.e. $L_{p,q}^0$ is the identity operator. Since the operator $L_{p,q}$ is a differential operator of order 2, we can use the ansatz

$$L_{p,q}^k f(x) = \sum_{\ell=1}^{2k} g_{\ell,k}(x) f^{(\ell)}(x), \quad k \geq 1, \tag{4.53}$$

with polynomials $g_{\ell,k}$, $k \in \mathbb{N}$, $\ell = 1, \ldots, 2k$. In particular, for $k = 1$ we have

$$L_{p,q} f(x) = p(x) f''(x) + q(x) f'(x),$$

i.e., $g_{1,1}(x) = q(x)$ and $g_{2,1}(x) = p(x)$. We now prove by induction on k that the coefficients $g_{\ell,k}(x)$ in (4.53) satisfy the recursion (4.52) for $k \geq 2$ and $\ell = 1, \ldots, 2k$. Using (4.53) and the general Leibniz rule, we find for all $n \in \mathbb{N}_0$,

$$\begin{aligned} L_{p,q}^k f(x) &= L_{p,q}^{k-1}(L_{p,q}f)(x) = \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x)\,[L_{p,q}f(x)]^{(\ell)} \\ &= \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x)[p(x)f''(x) + q(x)f'(x)]^{(\ell)} \\ &= \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x)[p(x)f^{(\ell+2)}(x) + \ell p'(x)f^{(\ell+1)}(x) \\ &\quad + \binom{\ell}{2} p''(x)f^{(\ell)}(x) + q(x)f^{(\ell+1)}(x) + \ell q'(x)f^{(\ell)}(x)] \\ &= \sum_{\ell=1}^{2k} g_{\ell,k}(x)f^{(\ell)}(x) \end{aligned}$$

due to the vanishing higher derivatives of $p(x), q(x)$. A comparison of coefficients leads to the recursion formulas for $g_{\ell,k}$ in (4.52). ■

Corollary 4.18 *If $x_0 \in \mathbb{R}$ is a zero of the polynomial $p(x)$ in the definition (4.49) of the Sturm-Liouville operator, i.e., if $p(x_0) = 0$, then the values $L_{p,q}^k f(x_0)$, $k = 0, \ldots, 2M-1$ can be determined by $f^{(m)}(x_0)$, $m = 0, \ldots, 2M-1$ only. More precisely, we have*

$$L_{p,q}^k f(x_0) = \sum_{\ell=1}^{k} g_{\ell,k}(x_0) f^{(\ell)}(x_0)$$

with $g_{1,1}(x_0) = q(x_0)$ and

$$\begin{aligned} g_{\ell,k}(x_0) = \ell &\left(\frac{\ell-1}{2} p''(x_0) + q'(x_0)\right) g_{\ell,k-1}(x_0) \\ &+ ((\ell-1)p'(x_0) + q(x_0)) g_{\ell-1,k-1}(x_0) \end{aligned}$$

for $k \geq 2$, $\ell \in \{1, \ldots, k\}$, where we assume that $g_{\ell,k}(x_0) = 0$ for $k \geq 1$, $\ell \notin \{1, \ldots, k\}$. In particular, for the Sturm-Liouville operator for Jacobi polynomials with $p(x) = (1-x^2)$, we need only the values $f^{(m)}(1)$ (respectively $f^{(m)}(-1)$), $m = 0, \ldots, 2M-1$, in order to reconstruct $L_{p,q}^k f(1)$, (respectively $L_{p,q}^k f(-1)$), $k = 0, \ldots, 2M-1$.

Proof. Let x_0 be such that $p(x_0) = 0$.

$$\begin{aligned}
L_{p,q}^k f(x_0) &= L_{p,q}^{k-1}(L_{p,q}f)(x_0) = \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x_0)\,[L_{p,q}f(x_0)]^{(\ell)} \\
&= \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x_0)[p(x_0)f''(x_0) + q(x_0)f'(x_0)]^{(\ell)} \\
&= \sum_{\ell=1}^{2k-2} g_{\ell,k-1}(x_0)[\underbrace{p(x_0)}_{=0} f^{(\ell+2)}(x_0) + \ell p'(x_0)f^{(\ell+1)}(x_0) \\
&\quad + \binom{\ell}{2} p''(x_0)f^{(\ell)}(x_0) + q(x_0)f^{(\ell+1)}(x_0) + \ell q'(x_0)f^{(\ell)}(x_0)] \\
&= \sum_{\ell=1}^{2k} g_{\ell,k}(x_0)f^{(\ell)}(x_0)
\end{aligned}$$

Note that we cannot cancel $p(x_0)$ after the second equal sign prior to applying the ℓ-th derivative, because $p(x_0) = 0$ does generally not imply that $p'(x_0) = 0$ or $p''(x_0) = 0$. Thus we would erase the effect of $p'(x_0)$ and $p''(x_0)$. ■

We summarize the algorithm for reconstructing orthogonal polynomial expansions as follows.

Algorithm 4.19 (Reconstruction of f in (4.50))

Input: Sturm Liouville operator with $p(x), q(x)$ and λ_n as well as the basis $\{Q_n : n \in \mathbb{N}_0\}$, $x_0 \in \mathbb{R}$, $f^{(m)}(x_0)$, $m = 0, \ldots, 4N-2$, optionally an upper bound $L_u \geq M$.
Preprocessing: Construct $\mathbf{G} = (g_{\ell,k})_{k,\ell=1}^{2N-1,4N-2} \in \mathbb{R}^{(2N-1)\times(4N-2)}$ with $g_{1,1} := q(x_0)$, $g_{2,1} := p(x_0)$, $g_{\ell,1} := 0$ for $\ell \notin \{1,2\}$, and

$$g_{\ell,k} := \begin{cases} \ell\left(\frac{\ell-1}{2}p''(x_0) + q'(x_0)\right) g_{\ell,k-1} & \\ +((\ell-1)p'(x_0) + q(x_0))g_{\ell-1,k-1} + p(x_0)g_{\ell-2,k-1}, & k > 1,\ \ell \in \{1,\ldots,2k\} \\ 0, & k > 1,\ \ell \notin \{1,\ldots,2k\} \end{cases}$$

Observe that the construction of $\mathbf{G}$ only depends on the operator $L_{p,q}$ and not on the given data $f^{(m)}(x_0)$.

1. Calculate $\mathbf{h}_1 := \mathbf{G}\mathbf{f}_1$, where $\mathbf{f}_1 := (f^{(m)}(x_0))_{m=1}^{4N-2}$. Put now

$$\mathbf{h} := \begin{pmatrix} f(x_0) \\ \mathbf{h_1} \end{pmatrix}$$

   such that $\mathbf{h} = (h_\ell)_{\ell=0}^{2N-1} = (L_{p,q}^\ell f(x_0))_{\ell=0}^{2N-1} \in \mathbb{C}^{2N}$.

2. (a) Construct the Hankel matrix $\mathbf{H}_{2N-L,L+1} = (h_{k+m})_{k,m=0}^{2N-L-1,L}$ and use the ESPRIT Method 3.1.2, the Matrix Pencil Method 3.2.1 or use

2.1 respectively 2.2 with an svd approach to find the eigenvalues λ_{n_j}, $j = 1, \ldots, M$ of the active eigenfunctions Q_{n_j}.

Alternatively:

(b) Construct the Hankel matrix $\mathbf{H}_{N-1} = (h_{k+m})_{k,m=0}^{N-1}$ and use 2.1 or 2.2 with an eigenvalue decomposition approach to find the eigenvalues λ_{n_j}, $j = 1, \ldots, M$ of the active eigenfunctions Q_{n_j}.

3. Compute the coefficients c_{n_j} by solving the overdetermined system

$$\sum_{j=1}^{M} c_{n_j} Q_{n_j}^{(\ell)}(x_0) = f^{(\ell)}(x_0), \quad \ell = 0, \ldots, 2N-1.$$

Output: M, c_{n_j}, Q_{n_j}, $j = 1, \ldots, M$.

Example 4.20 Sparse Laguerre expansions

The Laguerre polynomials with parameter α are solutions of the second order differential equation

$$x(L_n^{(\alpha)})''(x) + (\alpha + 1 - x)(L_n^{(\alpha)})'(x) = -n\, L_n^{(\alpha)}(x),$$

with eigenvalues $\lambda_n = -n$. Using Theorems 1.2 and 4.17 a sparse Laguerre expansion of the form

$$f(x) = \sum_{j=1}^{M} c_{n_j} L_{n_j}^{(\alpha)}(x)$$

with $c_{n_j} \in \mathbb{C}\backslash\{0\}$ and active indices $0 \le n_1 < \cdots < n_M = N$ can be reconstructed from $f^{(m)}(x_0)$, $m = 0, \ldots, 4M-2$. Here, x_0 has to satisfy $L_n^{(\alpha)}(x_0) \neq 0$ for all $n \in \mathbb{N}_0$. If we choose $x_0 = 0$, formula (4.52) simplifies to

$$\begin{aligned} &g_{1,1}(0) = \alpha + 1, \\ &g_{\ell,k}(0) = (\ell + \alpha) g_{\ell-1,k-1}(0) - \ell g_{\ell,k-1}(0), && k > 1, \ell = 1, \ldots, k, \\ &g_{\ell,k}(0) = 0, && k \ge 1, \ell \notin \{1, \ldots, k\}. \end{aligned}$$

For example, for $M = 2$, this leads to the triangular matrix

$$\mathbf{G} = \begin{pmatrix} (1+\alpha) & 0 & 0 \\ -(1+\alpha) & (1+\alpha)(2+\alpha) & 0 \\ (1+\alpha) & -3(1+\alpha)(2+\alpha) & (1+\alpha)(2+\alpha)(3+\alpha) \end{pmatrix}$$

in the preprocessing step of Algorithm 4.19. Let us give a small numerical example. For $\alpha = 0$ and given values $f(0), f'(0), \ldots, f^{(11)}(0)$ of the function

$$f(x) = \sum_{j=1}^{6} c_{n_j} L_{n_j}(x),$$

we use Algorithm 4.19 to calculate approximations $\tilde{n}_j, \tilde{c}_{n_j}$ of the original parameters n_j, c_{n_j} for $j = 1, \ldots, 6$, as shown in Table 9.

j	n_j	c_{n_j}	$\tilde{n}_j$	$\tilde{c}_{n_j}$
1	142	-3	142.0000000018223	-2.999999999999987
2	125	-1	125.0000000494359	-1.000000000000034
3	91	2	90.9999998114290	2.000000000000063
4	69	-3	69.0000003316075	-3.000000000000058
5	53	-1	53.0000003445395	-0.999999999999988
6	11	2	10.9999999973030	2.000000000000004

Table 9: Numerical evaluation of indices of active basis polynomials and coefficients of a sparse Laguerre expansion using Algorithm 4.19.

Here, since we know that the orders n_j of the polynomials are integers, we have rounded the values $\tilde{n}_j$ to the next integer before proceeding with the last step of Algorithm 4.19. While the degree of the polynomial $f(x)$ is 142, the 12 function and derivative values $f^{(m)}$, $m = 0, \ldots, 11$, are sufficient for reconstruction of the sparse expansion.

Example 4.21 Sparse Legendre expansions We consider sparse Legendre expansions that we have already studied in [35]. The n-th Legendre polynomial P_n satisfies the operator equation

$$(1 - x^2)P_n''(x) - 2xP_n'(x) = -n(n+1)P_n(x). \tag{4.54}$$

Hence, a sparse Legendre expansion of the form

$$f(x) = \sum_{j=1}^{M} c_{n_j} P_{n_j}(x)$$

with $c_{n_j} \in \mathbb{C}\backslash\{0\}$ and active indices $0 \le n_1 < \cdots < n_M = N$ can be reconstructed from the values $f^{(m)}(x_0)$, $m = 0, \ldots, 4M-2$, for arbitrarily chosen $x_0 \in \mathbb{R}$ satisfying $P_n(x_0) \neq 0$ for all $n \in \mathbb{N}_0$. In particular, for $x_0 = 1$ (or $x_0 = -1$) we need only the values from $f^{(m)}(1)$ (resp. $f^{(m)}(1)$), $m = 0, \ldots, 2M-1$, for the unique reconstruction of f. Multiplying (4.54) with a constant $\alpha \neq 0$ does not change the solutions. Thus we can consider

$$L^k_{p,q,\alpha} P_n(x) := \alpha(1 - x^2)P_n''(x) - 2\alpha x P_n'(x) = -n(n+1)\alpha P_n(x),$$

where $p_\alpha(x) = \alpha(1 - x^2)$, $q_\alpha(x) = -2\alpha x$ and $\lambda_{n,\alpha} = -n(n+1)\alpha$. Hence

$$L^k_{p,q,\alpha} f(1) = \sum_{\ell=1}^{k} g^\alpha_{\ell,k}(1) f^{(\ell)}(1)$$

with

$$
\begin{aligned}
&g^{\alpha}_{1,1}(1) = -2\alpha, \\
&g^{\alpha}_{\ell,k}(1) = -\ell(\ell+1)\alpha g^{\alpha}_{\ell,k-1}(1) - 2\alpha\ell g^{\alpha}_{\ell-1,k-1}(1), && k > 1, \ell = 1, \ldots, k, \\
&g^{\alpha}_{\ell,k}(1) = 0, && k \geq 1, \ell \notin \{1, \ldots, k\}.
\end{aligned}
$$

The constant α can be chosen suitably in order to improve the condition of the matrix $\mathbf{G}$. In particular, for $\alpha = -\frac{1}{2}$ we obtain

$$
g^{-1/2}_{\ell,k}(1) = \frac{\ell(\ell+1)}{2} g^{-1/2}_{\ell,k-1}(1) + \ell g^{-1/2}_{\ell-1,k-1}(1)
$$

yielding for $M = 3$,

$$
\mathbf{G} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 \\
1 & 8 & 6 & 0 & 0 \\
1 & 26 & 60 & 24 & 0 \\
1 & 80 & 438 & 480 & 120
\end{pmatrix},
$$

see also [35]. We use Algorithm 4.19 in order to recover the sparse Legendre expansion

$$
f(x) = -3P_{5492}(x) - P_{465}(x) + 2P_{54}(x)
$$

of degree 5492 from the given values $f(1)$, $f'(1), \ldots, f^{(5)}(1)$. Table 10 contains the computed approximations $\tilde{n}_j, \tilde{c}_{n_j}$ of the original parameters n_j, c_{n_j} for $j = 1, 2, 3$.

j	n_j	c_{n_j}	$\tilde{n}_j$	$\tilde{c}_{n_j}$
1	54	2	53.983951125658	2.000000000000048
2	465	−1	465.000054039331	−1.000000000000048
3	5492	−3	5491.999999999999	−3.000000000000000

Table 10: Numerical evaluation of indices of active basis polynomials and coefficients of a sparse Legendre expansion using Algorithm 4.19.

Here again, since we know that the orders n_j of the polynomials are integers, we have rounded the values $\tilde{n}_j$ to the next integer before proceeding with the last step of Algorithm 4.19.

Chebyshev reconstruction with sampling values

In the recent preprint [42], the authors consider a Prony-like method for the reconstruction of M-sparse Chebyshev expansions with Chebyshev polynomials of first kind of the form $T_n(x) = \cos n(\arccos x)$, for $x \in [-1, 1]$ and $n \in \mathbb{N}_0$. In their approach, an expansion of the form

$$
f = \sum_{j=1}^{M} c_{n_j} T_{n_j}
$$

is reconstructed from the function values $f\left(\cos(\frac{k\pi}{2N-1})\right)$, $k = 0, \ldots, 2M-1$, where $N > M$ is the true degree of the polynomial f. Obviously, this reconstruction algorithm is different from the Prony Method that is based on the Sturm-Liouville operator. Instead we have to take the so-called "Chebyshev-shift operator" $\mathcal{S}_h : C(\mathbb{R}) \to C(\mathbb{R})$ with $h \in [-1, 1]$,

$$(\mathcal{S}_h f)(x) := \frac{1}{2}\left(f(hx - (1-h^2)^{1/2}(1-x^2)^{1/2}) + f(hx + (1-h^2)^{1/2}(1-x^2)^{1/2})\right).$$

We consider $x \in [-1, 1]$ and set $x = \cos t$ with $t \in [0, \pi]$ as well as $h = \cos\alpha$ with $\alpha \in [0, \pi]$. Then, for Chebyshev polynomials of first kind we obtain indeed

$$\begin{aligned}
\mathcal{S}_h T_n(x) &= \frac{1}{2}\left(T_n(hx - (1-h^2)^{1/2}(1-x^2)^{1/2}) + T_n(hx + (1-h^2)^{1/2}(1-x^2)^{1/2})\right) \\
&= \frac{1}{2}\left(T_n(\cos(t+\alpha)) + T_n(\cos(t-\alpha))\right) \\
&= \cos(n\alpha)\,\cos(nt) = \cos(n\alpha)\,T_n(x).
\end{aligned}$$

Hence, the Chebyshev polynomials of first kind are eigenfunctions of the Chebyshev-shift operator $\mathcal{S}_{\cos\alpha}$ with the corresponding eigenvalues $\cos(n\alpha)$. We define $F(f) := f(1)$ and set $h := \cos(\frac{\pi}{2N-1})$. Applying Theorem 1.2 yields that the sparse Chebyshev expansion can be uniquely reconstructed from the values $F(\mathcal{S}_h^k f)$, $k = 0, \ldots, 2M-1$, where

$$\begin{aligned}
F(\mathcal{S}_h^0 f) &= f(1), \\
F(\mathcal{S}_h^1 f) &= f(\cos(\frac{\pi}{2N-1})), \\
F(\mathcal{S}_h^2 f) &= \frac{1}{2}\left(f(1) + f(\cos\frac{2\pi}{2N-1})\right), \\
F(\mathcal{S}_h^3 f) &= \frac{1}{4}\left(3f(\cos\frac{\pi}{2N-1}) + f(\cos\frac{3\pi}{2N-1})\right),
\end{aligned}$$

etc. Indeed, one can simply prove by induction, that the values $F(\mathcal{S}_h^k f)$, $k = 0, \ldots, 2M-1$, can be represented as linear combinations of the function values $f\left(\cos(\frac{k\pi}{2N-1})\right)$, $k = 0, \ldots, 2M-1$.

4.4 Prony Method for Finite Dimensional Linear Operators

The generalized Prony Method considered in Chapter 1.2 can also be applied to finite dimensional vector spaces. Let $\mathbf{x} \in \mathbb{C}^N$ be M-sparse, i.e., only M components of $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ differ from zero.

We want to recover $\mathbf{x}$ from only $2M$ linear measurements $y_k = \mathbf{a}_k^{\mathrm{T}}\mathbf{x}$, $k = 0, \ldots, 2M-1$, where the vectors $\mathbf{a}_k \in \mathbb{C}^N$ need to be chosen suitably. The problem of reconstructing sparse vectors using only a small amount of measurements has been heavily studied in the research field of compressed sensing, where the

recovery algorithms are usually based on ℓ^1-minimization or greedy methods. In this regard, often a stochastic matrix $\mathbf{A} \in \mathbb{C}^{M_1 \times N}$ is used in order to recover $\mathbf{x}$ from $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{C}^{M_1}$, $M_1 \geq 2M$ with high probability.

Here we want to derive a deterministic method to recover $\mathbf{x}$, where $\mathbf{A} \in \mathbb{C}^{2M \times N}$ with rows $\mathbf{a}_k^{\mathrm{T}} \in \mathbb{C}^N$ is explicitly given and of minimal dimension. For this purpose, we use a linear operator $\mathcal{A} : \mathbb{C}^N \to \mathbb{C}^N$ that can be represented by a diagonal matrix $\mathcal{A} = \mathrm{diag}(a_0, \ldots, a_{N-1})$ with pairwise different entries a_j, $j = 0, \ldots, N-1$. Obviously, the unit vectors $\mathbf{e}_j := (\delta_{j,\ell})_{\ell=0}^{N-1}$ form a system of eigenvectors of $\mathcal{A}$ with $\mathcal{A}\mathbf{e}_j = a_j \mathbf{e}_j$ for $j = 0, \ldots, N-1$. Further we choose a linear functional $F : \mathbb{C}^N \to \mathbb{C}$ e.g. of the form $F\mathbf{x} = \mathbb{1}^{\mathrm{T}}\mathbf{x} := \sum_{j=1}^N x_j$. Hence, $F\mathbf{e}_j = \mathbb{1}^{\mathrm{T}}\mathbf{e}_j = 1 \neq 0$ holds.

Using Theorem 1.2, we can now reconstruct a sparse vector $\mathbf{x}$ of the form

$$\mathbf{x} = \sum_{j=1}^{M} c_{n_j} \mathbf{e}_{n_j}$$

with $0 \leq n_1 < \cdots < n_M \leq N-1$ from the values

$$y_k := F(\mathcal{A}^k \mathbf{x}) = \mathbb{1}^{\mathrm{T}} \cdot \mathcal{A}^k \mathbf{x} = \mathbf{a}_k^{\mathrm{T}} \mathbf{x},$$

where $\mathbf{a}_k^{\mathrm{T}} = (a_0^k, \ldots, a_{N-1}^k)$, $k = 0, \ldots, 2L-1$. If the sparsity number M is not known, but only an upper bound L we can formulate the following algorithm.

Algorithm 4.22 (Reconstruction of a sparse vector)

Input: operator $\mathrm{diag}(a_0, \ldots, a_{N-1})$, $y_k = \mathbf{a}_k^{\mathrm{T}}\mathbf{x}$, $k = 0, \ldots, 2N-1$, optionally an upper bound $L \geq M$.

1. (a) Construct the Hankel matrix

$$\mathbf{H}_{2N-L,L+1} = (y_{k+m})_{k,m=0}^{2N-L-1,L}$$

and use the ESPRIT Method 3.1.2, the Matrix Pencil Method 3.2.1 or use 2.1 respectively 2.2 with an svd approach to find the eigenvalues $\lambda_j = a_{n_j}$, $j = 1, \ldots, M$ of the active eigenvectors $\mathbf{e}_{n_j}$.
Alternatively:

   (b) Construct the Hankel matrix $\mathbf{H}_{N-1} = (y_{k+m})_{k,m=0}^{N-1}$ and use 2.1 or 2.2 with an eigenvalue decomposition approach to find the eigenvalues $\lambda_{n_j} = a_{n_j}$, $j = 1, \ldots, M$ of the active eigenvectors $\mathbf{e}_{n_j}$.

2. Compute the entries x_{n_j} by solving the overdetermined system

$$y_k = \sum_{j=1}^{M} x_{n_j} a_{n_j}^k, \quad k = 0, \ldots, 2N-1.$$

Output: M, n_j, $x_{n_j} \neq 0$, $j = 1, \ldots, M$.

To demonstrate this approach we want to present a small numerical example. Let $\mathbf{x} \in \mathbb{R}^{128}$ be a 3-sparse vector with $x_{28} = 3, x_{71} = -1, x_{99} = 4$, and let $\mathcal{A} = \operatorname{diag}(k/32)_{k=-63}^{64} \in \mathbb{R}^{128\times 128}$. For a given vector of values $\mathbf{y} = (y_k)_{k=0}^{5}$ with $y_k = \mathbb{1} \cdot \mathcal{A}^k \mathbf{x}$ we compute approximations $\tilde{n}_j$ and $\tilde{x}_{n_j}$ according to Algorithm 4.22. The results are shown in the Table 11.

n_j	x_{n_j}	$\tilde{n}_j$	$\tilde{x}_{n_j}$
28	3	27.99999999999999	3
71	−1	71.00000000000001	−1
99	4	99.00000000000000	4

Table 11: Numerical evaluation of the indices and the coefficients of a sparse vector $\mathbf{x}$ using Algorithm 4.22.

Remark 4.23 *1. In order to obtain a stable algorithm, the operator $\mathcal{A}$ may for example be chosen as*

$$\mathcal{A} = \operatorname{diag}(\omega_N^0, \omega_N^1, \ldots, \omega_N^{N-1}),$$

where $\omega_N := \mathrm{e}^{-2\pi \mathrm{i}/N}$ denotes the N-th root of unity. For this choice of $\mathcal{A}$, the vector $\mathbf{y} = (y_k)_{k=0}^{2M-1}$ of needed input values for Algorithm 4.22 is given by

$$\mathbf{y} = \mathbf{F}_{N,2M}\mathbf{x},$$

where $F_{N,2M} = (\omega_N^{k\ell})_{k,\ell=0}^{2M-1,N-1} \in \mathbb{C}^{2M\times N}$ contains the first $2M$ rows of the Fouriermatrix of order N.

2. In the above considerations, the canonical basis can be replaced by any other basis $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$ of $\mathbb{C}^N$. Choose a diagonal matrix $\mathcal{A}$ with pairwise different (complex) entries $\lambda_1, \ldots, \lambda_N$. Then the operator $\mathbf{A} := \mathbf{B}\mathcal{A}\mathbf{B}^{-1} : \mathbb{C}^N \to \mathbb{C}^N$, where $\mathbf{B} = (\mathbf{b}_1 \ldots \mathbf{b}_N) \in \mathbb{C}^{N\times N}$ contains the columns $\mathbf{b}_j$, possesses the eigenvalues $\lambda_1, \ldots, \lambda_N$ with corresponding eigenvectors $\mathbf{b}_1, \ldots, \mathbf{b}_N$ by construction. Further, we define a functional $F : \mathbb{C}^N \to \mathbb{C}$ satisfying $F\mathbf{b}_\ell \neq 0$ for $\ell = 1, \ldots, N$. We can e.g. choose $F\mathbf{x} := \mathbf{d}^T\mathbf{x}$ for all $\mathbf{x} \in \mathbf{C}^N$, where $\mathbf{d}$ is taken suitably. Hence, a sparse expansion

$$\mathbf{x} = \sum_{j=1}^{M} c_{n_j}\, \mathbf{b}_{n_j}$$

in the basis B can by Theorem 1.2 be recovered by

$$F(\mathbf{A}^k\mathbf{x}) = \mathbf{d}^T\mathbf{A}^k\mathbf{x}, \qquad k = 0, \ldots, 2M-1.$$

3. Using Theorem 1.4, we can apply the recovery procedure also when the given operator possesses eigenvalues with higher multiplicity, where also generalized eigenvectors can be incorporated.

4.4.1 Comparison with Compressed Sensing

Our generalized Prony Method applied to finite dimensional vector spaces fits the setting that the research field of compressed sensing attends to. In [4] a decent introduction to compressed sensing is given. The ideas we address here were introduced in [9].

When a given signal is compressible in some domain, we try to realize the compressing of the signal directly at the sampling process. Let the signal $\mathbf{y} \in \mathbb{C}^N$ be M-sparse in a transformed domain, i.e. $\mathbf{y} = \mathbf{\Psi}\mathbf{x}$, for some transformation matrix $\mathbf{\Psi} \in \mathbb{C}^{N\times N}$ and with a vector $\mathbf{x}$ that has only M non vanishing entries. One can think of the vector $\mathbf{y}$ as for example an image and of the matrix $\mathbf{\Psi}$ as the inverse discrete Fourier transform matrix of size $N \times N$. Then, the signal $\mathbf{y}$ is compressible in the Fourier domain, if we relax the condition of strictly M-sparse vectors $\mathbf{x}$ to essentially compressible vectors $\mathbf{x}$ that have only M entries with sufficiently large absolute value. The classical approach takes the vector $\mathbf{y}$ and applies a Fourier transformation $\mathbf{x} = \mathbf{\Psi}^{-1}\mathbf{y}$ in order to compress the image $\mathbf{y}$. In compressed sensing we ask now if we can substitute the knowledge of every single entry of the signal $\mathbf{y}$ by fewer, but more general linear measurements $\mathbf{z} = \mathbf{\Phi}\mathbf{y}$. Here, the measurement matrix $\mathbf{\Phi}$ is an element of $\mathbb{C}^{M\times L}$, $M < L \ll N$. If we can recover the signal $\mathbf{y}$ only from the measurement $\mathbf{z} \in \mathbb{C}^L$, we have achieved an L-term compression directly at the sampling process.

There are two main questions to be answered here. Firstly, we have to concern ourselves with the question, which matrix $\mathbf{\Phi}$ preserves the information contained in $\mathbf{x}$ in the sampled vector $\mathbf{z}$, such that we can recover $\mathbf{y}$. Secondly, we have to categorize an algorithm that recovers $\mathbf{y}$ from $\mathbf{z}$. Of course the choice $\mathbf{\Phi} = \mathbf{I}$ ensures that all information stored in $\mathbf{y}$ remain in $\mathbf{z}$, but it keeps all samples of $\mathbf{y}$. We are looking for a linear mapping $\mathbf{\Phi}$ with as few rows as possible to guarantee the invertibility of the sampling process. One way to accomplish this, is to look at probabilistic measures. Instead of exact recovery, we focus on the probability that a random sparse signal fails to be correctly reconstructed. If that probability approaches zero we can state that the sampling scheme is successful in recovering $\mathbf{y}$ with probability 1. Let $\mathbf{\Phi}^{(L)}$ denote the first m rows of an invertible matrix $\mathbf{\Phi}_{N\times N}$. If we use $\{\mathbf{B}^{(m)}\}_{m=0}^{N}$ as sampling matrices, it follows that the failure probability of recovering for $\mathbf{\Phi}^{(0)}$, $\mathbf{\Phi}^{(N)}$ is 1 respectively 0. As L increases, the failure probability decreases. The important observation is that the decreasing rate of failure probability is exponential with respect to $\frac{m}{N}$ [9]. Therefore, we can expect an almost zero failure already for $L \ll N$. Of course the exact rate highly depends on the mutual behavior of $\mathbf{\Phi}, \mathbf{\Psi}$. In [9] is stated

$$P_{failure} < N\mathrm{e}^{-\frac{cL}{\mu^2(\mathbf{\Psi},\mathbf{\Phi})^{(L)}M}}, \quad c > 0.$$

Here $\mu(\mathbf{\Psi}, \mathbf{\Phi})$ is the maximum coherence between rows of $\mathbf{\Psi}$ and $\mathbf{\Phi}$ i.e.

$$\mu\left(\mathbf{\Psi}, \mathbf{\Phi}^{(L)}\right) = \sqrt{N} \max_{1\le a\le N, 1\le b\le L} |\langle \mathbf{\Psi}_a, \mathbf{\Phi}_b\rangle|,$$

where $\mathbf{\Psi}_a$ denotes the a-th row of $\mathbf{\Psi}$ and $\mathbf{\Phi}_b$ the b-th row of $\mathbf{\Phi}$. From this statement it follows that the probability of reconstruction is almost surely 1 for

$$L \geq \mu^2 (\mathbf{\Psi}, \mathbf{\Phi}) \frac{M \log(N)}{c} \tag{4.55}$$

In other words, the lower the maximum coherence between $\mathbf{\Psi}$ and $\mathbf{\Phi}$, the lower the number of required samples. Therefore a random matrix $\mathbf{\Phi}$ is used. In [15], Donoho has shown that the coherence between i.i.d. Gaussian $\mathbf{\Phi}$ and any unitary $\mathbf{\Psi}$ is considerably small.

In order to recover the M-sparse $\mathbf{x}$ from $\mathbf{y}$ one has to solve a ℓ_0 minimization problem, i.e.

$$\operatorname*{argmin}_{\tilde{\mathbf{x}} \in \mathbb{C}^N} \|\tilde{\mathbf{x}}\|_0, \tag{4.56}$$

$$\text{s.t. } \mathbf{\Phi}\mathbf{\Psi}\tilde{\mathbf{x}} = \mathbf{z}. \tag{4.57}$$

With $\|\mathbf{x}\|_0$ we denote the number of non zero entries in $\mathbf{x}$ even though $\|\cdot\|_0$ is not a norm. This is an NP hard problem, but in [11] it is shown that if the matrix $\mathbf{\Psi}\mathbf{\Phi}$ fulfills the restricted isometry property (RIP) (see, e.g. [20]), the solution of problem (4.56) is equivalent to the solution of the L_1 minimization problem

$$\operatorname*{argmin}_{\tilde{\mathbf{x}} \in \mathbb{C}^N} \|\tilde{\mathbf{x}}\|_1,$$
$$\text{s.t. } \mathbf{\Phi}\mathbf{\Psi}\tilde{\mathbf{x}} = \mathbf{z}.$$

The latter problem can be solved via linear programming.

If we set $\mathbf{A} := \mathbf{\Phi}\mathbf{\Psi}$ the problem above can be interpreted as the problem of recovering the indices j and the entries x_j, $j = 1, \ldots, M$, from L linear measurements $\mathbf{z}$, which we can approach with our generalized Prony Method applied to finite dimensional vector spaces. The advantage of the generalized Prony Method is that we can guarantee the recovery of $\mathbf{x}$ with the minimum of $2M$ measurements, if we can identify a suitable linear operator $\mathcal{A}$, such that the entries z_k can be represented as $z_k = \mathcal{A}^k\mathbf{x}$. As we have seen in Remark 4.23, the first $2M$ rows of the Fourier matrix $\mathbf{F}_N$ fulfill this condition. Therefore, the generalized Prony Method provides a deterministic solution to this recovery problem in the noiseless case. Unfortunately, it is not stable for large M and hence does not apply to essentially compressible signals. The compressed sensing approach on the other hand can only guarantee recovery with a certain probability and needs $\mathcal{O}(M \log N)$ measurements. This means that not only the sparsity M, but also the dimension of the original signal space $\mathbb{C}^N$ defines the number of needed input values and furthermore, the matrix $\mathbf{A}$ has to fulfill the RIP, which is non trivial to check. Then again, under the assumptions made above for compressed sensing, we can guarantee stable recovery of $\mathbf{x}$, which we lack up until now for our generalized Prony Method.

4.5 Prony Method for the Differential Operator

Finally, we want to mention that the exponentials can also be seen as eigenfunctions of differential operators. Let us consider the vector space $C^\infty(\mathbb{R})$ of infinitely differentiable functions, and let $\frac{\mathrm{d}}{\mathrm{d}x} : C^\infty(\mathbb{R}) \to C^\infty(\mathbb{R})$ be the differentiation operator. We observe that $\{T : T \in \mathbb{C}\}$ is a set of pairwise distinct eigenvalues of $\frac{\mathrm{d}}{\mathrm{d}x}$ and by

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^{Tx} = T\mathrm{e}^{Tx}$$

we can uniquely relate the eigenfunction e^{Tx} to the eigenvalue T. Let x_0 be a fixed real number. Then, with $F(f) := f(x_0)$, $\forall f \in \mathbb{C}^\infty(\mathbb{R})$, we can again apply Theorem 1.2 for recovering the sparse sum of exponentials

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{T_j x} \tag{4.58}$$

with pairwise different $T_j \in \mathbb{C}$. The reconstruction of f can be uniquely performed using the values

$$F\left(\frac{\mathrm{d}^k}{\mathrm{d}x^k} f\right)(x) = f^{(k)}(x_0), \quad k = 0, \ldots, 2M-1,$$

where $x_0 \in \mathbb{R}$ can be chosen arbitrarily.

Algorithm 4.24 (Reconstruction of f in (4.58))

Input: $F\left(\frac{\mathrm{d}^k}{\mathrm{d}x^k} f\right)(x)$, $k = 0, \ldots, 2N-1$. Optionally an upper bound L for the sparsity M, else $L = N$.

1. (a) Construct the Hankel matrix

$$\mathbf{H}_{2N-L,L+1} = \left(F\left(\frac{\mathrm{d}^{k+m}}{\mathrm{d}x^{k+m}} f\right)(x)\right)_{k,m=0}^{2N-L-1,L}$$

and use the ESPRIT Method 3.1.2, the Matrix Pencil Method 3.2.1 or use 2.1 respectively 2.2 with an svd approach to find the eigenvalues $\lambda_j = T_j$, $j = 1, \ldots, M$ of the active eigenfunctions $\mathrm{e}^{T_j x}$.
Alternatively:

(b) Construct the Hankel matrix

$$\mathbf{H}_{N-1} = \left(F\left(\frac{\mathrm{d}^{k+m}}{\mathrm{d}x^{k+m}} f\right)(x)\right)_{k,m=0}^{N-1}$$

and use 2.1 or 2.2 with an eigenvalue decomposition approach to find the eigenvalues $\lambda_j = T_j$, $j = 1, \ldots, M$ of the active eigenfunctions $\mathrm{e}^{T_j x}$.

2. Compute the coefficients c_j by solving the overdetermined system

$$\left(F\left(\frac{\mathrm{d}^k}{\mathrm{d}x^k}f\right)(x)\right) = \sum_{j=1}^{M} c_j F(T_j^k \mathrm{e}^{T_j k}) \qquad k = 0, \ldots, 2N-1.$$

Output: M, c_j, T_j, $j = 1, \ldots, M$.

Moreover, let $\{w_\ell\}_{\ell=0}^r$ be a basis of the space of polynomials of degree at most r and $\deg(w_\ell) = \ell$, $\ell = 0, \ldots, r$. Then we easily check that the functions $w_\ell(x)\mathrm{e}^{Tx}$, $\ell = 0, \ldots, r$, form linearly independent generalized eigenfunctions of multiplicity $\ell + 1$ of the linear operator $\frac{\mathrm{d}}{\mathrm{d}x}$, and Theorem 1.4 applies for the recovery of sparse expansions of the form

$$f(x) = \sum_{j=1}^{M} \sum_{\ell=0}^{r} c_{j,\ell} w_\ell(x) \mathrm{e}^{T_j x}$$

using the derivative values $f^{(k)}(x_0)$, $k = 0, \ldots, 2M(r+1) - 1$, for $r \geq 0$.

Eigenfunctions of monomial form can also be obtained using suitable differential operators. Let $d_x : C^\infty(\mathbb{R}) \to C^\infty(\mathbb{R})$ be the differential operator of the form

$$d_x f(x) := \frac{\mathrm{d}}{\mathrm{d}x}(xf(x)) = f(x) + xf'(x).$$

Then we have

$$d_x(x^p) = \frac{\mathrm{d}}{\mathrm{d}x}(x^{p+1}) = (p+1)x^p, \qquad p \in \mathbb{R},$$

and the operator d_x possesses the set $\{p + 1 : p \in \mathbb{R}\}$ of pairwise different eigenvalues with corresponding eigenfunctions x^p. We consider now a sparse monomial expansion of the form

$$f(x) = \sum_{j=1}^{M} c_j x^{p_j}$$

with $c_j \in \mathbb{C} \setminus \{0\}$ and pairwise different $p_j \in \mathbb{R}$. Using Theorem 1.2, this expansion can be completely recovered from $F((d_x)^k f)$, $k = 0, \ldots 2M-1$. A simple induction argument shows that the values $(d_x)^k f$ can be obtained recursively from the derivative values $f^{(\ell)}(x_0)$, $\ell = 0, \ldots, 2M-1$.

Lemma 4.25 *For $a_{k,\ell} = a_{k-1,\ell-1} + 2a_{k-1,\ell}$, $k, \ell > 0$ with $a_{k,0} = 1$, $a_{k,\ell=0}$ for $\ell > k$ and $a_{0,1} = 1$, the functions $(d_x)^k f(x)$ can be represented as*

$$(d_x)^k f(x) = \sum_{\ell=0}^{k} a_{k,\ell} x^\ell f^{(\ell)}(x).$$

Proof. We observe

$$(d_x)x^\ell f^{(\ell)}(x) = 2x^\ell f^{(\ell)} + x^{\ell+1} f^{(\ell+1)}, \quad \ell \geq 1,$$

and proof the induction argument by computing

$$\begin{aligned}
(d_x)^k f(x) &= (d_x) \sum_{\ell=0}^{k-1} a_{k-1,\ell} x^\ell f^{(\ell)}(x) \\
&= (d_x) \underbrace{a_{k-1,0}}_{=1} f(x) + \sum_{\ell=1}^{k-1} a_{k-1,\ell} (d_x) x^\ell f^{(\ell)}(x) \\
&= f(x) + \underbrace{1}_{:=a_{k-1,0}} \cdot x f(x) + \sum_{\ell=1}^{k-1} a_{k-1,\ell} \left(2x^\ell f^{(\ell)}(x) + x^{\ell+1} f^{(\ell+1)}(x) \right) \\
&= f(x) + \sum_{\ell=1}^{k} (2 \underbrace{a_{k-1,\ell}}_{a_{k-1,k}=0} + a_{k-1,\ell-1}) x^\ell f^{(\ell)}(x)
\end{aligned}$$

Therefore we have $a_{k,\ell} = 2a_{k-1,\ell} + a_{k-1,\ell-1}$ for $\ell \geq 1$ and $a_{k,0} = 1$. Note that by assumption $a_{k-1,\ell} = 0$, if $\ell > k$. For that reason we have $a_{k,k} := 2 \cdot 0 + a_{k-1,\ell-1} = 1$ in the last step. ■

In this chapter we saw examples of suitable linear operators and their eigenfunctions that apply to our generalized Prony Method. The freedom of choice that we have in selecting the functional F inherits further generalizations which will be part of our considerations in the next chapter.

5 Extending the Generalized Prony Method

The examples in the last chapter display the range of our generalized Prony Method, concerning eigenfunctions of suitable linear operators. In this chapter we want to examine possible extensions to multivariate signals and we want to study the functional F that is introduced in Theorem 1.2.

5.1 Freedom of Choice in the Functional F

We saw that we can reconstruct M-sparse functions f of the form (1.8) when information in the form $F(\mathcal{A}^k f)$, $k = 0, \dots, L$, $L \geq 2M-1$ are given. But up until now we have not exploited the freedom of choice in the functional F. In this chapter we want to elaborate possible classifications of the functional F.

Once we made a choice for the operator $\mathcal{A}$ we have fixed the set of functions, with which we can represent f. As we have seen, these functions are the eigenfunctions of the operator $\mathcal{A}$. In order to use Theorem 1.2 and thereby the generalized Prony Method, we need input data that use the k-th power of the operator $\mathcal{A}$ applied to f. These powers define the kind of input data we need. As we have seen, for example the translation operator (4.34) needs equidistant sampling points, the Sturm-Liouville (4.49) needs first derivatives of f. Of course the application of the k-th power of $\mathcal{A}$ to f gives an element of the vector space V which can in general not be handled as an input data. We need a functional $F : V \to \mathbb{C}$, that reduces the information stored in $\mathcal{A}^k$ to a single value. And here we have a huge freedom in choosing the functional F. We hope that we can use this freedom in order to stabilize the method.

The first examples of functionals that might come to mind are convolutions of elements f from the vector space V with *finitely supported* or at least *essentially finite supported kernels φ with respect to V*. With essentially finite supported with respect to V we mean kernels φ that decay fast enough, such that $(f * \varphi)(x) < \infty$ for all $x \in \mathbb{C}$ and all $f \in V$. The *continuous convolution* of two functions f, g is defined as

$$(f * g)(x) := \int_{-\infty}^{\infty} f(\xi) g(x - \xi) \mathrm{d}\xi$$

We define the mirrored kernel $\tilde{g}(x) := g(-x)$ and observe

$$(f * \tilde{g})(x) := \int_{-\infty}^{\infty} f(\xi) g(\xi - x) \mathrm{d}\xi.$$

This defines of course a function and not a scalar, but we can set F to be

$$F_{x_0,g}(f) := (f * \tilde{g})(x_0) = \int_{-\infty}^{\infty} f(\xi) g(\xi - x_0) \mathrm{d}\xi.$$

Note that for example the functional $F_k(f) := f^{(k)}(x_0)$ that evaluates the k-th derivative of a function f at the point x_0 does not lie in the class defined above,

but still fulfills all requirements needed as the functional F in the generalized Prony Method.

For the purpose of understanding let us fix the operator $\mathcal{A}$ as the translation operator $\mathcal{A}_h f(x) := f(x+h)$ and observe the effect of different kernels φ to the signal

$$f(x) = 2\sin(0.4x) + \cos(x) - 2\mathrm{e}^{-1.1x} + 0.5\mathrm{e}^{x/10}. \tag{5.59}$$

Dirac distribution

Let φ be the **Dirac** distribution δ_{x_0} with the property $\langle \delta_{x_0}, f\rangle = \int_{-\infty}^{\infty} \delta(x - x_0) f(x)\mathrm{d}x = f(x_0)$. Then the functional F becomes the point evaluation functional

$$F(f) := (\varphi * f)(x_0) = \langle \delta_{x_0}, f\rangle = \int_{-\infty}^{\infty} \delta(x - x_0) f(x)\mathrm{d}x = f(x_0)$$
$$F(\mathcal{A}_h^\ell f) = \int_{-\infty}^{\infty} \delta(x - x_0) f(x + \ell h)\mathrm{d}x = f(x_0 + \ell h).$$

We know this functional well from the previous examples, but for the sake of completion we show a visualization in figure 10.

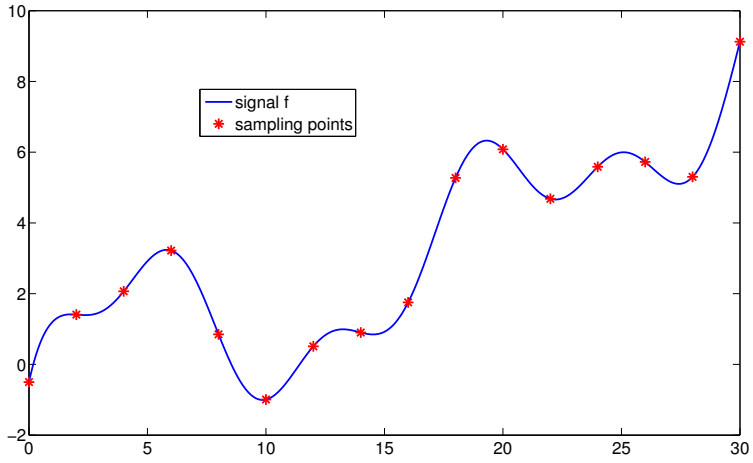

Figure 10: The blue graph shows the characteristics of the signal f in the interval $[0, 30]$, whereas the red stars show the equidistant input data $F(\mathcal{A}_2^\ell f)$, $\ell = 0, \ldots, 15$.

This shows that the classical Prony Method falls into this setting.

Constant spline

Let

$$\varphi(x) = \begin{cases} 1, & x \in [-1, 1], \\ 0, & \text{else} \end{cases}$$

be the **constant spline**, then the functional F becomes an integral evaluation of consecutive intervals

$$F(f) := \int_{-\infty}^{\infty} \varphi(x) f(x) \mathrm{d}x = \int_0^1 f(x) \mathrm{d}x$$
$$F(\mathcal{A}_h^\ell f) := \int_0^1 f(x + \ell h) \mathrm{d}x = \int_{\ell h}^{(\ell+1)h} f(x) \mathrm{d}x.$$

We apply the integral approach to our test scenario and depict an example in Figure 11.

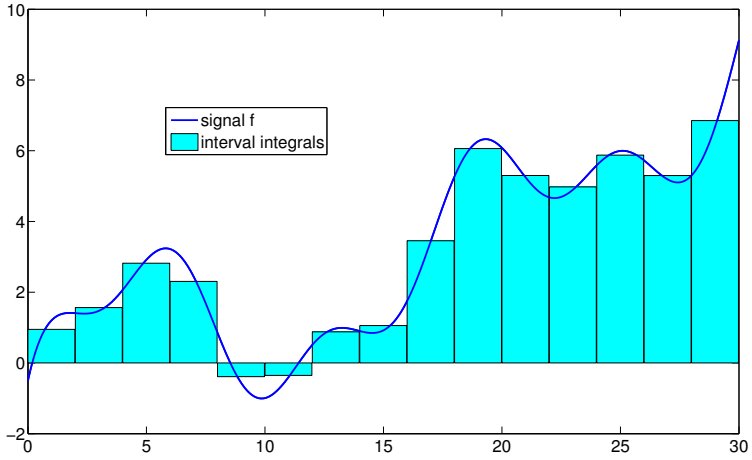

Figure 11: The blue graph shows the signal f in the interval $[0, 30]$. The bars represent the integral $F(\mathcal{A}_2^\ell f) = \int_{\ell h}^{(\ell+1)h} f(x) \mathrm{d}x$, $\ell = 0, \ldots, 14$.

We want to present a way, how this approach might become expedient if we are provided with a lot of, possibly, erroneous input data. The idea is to approximate the integral at a certain predefined interval out of given sampling points, e.g.

$$\int_{\ell h}^{(\ell+1)h} f(x) \mathrm{d}x \approx \sum_{k=1}^{N} \omega_{\ell,k} f(k), \quad c_\ell \le k \le C_\ell,$$

with quadrature weights $\omega_{\ell,k}$, $k = 1 \ldots, N$. Note that for $\omega_{\ell,k} = \frac{1}{N}$, $c_\ell = \ell h$, $C_\ell = (\ell + 1)h$ and equidistant input data the quadrature formula becomes the

arithmetic mean. The prospect of this approach is that the smoothing operation of evaluating an integral over a certain interval levels out the failures with which the input data are corrupted.

Furthermore, this special functional is also advantageous for a low noise level, if very many input data are given. Because we saw in Example 2.4 that the running time of the Prony Method increases dramatically for a high number of input data and now we can use all $2N-1$ input data to approximate $2L+1$ consecutive interval-integrals. Thus, we use all the given information in a preprocessing step in order to compute $2L+1$ input data for our generalized Prony Method, where L is an upper bound for the sparsity M, but $L \ll N$.

The probably most promising feature is that we are able to process non-equidistant input data. Assume we have many sampling points given that do not lie on an equispaced grid, then we can artificially construct equidistant input data by approximating consecutive interval-integrals out of the non-equidistant given data.

Let us look at an example.

Example 5.1 *For the test scenario stated above we use* 1500 *input values* $\tilde{f}(sh_0) = f(sh_0)+e_{sh_0}$, $s = 0, \ldots, 1499$, $h_0 = 1/50$, *with additional gaussian noise* e_{sh_0}, *with zero mean and variance* $\sigma^2 = 1$. *The PSNR for these perturbed input data is about* $20dB$. *The input values are depicted as black dots in Figure 12. We choose the interval length* $h = 2$ *and the mean value of* 100 *consecutive sampling points as integral approximation*

$$\int_{kh}^{(k+1)h} f(x)\mathrm{d}x \approx \frac{1}{100}\sum_{s=1}^{100} \tilde{f}(s+k), \quad k = 0, \ldots, 14,$$

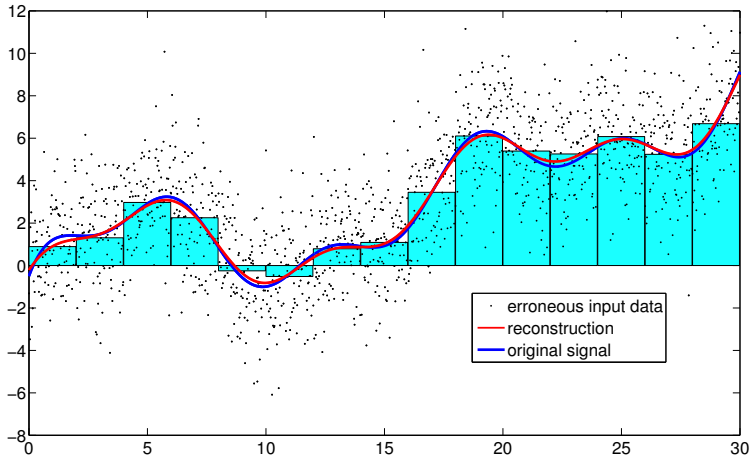

Figure 12: The blue graph shows the signal f in the interval $[0, 30]$. The bars represent the approximated interval-integrals. The black dots depict the erroneous given sampling points and the red graph shows the characteristic of the reconstructed signal.

We use the ESPRIT Method with the 15 *approximated interval integrals as described above and reconstruct the eigenvalues* $\lambda_1 = 0.4\mathrm{i}$, $\lambda_2 = -0.4\mathrm{i}$, $\lambda_3 = \mathrm{i}$, $\lambda_4 = -\mathrm{i}$, $\lambda_5 = -1.1$, $\lambda_6 = 0.1$ *to the active eigenfunctions of the signal* f *in (5.59) with a failure in the range of* 10^{-2}. *If we use instead all* 1500 *given sampling points, without preprocessing, the ESPRIT Method is considerably slower than the approach introduced here and the method fails to detect any eigenvalue* λ_j, $j = 1, \ldots, 6$ *with an accuracy below* 0.3. *In other words, without the benefiting smoothing feature of the interval integration, the method fails to detect the active eigenfunctions in the perturbed signal.*

As mentioned before, due to our freedom of choosing the interval length artificially, we are no longer restricted to equidistant input data. Of course, there have to be enough non equispaced input data available in every newly chosen interval to approximate the integral of f in that interval. Let us look at an example.

Example 5.2 *Assume* 61 *non equidistant given sampling points of the signal* f *in 5.59 in the interval* $[0, 30]$. *That means, on average we have a distance* 0.5 *between two sampling points. In Figure 13 we see the original signal depicted in blue and the non equidistant input data as red dots. In a first step we make a cubic spline interpolation of the given sampling points. In the next step we approximate the integral in every artificially chosen interval* $[k, k + h]$, $k = 0, \ldots, 14$, $h = 2$ *as the exact integral of the spline function in those intervals.*

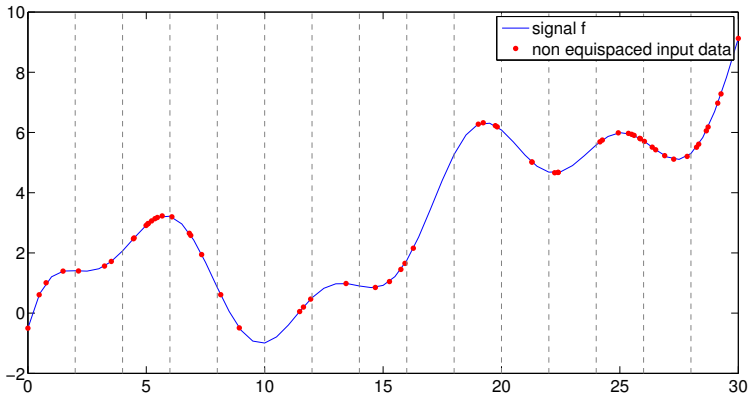

Figure 13: The blue graph shows the signal f in the interval $[0, 30]$. The red dots depict 61 non equidistant sampling points and the gray dashed lines indicate the arbitrarily chosen intervals.

Again we choose 15 *intervals of length* 2 *and approximate the integral in those intervals. In Figure 13 we see that* 61 *non equidistant input data are rather few, because there are some intervals that inherit almost too few samples to reliably approximate the integral in that interval, without a previous spline interpolation. Nevertheless, with our approach for non equidistant input data we get to recover the eigenvalues* λ_j, $j = 1, \ldots, M$ *quit accurate, except for* $\lambda_5 = -1.1$. *The results are presented in Table 12.*

j	$\tilde{\lambda}_j$	λ_j
1	0.4006i	0.4i
2	−0.4006i	−0.4i
3	0.9987i	i
4	−0.9987i	−i
5	−1.31166	−1.1
6	0.10020	0.1

Table 12: The computed eigenvalues of the signal f in 13.

Note that in Figure 13 there is each a sampling point at 0 *and at* 30. *This is no loss of generality, because for a given set of non equidistant sampling points we can always assume the first given sampling point to be at zero and the last one at any preferred point. Translating the first sampling point to* 0 *does not effect the eigenvalues* λ_j, $j = 1, \ldots, M$ *but only the initial phase of each exponential function in* f, *which just effects the coefficients* c_j, $j = 1, \ldots, M$. *Setting the last sampling point to an arbitrary point apart from* 0 *on the other hand scales the eigenvalues* λ_j, $j = 1, \ldots, M$.

Remark 5.3 *The Prony Method for non equidistant input data was already investigated, in our paper [36].*

Let us summarize the advantages that we can observe with this approach.

1. For input data with gaussian noise corruption, the smoothing operation of approximating an integral has an error correcting feature.

2. All input data can be used and we still end up with a method that works in reasonably short time.

3. Maybe the most interesting feature is the possibility of handling non equispaced input data. If we adjust the weights $\omega_{\ell,k}$ properly according to the local step width we can construct approximations of the needed input values in the form of a convolution as described above.

4. Instead of approximating the integral in a certain interval as the weighted sum of the (possibly non equispaced) given input data, we can also approximate the given data with a spline function first and evaluate the desired integral approximation of the original function as the exact integral of the spline in a second step.

It is reasonable to assume that the more data we use to approximate the convolution of the underlying function with a certain kernel, the better this approximation will be. That is why we can also look at kernels with a wider support than the interval we choose.

Gaussian

An extreme version of a kernel that has a wider support than the interval we choose is an infinitely supported kernel. Of course here we have to ensure that the kernel will decay faster than any eigenfunction of the operator $\mathcal{A}$, such that the convolution of the underlying function f with the chosen kernel will stay finite. A kernel that meets this condition for the translation operator, and thus the exponential functions as eigenfunctions, is the Gauss function.

Let φ be the **gauss function**

$$\varphi_b(x) := \mathrm{e}^{-bx^2}, \quad b > 0.$$

The functional F becomes a convolution of the Gauss kernel with the function f translated by $h\ell$,

$$F(\mathcal{A}_h^\ell f) := \int_{-\infty}^{\infty} f(x+\ell h)\mathrm{e}^{-bx^2}\mathrm{d}x = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-b(x-\ell h)^2}\mathrm{d}x.$$

Other Functionals

Linear Combinations

Of course also linear combinations of different kernels are again suitable kernels. We can, for example use

$$\begin{aligned} F(f)(x) &:= \sum_{k=-N}^{N} \alpha(k)[\langle \delta_{x-k}, f\rangle + \langle \delta_{x+k}, f\rangle] \\ &= \sum_{k=-N}^{N} \alpha(k)\left[f(x+k) + f(x-k)\right]. \end{aligned}$$

Let ψ be a functional $\psi : \mathbb{Q} \to (-1,1)$ with $\psi(x) = \psi(-x)$ and $\psi(x) = 0$, $|x| \geq 1$. Possible realizations of ψ are, e.g. spline functions, centered at zero and shrunken to the interval $[-1,1]$ or

$$\psi(x) := \begin{cases} \mathrm{e}^{1/(x^2-1)}, & x \in (-1,1) \\ 0, & \text{else} \end{cases}$$

if $\psi : \mathbb{R} \to \mathbb{R}$ is preferred infinitely differentiable. If we set now

$$\alpha(k) := \psi\left(\frac{k}{N}\right) f(k)$$

we end up with the functional

$$F_{N,\psi}(f)(x) := \sum_{k\in\mathbb{Z}} \psi\left(\frac{k}{N}\right) f(k) \frac{f(x-k) + f(x+k)}{2},$$

which was proposed in [19] as a preconditioning for erroneous input data for Prony-like methods in order to stabilize the calculations.

Remark 5.4 *As we already mentioned in the introduction of this subchapter, we can also use functionals that cannot be written as a convolution of the function f with a certain kernel. Let us take for example the functional*

$$\begin{aligned} F(f)(x) &= f'(x_0), \\ F(\mathcal{A}^k f)(x) &= (\mathcal{A}^k f)'(x_0), \end{aligned}$$

which uses derivatives instead of sampling points.

Let us close this subchapter about the functional F with an open problem.

In order to apply theorem 1.2 we have to ensure that $F(v_j) \neq 0$ and $|F(v_j)| < \infty$, $j \in I$, for all eigenfunctions of the operator $\mathcal{A}$. But we have not taken into account that the functional F applied to different eigenfunctions v_j, $j \in I$ can differ vastly in magnitude. An interesting question in that direction is, if we benefit from constructing special functionals with a priori knowledge of the active eigenfunctions in the signal at hand. To be more specific, assume we have already some (presumably perturbed) results from a first run of our generalized Prony Method. Can we construct a functional F such that $|F(v_j)| > |F(v_k)|$ for all $j \in J$ and $k \in I \backslash J$?

5.2 Multivariate Version of the Generalized Prony Method

The reconstruction method also applies to the multivariate case. Let $S_{\mathbf{a}} : C(\mathbb{R}^d) \to C(\mathbb{R}^d)$ be the shift operator with

$$S_{\mathbf{a}} f(x_1, \ldots, x_d) = f(x_1 + a_1, x_2 + a_2, \ldots, x_d + a_d),$$

with the set of eigenfunctions $\{\mathrm{e}^{\mathbf{T}\cdot\mathbf{x}} = \mathrm{e}^{T_1 x_1 + \cdots + T_d x_d} : T_\ell \in \mathbb{C}, \operatorname{Im} T_\ell \in [-\frac{\pi}{a}, \frac{\pi}{a}), \ell = 1, \ldots, d\}$. The corresponding eigenvalues $\mathrm{e}^{\mathbf{T}\cdot\mathbf{a}}$ allow a unique conclusion to the corresponding eigenfunction $\mathrm{e}^{\mathbf{T}\cdot\mathbf{x}}$ if there exists an injective linear mapping that maps $\mathbf{T}$ to $\mathbf{T}\cdot\mathbf{a}$. This condition can be satisfied by a suitable restriction of $\mathbf{T}$ and a special choice of $\mathbf{a}$. Let $f(\mathbf{x})$ be of the form

$$f(\mathbf{x}) = \sum_{j=1}^{M} c_j \mathrm{e}^{\mathbf{T}_j \cdot \mathbf{x}}$$

with $\mathbf{T}_j = (T_{j,1}, \ldots, T_{j,d})^{\mathrm{T}}$, where $T_{j,\ell} \in \mathbb{N}$ for $j = 1, \ldots, M$, $\ell = 1, \ldots, d$. Choose now pairwise relatively prime numbers $p_1, \ldots, p_d$ with $p_\ell > \max\limits_{j=1,\ldots,M} T_{j,\ell}$ for $\ell = 1, \ldots, d$. Further, let $N = p_1 p_2 \cdots p_d$, and $\mathbf{a} := (\frac{N}{p_1}, \ldots, \frac{N}{p_d})$. Then each variable $\mathbf{T}_j \in \mathbb{N}^d$ can be uniquely determined from $\tau_j := \mathbf{T}_j \cdot \mathbf{a}$ using the reverse steps of the Chinese remainder theorem which is discussed for example in [51]. We have

$$\tau_j = \sum_{\ell=1}^{d} \frac{T_{j,\ell} N}{p_\ell}.$$

Hence, $\tau_j \equiv T_{j,\ell} \bmod p_\ell$ for $1 \le \ell \le d$, and we can recover $\mathbf{T}_j$ from τ_j by $T_{j,\ell} = \tau_j - p_\ell \left\lfloor \frac{\tau_j}{p_\ell} \right\rfloor$, see [23]. Unfortunately, this procedure is highly unstable.

Another procedure for recovery of multivariate exponential sums is based on the determination of $\mathbf{T}_j \in \mathbb{C}^d$, $\operatorname{Im} \mathbf{T}_j \in [-\frac{\pi}{a}, \frac{\pi}{a})^d$, from different scalar products $\mathbf{T}_j \cdot \mathbf{a}_1$, $\mathbf{T}_j \cdot \mathbf{a}_2$, etc., see [38, 41].

5.3 Analyzing Translations of Multivariate Gaussians in Time Domain

In this subchapter we want to present a very new approach to analyze translations of certain kernels when sampling points in the time domain are given. The results presented here evolved from discussion with Robert Schaback. Note that the analysis of translations of 1-periodic window functions via Prony's method was already discussed in our paper [36].

Define the kernel

$$K_b(x) := \mathrm{e}^{-b x^{\mathrm{T}} x}, \quad b > 0$$

component wise for $x \in \mathbb{R}^d$ and consider

$$f(x) = \sum_{j=1}^{M} c_j K_b(x - s_j) = \sum_{j=1}^{M} c_j \mathrm{e}^{-b(x-s_j)^{\mathrm{T}}(x-s_j)}.$$

For known b we want to determine the unknown translations $s_j \in \mathbb{R}^d$ and the corresponding coefficients c_j out of given sampling points $f(n)$, $n \in \Omega \subset \mathbb{R}^d$. We define the Prony polynomial

$$P(z) = \sum_{k=0}^{N} p_k z^{n_k}, \quad n_k \in K \subset \mathbb{N}^k$$

with roots e^{2bs_j}, i.e. $P(\mathrm{e}^{2bs_j}) = 0$, $j = 1, \ldots, M$. Let us investigate the following weighted sum

$$\begin{aligned}
&\sum_{k=1}^{N} q_k f(n_k + m)\alpha(m,k) \\
=&\sum_{k=1}^{N} q_k \sum_{j=1}^{M} c_j \mathrm{e}^{-b(n_k+m-s_j)^{\mathrm{T}}(n_k+m-s_j)}\alpha(m,k) \\
=&\sum_{j=1}^{M} c_j \sum_{k=1}^{N} q_k \mathrm{e}^{-b(n_k^{\mathrm{T}}n_k+m^{\mathrm{T}}m+s_j^{\mathrm{T}}s_j+2n_k^{\mathrm{T}}m-2n_k^{\mathrm{T}}s_j-2m^{\mathrm{T}}s_j)}\alpha(m,k) \\
=&\sum_{j=1}^{M} c_j \underbrace{\mathrm{e}^{-b(m^{\mathrm{T}}m+s_j^{\mathrm{T}}s_j-2m^{\mathrm{T}}s_j)}}_{=\mathrm{e}^{-b(m-s_j)^{\mathrm{T}}(m-s_j)}} \sum_{k=1}^{N} \underbrace{p_k}_{:=q_k\mathrm{e}^{-b(n_k^{\mathrm{T}}n_k)}} \mathrm{e}^{-2b(n_k^{\mathrm{T}}m-n_k^{\mathrm{T}}s_j)}\alpha(m,k) \\
=&\sum_{j=1}^{M} c_j \mathrm{e}^{-b(m-s_j)^{\mathrm{T}}(m-s_j)} \sum_{k=1}^{N} p_k \left(\mathrm{e}^{2bs_j}\right)^{n_k} \underbrace{\alpha(m,k)}_{:=\mathrm{e}^{2bm^{\mathrm{T}}n_k}} \\
=&\sum_{j=1}^{M} c_j \mathrm{e}^{-b(m-s_j)^{\mathrm{T}}(m-s_j)} \underbrace{\sum_{k=1}^{N} p_k \left(\mathrm{e}^{2bs_j}\right)^{n_k}}_{=P(\mathrm{e}^{2bs_j})=0} = 0.
\end{aligned}$$

Defining $\alpha(m,k) := \mathrm{e}^{2bm^{\mathrm{T}}n_k}$ and $q_k := p_k \mathrm{e}^{b(n_k^{\mathrm{T}}n_k)}$ we have to solve the following system

$$\begin{aligned}
\mathbf{Hq} =& \mathbf{0}, \\
\mathbf{H} :=& (f(n_k + m_\ell)\mathrm{e}^{2bm_\ell^{\mathrm{T}}n_k})_{\ell,k=1}^{N} \\
\mathbf{q} :=& (p_k \mathrm{e}^{b(n_k^{\mathrm{T}}n_k)})_{k=1}^{N},
\end{aligned}$$

thus we compute the coefficients of the Prony polynomial by $p_k = q_k \mathrm{e}^{-b(n_k^{\mathrm{T}}n_k)}$.

Remark 5.5 *In one dimension we can define a polynomial as a product of linear factors, i.e.*

$$P(x) = \prod_{j=1}^{M}(x - s_j) = \sum_{k=0}^{M} p_k x^k,$$

and we know that it has a monomic representation. These two possibilities to define a polynomial are the key to Prony's method as we have introduced it in this thesis.

In higher dimensions we generally cannot define a polynomial as a product of linear factors, since higher dimensional polynomials do not necessarily decompose into linear factors. That is why we just postulated that there exists a polynomial with the desired roots. But a necessary condition for the minimal degree of the multivariate polynomial is an open problem. One way to get a sufficient condition is to decouple the multivariate polynomial for given roots $\mathbf{s}_j \in \mathbb{R}^d$, $j = 1, \ldots, M$

$$\begin{aligned} P_i(x_i) &:= \prod_{j=1}^{M}(x_i - s_{i,j}) = \sum_{k=0}^{M} p_{i,k} x_i^k, \quad i = 1, \ldots, d \\ P(\mathbf{x}) &:= \begin{pmatrix} P_1(x_1) \\ \vdots \\ P_d(x_d) \end{pmatrix} \\ \Rightarrow P(\mathbf{s}_j) &= 0. \end{aligned}$$

Since the multi index n_k *defines the sampling points we need, this approach samples along the axis. And here we need* $2M$ *sampling points in each dimension. Can we guarantee to stay closer to the origin with the needed sampling values if we allow mixed terms?*

Now, that we have seen the general approach, let us consider the simpler one dimensional case. Assume a function

$$f(x) = \sum_{j=1}^{M} c_j \mathrm{e}^{-b(x-s_j)^2}$$

and given sampling points $f(hn)$, $n = 0, \ldots, N$, $h > 0$. The task is to determine the unknowns s_j and the coefficients c_j. We define the Prony polynomial and some assisting functions

$$\begin{aligned} P(z) &:= \prod_{j=1}^{M}(z - \mathrm{e}^{2bhs_j}) = \sum_{k=0}^{M} p_k z^k, \\ \alpha(m,k) &:= \mathrm{e}^{2bhmk} \\ q_k &:= p_k \mathrm{e}^{bhk^2}, \end{aligned}$$

with roots e^{2bhs_j}, i.e. $P(\mathrm{e}^{2bhs_j}) = 0$. Now we analyze the following weighted sum

$$\begin{aligned}
&\sum_{k=0}^{M} q_k f(hk+m)\alpha(m,k) \\
&= \sum_{k=0}^{M} q_k \sum_{j=1}^{M} c_j \mathrm{e}^{-b(hk+m-s_j)^2}\alpha(m,k) \\
&= \sum_{j=1}^{M} c_j \sum_{k=0}^{M} q_k \mathrm{e}^{-b(h^2k^2+m^2+s_j^2+2hkm-2hks_j-2ms_j)}\alpha(m,k) \\
&= \sum_{j=1}^{M} c_j \mathrm{e}^{-b(m^2+s_j^2-2ms_j)} \sum_{k=0}^{M} p_k \mathrm{e}^{-b(h^2k^2+2hkm-2hks_j)}\mathrm{e}^{2bhmk} \\
&= \sum_{j=1}^{M} c_j \mathrm{e}^{-b(m^2+s_j^2-2ms_j)} \underbrace{\sum_{k=0}^{M} p_k \left(\mathrm{e}^{2bhs_j}\right)^k}_{=0} = 0
\end{aligned}$$

Example 5.6 *Set $M = b = h = c = s = 1$, i.e.*

$$\begin{aligned}
f(x) &= \mathrm{e}^{-(x-1)^2} \\
\alpha(m,k) &= \mathrm{e}^{2mk} \\
q_k &= p_k \mathrm{e}^{k^2}
\end{aligned}$$

We need just one root, so a linear polynomial is sufficient, which means we need a 2×2 matrix $\mathbf{H}$

$$\begin{aligned}
\begin{pmatrix} f(0)\alpha(0,0) & f(1)\alpha(0,1) \\ f(1)\alpha(1,0) & f(2)\alpha(1,1) \end{pmatrix} \begin{pmatrix} p_0 \\ p_1\mathrm{e} \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\begin{pmatrix} \mathrm{e}^{-1} & 1 \\ 1 & \mathrm{e}^{1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1\mathrm{e} \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\Rightarrow P(z) &\approx z - 0.1353352832366127 \\
\log(0.1353352832366127)/2 &\approx -1
\end{aligned}$$

In the multi dimensional case we are faced with a further numerical issue. The zero sets of a d dimensional polynomial usually lie on a $d-1$ dimensional manifold. The numerical calculation of those zero sets is challenging. Furthermore we are interested in specific points e^{2bs_j}, $j = 1, \ldots, M$ that are elements of the zero sets of each Prony related polynomial. Therefore, we have to calculate intersections of the zero sets of different Prony related polynomials. Let us look at a two dimensional example.

Example 5.7 *We set* $b = h = 1$,

$$s_1 = \mathrm{e}^{[1,0]^{\mathrm{T}}} \approx [2.718, 1]^{\mathrm{T}}$$
$$s_2 = \mathrm{e}^{[1,2]^{\mathrm{T}}} \approx [2.718, 7.389]^{\mathrm{T}}$$
$$s_3 = \mathrm{e}^{[-1,3]^{\mathrm{T}}} \approx [0.368, 20.09]^{\mathrm{T}}$$

and consider the function

$$f(x) = \sum_{j=1}^{3} \mathrm{e}^{-(x-s_j)^{\mathrm{T}}(x-s_j)}.$$

Thus $c_1 = c_2 = c_3 = 1$. *We have some freedom in choosing the two dimensional Prony related polynomial* $Q(x)$, $x \in \mathbb{R}^2$, *which in turn defines the needed sampling points. Let us choose for example*

$$Q(x_1, x_2) = q_0 \cdot 1 + q_1 x_2 + q_2 x_2^2 + q_3 x_1 + q_4 x_1 x_2 + q_5 x_1^2$$

and $m_k = (k, 0)^{\mathrm{T}}$, $k = 0, \ldots, 5$, *i.e.* $n_0 = (0, 0)^{\mathrm{T}}, n_1 = (0, 1)^{\mathrm{T}}, n_2 = (0, 2)^{\mathrm{T}}, n_3 = (1, 0)^{\mathrm{T}}, n_4 = (1, 1)^{\mathrm{T}}, n_5 = (2, 0)^{\mathrm{T}}$. *This choice defines the structure of sampling points we need for each row in the Hankel matrix* $\mathbf{H}$. *By varying* m_k *we just shift the pattern defined by* n_k, $k = 0, \ldots, 5$ *in the* x_1, x_2 *plane. The choice* $m_k = (k, 0)^{\mathrm{T}}$, $k = 0, \ldots, 5$ *is somewhat better than other choices, since we use again several points for different rows in* $\mathbf{H}$, *but for the purpose of demonstration let us choose* $\tilde{m}_0 = (0, 0)^{\mathrm{T}}, \tilde{m}_1 = (2, 1)^{\mathrm{T}}, \tilde{m}_2 = (3.5, 0)^{\mathrm{T}}, \tilde{m}_3 = (1, 2.5)^{\mathrm{T}}, \tilde{m}_4 = (5, 2)^{\mathrm{T}}, \tilde{m}_5 = (3, 4)^{\mathrm{T}}$. *We see in Figure 14 the recurring pattern of our choice of* $Q(x)$ *and* $\tilde{m}_k$, $k = 0, \cdots 5$, *where the patterns are color-coded from dark blue for* $\tilde{m}_0$ *to light blue for* $\tilde{m}_5$.

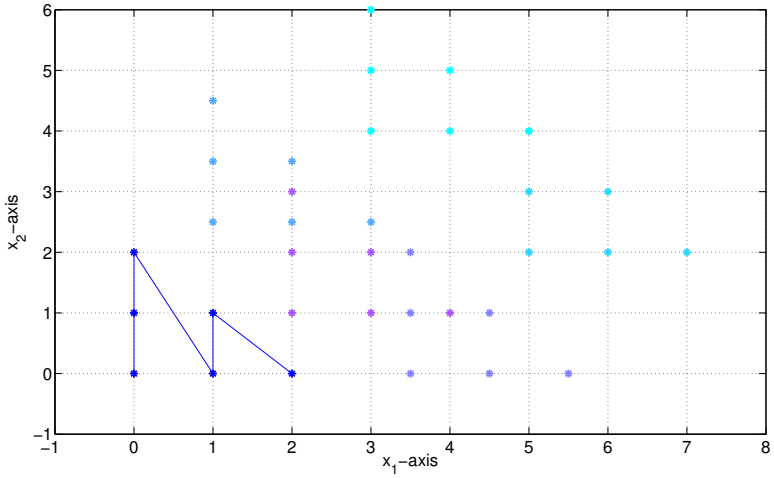

Figure 14: Example of possible sampling patterns for 6 different rows of the Hankel matrix $\mathbf{H}$.

Let

$$\mathbf{H} = \left(f(n_k + m_\ell) \mathrm{e}^{2m_\ell^{\mathrm{T}} n_k} \right)_{\ell=0,k=0}^{5}$$

be the Hankel matrix we use for the Prony Method, then $\operatorname{rank}(\mathbf{H}) = 3$*, since we have three active translations present in the signal* $f(x)$*. For that reason, we also get three different vectors* $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ *that lie in the kernel of* $\mathbf{H}$*. The entries of those vectors define the coefficients of three different Prony related polynomials corresponding to the choice we made in appointing* n_k*,* $k = 0, \ldots, 5$*. We now have to find the common roots of those polynomials.*

The roots of the Prony related polynomials lie on one dimensional manifolds in the x_1, x_2 *plane. We approximate those zero set by fixing* x_1 *which makes the Prony related polynomial only dependent on* x_2 *and its roots can be evaluated as the eigenvalues of the companion matrix. In Figure 15 we vary* x_1 *in steps of* $1/1000$ *and compute the roots of each of the three Prony related polynomials. Indeed we see that these polynomials share three roots at the shifts* s_1, s_2, s_3*, which are depicted as red circles. (The straight lines that appear in Figure 15 are the absolute values of the non real roots of the Prony related polynomials.)*

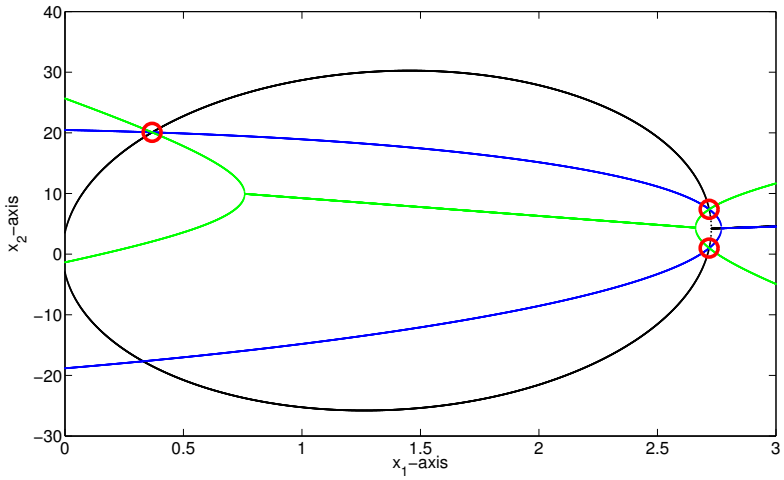

Figure 15: The roots of three different two dimensional Prony related polynomials depicted in blue, green and black. The red circles indicate the location of the shifts s_1, s_2, s_3.

In this last example we have seen that this method for analyzing translations of multivariate kernels works in principal, but there are still open questions, such as developing suitable numerical implementations for finding zero sets of multi dimensional polynomials. Another open problem is to answer the question which criteria we need for the choice of n_k, $k = 0, \ldots, L$ to ensure that the

corresponding Prony related polynomial has the unknown shifts s_j, $j = 1, \ldots, M$ as roots. In chapter 7 we will pay attention to other open problems that occurred during the examination of our generalized Prony Method.

6 Applications of the Prony Method

In this chapter we want to present an application of Prony's method for non destructive material examination. In order to apply Prony's method for this scenario we have to perform a preprocessing step based on Fourier techniques that transforms the problem at hand into a problem we can analyze with the Prony Method. For that reason we will explain this approach in detail in subchapter 6.1 before we start with the actual example in subchapter 6.2. .

6.1 Prony Method after Preprocessing

Up until now we always considered the ansatz that the underlying function of the given data can be expressed as an M-term representation of eigenfunctions of a certain linear operator. Now we want to consider the case that not the data themselves but only a transformed version is M-sparse in the transformed space. In a joined work with Manfred Tasche and Daniel Potts [36] we examined signals that needed a Fourier transform prior to being applicable as input data for Prony's method.

Let $N \in 2\,\mathbb{N}$ be fixed. We introduce an *oversampling factor* $\alpha > 1$ such that $n := \alpha\, N$ is a power of 2. Let $\varphi \in C(\mathbb{R})$ be a 1-periodic even, nonnegative function with a uniformly convergent Fourier expansion. Further we assume that all Fourier coefficients

$$c_k(\varphi) := \int_{-1/2}^{1/2} \varphi(x)\, \mathrm{e}^{-2\pi \mathrm{i} k x}\, \mathrm{d}x = 2 \int_{0}^{1/2} \varphi(x)\, \cos(2\pi k x)\, \mathrm{d}x, \quad k \in \mathbb{Z}$$

are nonnegative and that $c_k(\varphi) > 0$ for $k = 0, \ldots, N/2$. Such a function φ is called a *window function*. We can consider one of the following window functions.

Example 6.1 *A well known window function is the 1–periodization of a Gaussian function (see [17, 18, 52])*

$$\varphi(x) = \sum_{k=-\infty}^{\infty} \varphi_0(x+k), \quad \varphi_0(x) := \frac{1}{\sqrt{\pi b}}\, \mathrm{e}^{-(nx)^2/b}, \quad x \in \mathbb{R},\, b \geq 1$$

with the Fourier coefficients $c_k(\varphi) = \frac{1}{n}\, \mathrm{e}^{-b(\pi k/n)^2} > 0$, $k \in \mathbb{Z}$.

Example 6.2 *Another window function is the 1-periodization of a centered cardinal B–spline (see [5, 52])*

$$\varphi(x) = \sum_{k=-\infty}^{\infty} \varphi_0(x+k), \quad \varphi_0(x) := M_{2m}(nx), \quad x \in \mathbb{R};\, m \in \mathbb{N}$$

with the Fourier coefficients $c_k(\varphi) = \frac{1}{n} \left(\operatorname{sinc} \frac{k\pi}{n}\right)^{2m}$, $k \in \mathbb{Z}$. *With* M_{2m}, $m \in \mathbb{N}$ *we denote the centered cardinal B–spline of order* $2m$.

Example 6.3 *Let $m \in \mathbb{N}$ be fixed. A possible window function is the 1-periodization of the $2m$-th power of a sinc-function (see [31])*

$$\varphi(x) = \sum_{k=-\infty}^{\infty} \varphi_0(x+k)\,, \quad \varphi_0(x) := \frac{N\,(2\alpha-1)}{2m}\,\mathrm{sinc}^{2m}\left(\frac{\pi N x\,(2\alpha-1)}{2m}\right)$$

with the Fourier coefficients $c_k(\varphi) = M_{2m}\big(\frac{2mk}{(2\alpha-1)N}\big)$, $k \in \mathbb{Z}$.

Example 6.4 *Let $m \in \mathbb{N}$ be fixed. As next window function we mention the 1-periodization of a Kaiser–Bessel function (see [29])*

$$\begin{aligned} \varphi(x) &= \sum_{k=-\infty}^{\infty} \varphi_0(x+k)\,, \\ \varphi_0(x) &:= \begin{cases} \frac{\sinh(b\sqrt{m^2-n^2x^2})}{\pi\sqrt{m^2-n^2x^2}} & \text{for } |x| \le \frac{m}{n} \quad \big(b := \pi\,\big(2-\frac{1}{\alpha}\big)\big)\,, \\ \frac{\sin(b\sqrt{n^2x^2-m^2})}{\pi\sqrt{n^2x^2-m^2}} & \text{otherwise} \end{cases} \end{aligned}$$

with the Fourier coefficients

$$c_k(\varphi) = \begin{cases} \frac{1}{n}\,I_0\big(m\sqrt{b^2-(2\pi k/n)^2}\big) & \text{for } |k| \le n\,\big(1-\frac{1}{2\alpha}\big), \\ 0 & \text{otherwise,} \end{cases}$$

where I_0 denotes the modified zero–order Bessel function.

Now we consider a linear combination

$$f(x) = \sum_{j=1}^{M} c_j \varphi(x+s_j) \tag{6.60}$$

of translates with complex coefficients $c_j \neq 0$ and pairwise different shift parameters s_j, where

$$-\frac{1}{2} < s_1 < \ldots < s_M < \frac{1}{2} \tag{6.61}$$

is fulfilled. Then $f \in C(\mathbb{R})$ is a complex-valued 1-periodic function. Further let $N \ge 2M+1$. Assume that perturbed, uniformly sampled data

$$\tilde{f}_l = f\left(\frac{l}{n}\right) + e_l, \quad |e_l| \le \varepsilon_1 \quad l = -n/2, \ldots, n/2-1$$

are given, where the error terms $e_l \in \mathbb{C}$ are bounded by a certain accuracy ε_1 ($0 < \varepsilon_1 \ll 1$). We suppose that $|c_j| \gg \varepsilon_1$, $j = 1, \ldots, M$.

Then we consider the following *nonlinear approximation problem for a sum* (6.60) *of translates*: Determine the pairwise different shift parameters $s_j \in (-\frac{1}{2}, \frac{1}{2})$ and the complex coefficients c_j in such a way that

$$\left| \tilde{f}_l - \sum_{j=1}^{M} c_j\,\varphi\left(\frac{l}{n}+s_j\right) \right| \le \varepsilon, \quad l = -n/2, \ldots, n/2-1 \tag{6.62}$$

for very small accuracy $\varepsilon > 0$ and for minimal number M of translates. Note that all reconstructed values of the shift parameters s_j, the coefficients c_j, and the number M of translates depend on ε, ε_1, and n. By the assumption $|c_j| \gg \varepsilon_1$, $j = 1, \ldots, M$, we will be able to recover the original integer M in the case of small error bounds ε and ε_1.

This nonlinear inverse problem (6.62) can be numerically solved in two steps. First we convert the given problem (6.62) into a parameter estimation problem for an exponential sum by using Fourier technique. Then the parameters of the transformed exponential sum are recovered by a Prony-like method as introduced in chapter 3.

For the 1–periodic function (6.60), we compute the corresponding Fourier coefficients. By (6.60) we obtain for $k \in \mathbb{Z}$

$$c_k(f) = \int_{-1/2}^{1/2} f(x)\, \mathrm{e}^{-2\pi \mathrm{i} k x}\, \mathrm{d}x = \Big(\sum_{j=1}^{M} c_j\, \mathrm{e}^{2\pi \mathrm{i} k s_j} \Big)\, c_k(\varphi) = h(k)\, c_k(\varphi) \qquad (6.63)$$

with the exponential sum

$$h(x) := \sum_{j=1}^{M} c_j\, \mathrm{e}^{2\pi \mathrm{i} x s_j} \quad (x \in \mathbb{R})\,. \qquad (6.64)$$

In applications, the Fourier coefficients $c_k(\varphi)$ of the window function φ are often explicitly known, where $c_k(\varphi) > 0$, $k = 0, \ldots, N/2$ by assumption. Further the function f is sampled on a fine grid, i.e., we know noisy sampled data $\tilde{f}_l = f(l/n) + e_l$, $l = -n/2, \ldots, n/2-1$ on the fine grid $\{l/n : l = -n/2, \ldots, n/2-1\}$ of $[-1/2,\, 1/2]$, where e_l are small error terms. Then we can compute $c_k(f)$, $k = -N/2, \ldots,\, N/2$ by discrete Fourier transform

$$\begin{aligned} c_k(f) &\approx \frac{1}{n} \sum_{l=-n/2}^{n/2-1} f\left(\frac{l}{n}\right)\, \mathrm{e}^{-2\pi \mathrm{i} k l/n} \\ &\approx \hat{f}_k := \frac{1}{n} \sum_{l=-n/2}^{n/2-1} \tilde{f}_l\, \mathrm{e}^{-2\pi \mathrm{i} k l/n}\,. \end{aligned}$$

For shortness we set

$$\tilde{h}_k := \hat{f}_k / c_k(\varphi), \quad k = -N/2, \ldots, N/2\,. \qquad (6.65)$$

6.1.1 Algorithm

Prony Method for Sums of Translates

Input: $N \in 2\mathbb{N}$, L with $M \le L \le N/2$, $n = \alpha N$ power of 2 with $\alpha > 1$, $\tilde{f}_l = f(l/n) + e_l$, $l = -n/2, \ldots, n/2-1$, $c_k(\varphi) > 0$, $k = 0, \ldots, N/2$ and accuracy $\varepsilon > 0$.

1. By fast Fourier transform compute

$$\begin{aligned} \hat{f}_k &:= \frac{1}{n} \sum_{l=-n/2}^{n/2-1} \tilde{f}_l \, \mathrm{e}^{-2\pi \mathrm{i} k l/n}, \quad k = -N/2, \ldots, N/2\,, \\ \tilde{h}_k &:= \hat{f}_k / c_k(\varphi), \quad k = -N/2, \ldots, N/2\,. \end{aligned}$$

2. Apply algorithm 2.1 or 2.2 to $h_{k+N/2} = \tilde{h}_k$, $k = -N/2, \ldots, N/2$ with threshold ε.

Remark 6.5 *Note that we used the special property of the Fourier transform that shifts in the time domain become modulations in the frequency domain, which leads to a parameter identification problem of the form (6.64). For that reason the approach analyzed here is not transferable to other function systems than the exponential functions.*

In the next subchapter 6.2, we will exemplarily use this approach for nondestructive material testing.

6.2 Nondestructive Material Examination

Recently we considered in [7] the application of Prony's method for nondestructive material examination using the ideas we presented in [36].

Many ultrasonic testing applications are based on the estimation of the time of arrival (TOA), time of flight diffraction (TOFD) or the time difference of arrival (TDOA) of ultrasonic echos. In order to analyze the received signals, one can usually suppose that the diffracted and backscattered echo from an isolated defect is a time-shifted, frequency-dissipated replica of the transmitted pulse with attenuated energy and inverted phase. In case of various flaw defects, the backscattered ultrasonic signal is a convolution of a modification of the transmitted pulse with the reflection centers. Generally, we are faced with noisy measurements caused by reflections on microstructures of the tested material and electronic disturbances. It is therefore desirable to remove the effect of the pulse from the recorded signal, i.e. to perform a deconvolution.

For representation of a received signal $s(t)$, we suppose that it can be obtained as a linear combination of time-shifted, energy-attenuated versions of the transmitted pulse function with inverted phase, where each shift is caused by an isolated flaw scattering the transmitted pulse. Usually, we have only a certain estimate of the transmitted pulse function. Using a similar approach as in [14], we model the pulse echo by a real-valued Gabor function of the form

$$f_\theta(t) = K_\theta \, e^{-\alpha t^2} \cos(\omega t + \phi), \tag{6.66}$$

with the parameters $\theta = (\alpha, \omega, \phi)$. Here, α describes the bandwidth factor, ω is the center frequency, and ϕ the phase of the pulse echo. Because of its Gaussian shape envelope, this model is called Gaussian echo model. These parameters have intuitive meanings for the reflected pulse; the bandwith factor α determines

the bandwith of the echo and hence the time duration of the echo in time domain. The frequency ω is governed by the transducer center frequency.

The normalization factor K_θ is taken such that $\|f_\theta\|_2 = 1$. More precisely, we obtain

$$\begin{aligned} K_\theta^{-2} &= \|e^{-\alpha t^2}\cos(\omega t+\phi)\|_2^2 = \int_{-\infty}^{\infty} e^{-2\alpha t^2}\,\cos^2(\omega t+\phi)\,dt \\ &= \frac{\sqrt{\pi}}{2\sqrt{2\alpha}}(1+\cos(2\phi)e^{-\omega^2/8\alpha}), \end{aligned} \tag{6.67}$$

where we have used that $\int_{-\infty}^{\infty} e^{-2\alpha t^2}\sin(2\omega t)dt = 0$ since the integrand is an odd function. In [14], the feasibility of this model has been demonstrated by a setup for a planar surface reflector using a steel sample, where the experimental echo is fitted by the Gaussian echo.

Let us consider the sparsity model

$$s(t) = \sum_{m=1}^{M} \tilde{a}(m)\, f_\theta(t-\tau_m) + \nu(t), \tag{6.68}$$

with additional noise $\nu(t)$, where we want to optimize over the time shifts $\boldsymbol{\tau} = (\tau_1,\ldots,\tau_M)$ and the amplitudes $\tilde{\mathbf{a}} = (\tilde{a}(1),\ldots,\tilde{a}(M))$ where M is unknown but small. Usually one also wants to optimize over the pulse parameters θ, but for simplicity we assume that we have a good estimate for the parameter vector θ calculated beforehand such that we can concentrate on the computation of $\boldsymbol{\tau}$ and $\tilde{\mathbf{a}}$ from the samples of s. Furthermore, we assume here that we have a suitable bound $\tilde{M} > M$ for the true number of relevant coefficients and can replace M by $\tilde{M}$ in the above model. For that purpose, we now adapt the ideas of calculating shift parameters with Prony-like methods as considered in our paper [36].

Let the Fourier transform of a function $f \in L^1(\mathbb{R})$ be given by

$$\hat{f}(\xi) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)\, e^{-i\xi t}\, dt.$$

Applying the Fourier transform to (6.68) (with M replaced by the bound $\tilde{M} > M$), we find

$$\hat{s}(\xi) = \left(\sum_{m=1}^{\tilde{M}} \tilde{a}(m)\, e^{-i\xi\tau_m} \right) \hat{f}_\theta(\xi) + \hat{\nu}(\xi).$$

In our case, the real-valued Gabor function $f_\theta(t) = K_\theta\, e^{-\alpha t^2}\cos(\omega t+\phi)$ is the real part of $g_\theta(t) = K_\theta e^{-\alpha t^2} e^{i(\omega t+\phi)} = K_\theta\, e^{i\phi} e^{-\alpha t^2} e^{i\omega t}$. We obtain the Fourier transform of g_θ,

$$\hat{g}_\theta(\xi) = \frac{K_\theta e^{i\phi}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\alpha t^2} e^{-it(\xi-\omega)}\, dt = \frac{K_\theta e^{i\phi}}{\sqrt{2\alpha}}\, e^{-(\omega-\xi)^2/4\alpha},$$

and hence

$$\hat{f}_\theta(\xi) = \frac{1}{2}\left(\hat{g}_\theta(\xi) + \hat{\bar{g}}_\theta(\xi)\right) = \frac{K_\theta}{2\sqrt{2\alpha}}\left(e^{i\phi}e^{-(\omega-\xi)^2/4\alpha} + e^{-i\phi}e^{-(\omega+\xi)^2/4\alpha}\right). \tag{6.69}$$

Particularly, the function $\hat{f}_\theta(\xi)$ possesses only a zero at $\xi = 0$ if $\phi = \frac{(2\pi+1)\pi}{2}$ while $\hat{f}(\xi) \neq 0$ for all $\xi \neq 0$. Avoiding the case $\xi = 0$, we can hence write

$$\hat{h}(\xi) := \frac{\hat{s}(\xi)}{\hat{f}_\theta(\xi)} = \sum_{m=1}^{\tilde{M}} \tilde{a}(m)\, e^{-i\xi\tau_m} + \hat{\epsilon}(\xi),$$

where the noise term $\hat{\epsilon}(\xi) := \hat{\nu}(\xi)/\hat{f}_\theta(\xi)$ is assumed to be small.

For given samples $\hat{h}(k\Delta_\xi)$, (where Δ_ξ is a fixed sampling distance) we now aim to compute the frequencies $\tau_m \in \mathbb{R}_+$ and the corresponding amplitudes $\tilde{a}(m)$, for $m = 1, \ldots, \tilde{M}$ separately using algorithm 6.1.1.

In the third step of algorithm 2.1 (or in second step of algorithm 2.2) that is incorporated in 6.1.1, we can compute the amplitudes $\tilde{a}_m$ as least square solution of the overdetermined linear system

$$\sum_{m=1}^{M} \tilde{a}_m\, f_\theta(\ell\Delta_t - \tau_m) = s(\ell\Delta_t), \qquad \ell = 0, \ldots, N,$$

thereby neglecting the noise function $\nu(t)$.

As a test scenario for a back wall deformation we use the pulse function depicted in Figure 16. In Figure 17 we see a corrupted simulation of the detected echo. Here, every column of the picture represents measurements over time at the same spot of the scanning device. The bulge in the center of the picture indicates a back wall deformation, whereas the straight lines at the left and right indicate an intact back wall. We see additional noise at the measured data that we want to eliminate while we still keep the information about the back wall. Finally, in Figure 18 we see location of the found translations, whereas the corresponding amplitudes are color coded.

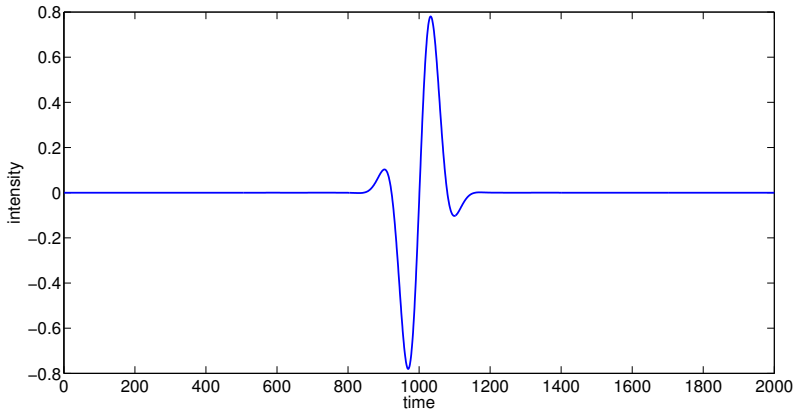

Figure 16: The pulse function used by the ultra sonic scanning devise for non destructive material examination.

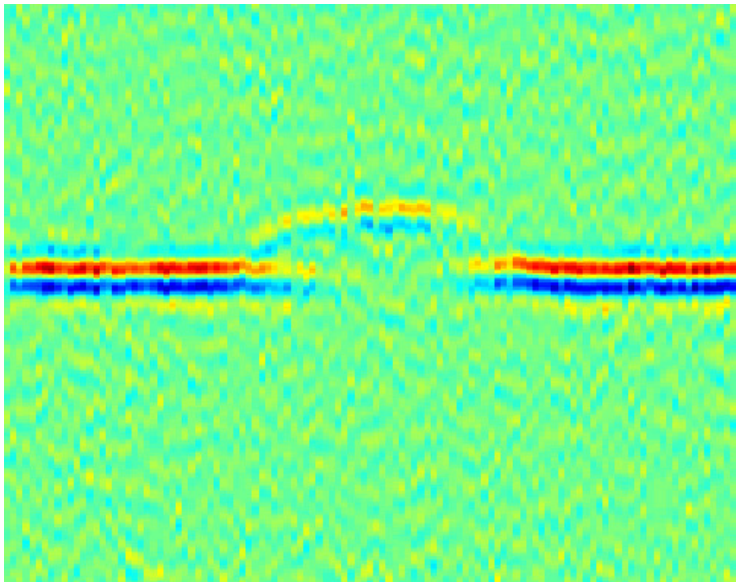

Figure 17: Simulation of perturbed measurements.

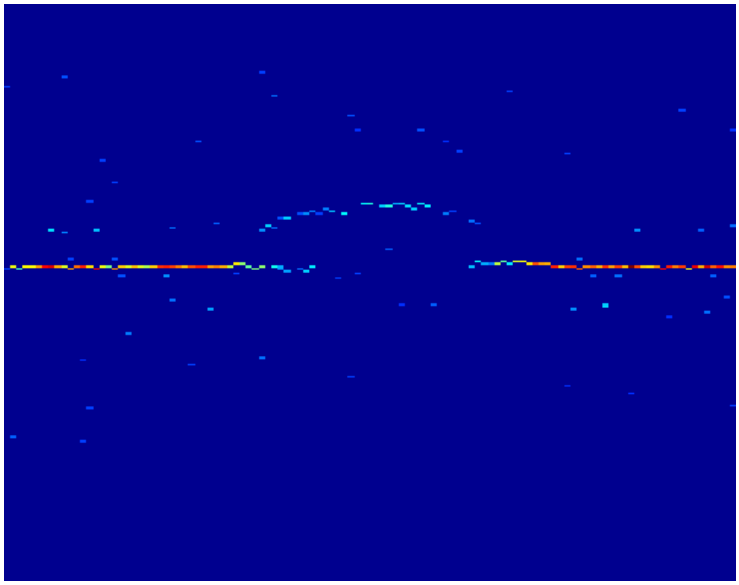

Figure 18: Found translations in Figure 17 of the pulse function depicted in Figure 16.

We see that even though the measurements are heavily corrupted with noise, we still can recover the essential information in Figure 17 via Prony's method.

7 Open Problems

7.1 Denoising with Regularization

For a linear operator $\mathcal{A} : V \to V$ mapping from vector space V to V and a functional $F : V \to \mathbb{C}$ let the function f be as in (1.8). Let $\mathbf{f}, \mathbf{e} \in \mathbb{C}^{2N+1}$, where $\mathbf{f} = (F(\mathcal{A}^k f))_{k=0}^{2N}$ is a signal vector and $\mathbf{e}$ an additive noise vector. Assume that only erroneous input data

$$\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{e}$$

are given. The theory of Prony's method presented in chapter 1 tells us that the Hankel-type matrix in algorithm 2.1 or algorithm 2.2 should be singular. But when we have to handle erroneous input data, the corresponding Hankel-type matrix becomes "more and more regular", i.e. the smallest eigenvalue respectively singular value increases in magnitude. Let us observe the general behavior in an example.

Example 7.1 *Let us observe the behavior of the singular values of the matrix* $\mathbf{H} := (F(\mathcal{A}^{k+m})\tilde{f})_{k,m=0}^{40,10}$ *of the function*

$$f(x) = \sin(1.3x) - 3\cos(x),$$

with the functional $F(f) := f(0)$*, operator* $(Af)(x) := f(x+1)$ *and input data* $F(\mathcal{A}^k f) = f(k)$*,* $k = 0, \ldots, 40$*. We use the* MATLAB *function* `randn` *which generates random numbers with mean* 0 *and variance* 1*. In order to adapt the noise level we multiply the outcome of the random number generator with* $C_s = 10^{-15+s}$*,* $s = 0, \ldots, 15$ *prior to adding the noise to the exact function values. Note that due to the machine accuracy* `eps` *even the putatively exact data have a round of failure in the range of* 10^{-15}*. The PSNR for the observed interval decreases about roughly 20 dB for the transition from* C_s *to* C_{s+1}*,* $s = 0, \ldots, 14$*. Starting with a PSNR of* 310 *for* C_0 *we indeed end up with a PSNR of* 11 *for* C_{15} *in this example. In Figure 19 we have shown the found singular values of algorithm 2.2 for the different noise levels explained above. The color code goes from orange for the multiple* $C_0 = 10^{-15}$ *to blue for multiple* $C_{15} = 1$*. We observe that the noise space is less and less distinguishable from the signal space, the more corrupted the input data get.*

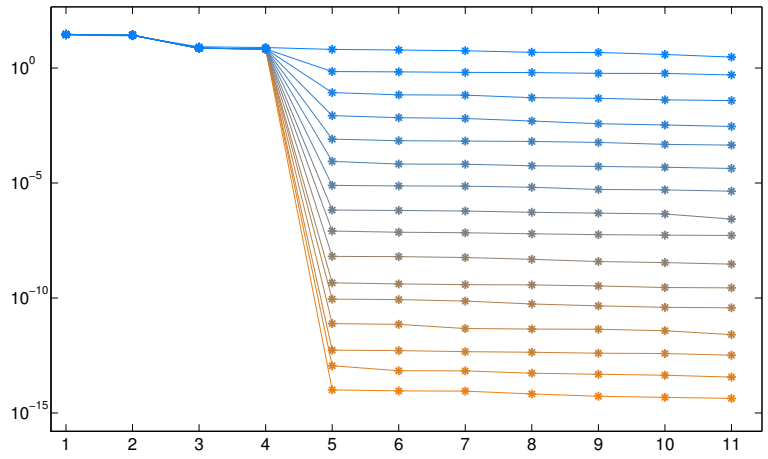

Figure 19: Here, the singularvalues of $\mathbf{H} := (F(\mathcal{A}^{k+m})\tilde{f})_{k,m=0}^{40,10}$ are shown for different noise levels. The noise levels range from 10^{-15} to 1 and are color coded, from orange for 10^{-15} to blue for the noise level 1. The first four singular values corresponding to the signal space remain relatively alike for different noise levels, whereas the seven smallest singular values increase linearly with the corruption level of the input data.

Remark 7.2 *Figure 19 is not an average of the calculated eigenvalues over many perturbed signals and still we observe a compelling correspondence between the noise level and the magnitude of the smallest eigenvalues.*

In example 7.1 we saw a direct correspondence between the noise level the input data are corrupted with and the smallest singular value of the Hankel matrix containing the given data. Since the given data are erroneous, we can try to allow small changes in the given data while we simultaneously aim for a matrix that has a low rank. This is because a low rank matrix in algorithm 2.1 or 2.2 returns a sparse approximation of the given data in the desired function space.

We want to regularize the input data $\tilde{\mathbf{f}}$ prior to applying Prony's method. Define the operator $D_L : \mathbb{C}^{2N} \to \mathbb{C}^{(2N-L)\times(L+1)}$ that constructs a rectangular Hankel-type matrix out of a given vector as

$$
\begin{aligned}
D_L f :=& (F(\mathcal{A}^{k+m})f)_{k,m=0}^{2N-L-1,L} \\
=& \begin{pmatrix} F(\mathcal{A}^0 f) & F(\mathcal{A}^1 f) & \dots & F(\mathcal{A}^L f) \\ F(\mathcal{A}^1 f) & F(\mathcal{A}^2 f) & \dots & F(\mathcal{A}^{L+1} f) \\ \vdots & \vdots & & \vdots \\ F(\mathcal{A}^{2N-L} f) & F(\mathcal{A}^{2N-L+1} f) & \dots & F(\mathcal{A}^{2N-1} f) \end{pmatrix}.
\end{aligned}
$$

Let $\mathbf{G} \in \mathbb{C}^{(2N-L)\times(L+1)}$. We now define the minimization problem

$$f_0 = \underset{\mathbf{G}}{\operatorname{argmin}} \left[\|\mathbf{G} - D_L \tilde{f}\|_{\mathrm{F}}^2 + \alpha \operatorname{rank}(\mathbf{G}) \right].$$

In the first term we allow small changes of the input data, with respect to the Frobenius norm, which is reasonable for erroneous input data. In the second term we want the Hankel-type matrix $\mathbf{G}$ to have a small rank. These two constrictions are balanced by the regularization parameter α. Problematic in this setting is the constraint that the rank of $\mathbf{G}$ should be small, hence this minimization problem is again NP hard. Literature contributing to this topic include the paper [47].

7.2 Prony Method for Not Exactly Sparse Functions

In this thesis we have always assumed that the signal we investigate is M-sparse in a certain function space. What happens, if the signal is not exactly M-sparse but only has coefficients c_j that decay "fast enough"? Let V be a vector space and $\mathcal{A}$ a linear operator with eigenfunctions v_j to eigenvalues λ_j, $j \in J$. Consider the function

$$f = \sum_{j=1}^{\infty} c_j v_j,$$

with, for example $\mathbf{c} := (c_j)_{j=1}^{\infty} \in \ell_1$. One important question that we have to answer in this setting is the following. What sufficient conditions does the coefficient vector $\mathbf{c}$ have to fulfill, to guarantee recovery of the active eigenfunctions v_j corresponding to coefficients with large enough absolute value $|c_j| > \varepsilon$, via our generalized Prony Method? Let us split the set of summands in f into two sets

$$f(x) = f_s(x) + f_n(x) = \sum_{j \in J_s} c_j v_j + \sum_{m \in J_n} c_m v_m,$$

with $J_s = \{j \,|\, |c_j| > \varepsilon\}$ and $J_n = \{m || c_m| \leq \varepsilon\}$. Can we answer the previous question by pretending that the summands $c_m v_m$, $m \in J_m$ are (colored) additional noise to the essential signal $f_s(x)$? If that is the case, we can use our generalized Prony Method to approximate functions in the function space V with just M terms. Again, a special case would be that $V = C(\mathbb{R})$ and $\mathcal{A}$ is the translation operator such that we have exponential functions as eigenfunctions of $\mathcal{A}$. Then we can approximate special functions with fast decaying Fourier coefficients with an a priori defined accuracy δ and only M terms. The word "special" has to be specified too. The hope is that by using only M terms calculated via Prony's method, we still get an approximation $\tilde{f}$ to the signal f with $\|f - \tilde{f}\|_2 < \delta$, whereas the approximation via, for example the DFT, needs $L > M$ terms to achieve the same accuracy. This hope is motivated by the fact that the eigenvalues (frequencies in this case) corresponding to the active eigenfunctions in the approximation $\tilde{f}$ are not restricted to lie on a grid, as they are, when calculated via the DFT.

7.3 Negative Results

This section is dedicated to ideas that turned out to be inapplicable. Often, when one is confronted with a new topic, one has immediate crosslinks to other topics, wondering if they can be combined. Sadly, it is seldom documented if such a link does not turn out to work properly. Here we want to present some of those routes with a certain expectation that did not help to improve the results.

7.3.1 Approximating Derivatives of Orthogonal Polynomials

Theorem 4.17 showed that we can relax the restrictive requirement of having input data of the form $L_{p,q}^k f(x)$ to derivatives as input data, when analyzing orthogonal polynomials. In general this doubles the number of input data. But even derivatives might not be available in some applications. At that point the idea arose that we could approximate the needed derivatives out of for instance equidistant sampling points using Taylor expansion

$$f(x+kh) = f(x) + \sum_{\ell=1}^{\infty} \frac{f^{(\ell)}(x)(kh)^\ell}{\ell!}, \quad h > 0, k \in \mathbb{Z}. \tag{7.70}$$

A common way to generate approximation formulas of the n-th derivative of $f(x)$ at the point x_0 with order d uses combinations of (7.70) for different $k \in \mathbb{Z}$.

Example 7.3

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + f''(x)h^2 + \mathcal{O}(h^3) \\ f(x-h) &= f(x) - f'(x)h + f''(x)h^2 + \mathcal{O}(h^3) \\ f(x+h) - f(x-h) &= 2f'(x)h + \mathcal{O}(h^3) \\ f'(x) &= \frac{1}{2h} f(x+h) - \frac{1}{2h} f(x-h) + \mathcal{O}(h^2) \end{aligned}$$

If equidistant sampling points $f(x_0 + mh)$, $m = 0, \ldots, N$ of a function $f(x)$ are given, the formula above tells us that we can approximate the first derivative of $f(x_0 + mh)$ as

$$f'(x+mh) \approx \frac{1}{2h} f(x+(m+1)h) - \frac{1}{2h} f(x+(m-1)h), \quad m = 1, \ldots, N-1$$

with 2-nd order precision in stepwidth h.

We can combine more Taylor expansions around an evaluation point $f(x+mh)$ in order to get approximation formulas for higher order derivatives and higher order approximations. The hope is that we compute the first $4M$ derivatives of a function $f(x)$ at point $x_1 = x_0 + mh$ for some m, out of given equidistant sampling points such that we can compute the values $F(L_{p,q} f(x_1))$ afterwards. Unfortunately, this approach is not stable enough for practical use. The coefficients of the high order derivative approximations have a too large variety in magnitude and inherit cancelation such that, even in the noiseless case, an approximation with higher orders does not provide better results.

Example 7.4 *Let*

$$f(x) = -P_{10}(x) + 3P_5(x) + 2P_2(x) \tag{7.71}$$

be a sparse linear combination of Legendre polynomials with given equidistant sampling points $f(-1+kh)$, $h = 0.05$, $k = 0, \dots, 40$. *From those points we want to evaluate the first derivatives of* f *at the point* $x_0 = 0.1$. *The exact derivatives are presented in Table 13.*

k	$f^{(k)}(0.1)$
0	−0.311388752612890
1	3.190725211367187
2	−23.30322441445312
3	973.9938121875002
4	1902.802997812500
5	−19685.75450625000
6	−169957.6003125000
7	1613849.737500000
8	13956067.12500000
9	−65472907.50000000
10	−654729075.0000000

Table 13: Accurate first 10 derivatives of the function f in (7.71).

Note that all further derivatives are zero because $f(x)$ *is a* 10*-th order polynomial. The polynomial (7.71) is* 3*-sparse in the Legendre basis. Thus, Theorem 1.2 tells us that we need at least* 6 *input data of the form* $F(L^k_{p,q} f)$, $k = 0, \dots, 5$. *Here we use algorithm 2.2 which needs an additional input value. Furthermore, we oversample slightly and use a total of* 9 *input values of the form* $F(L^k_{p,q} f)$, $k = 0, \dots, 8$, *which leads to a Hankel-matrix* $\mathbf{H} = F(L^{k+m}_{p,q} f)^4_{k,m=0} \in \mathbb{R}^{5\times 5}$. *In Theorem 4.17 we saw, that we can evaluate the values* $F(L^k_{p,q} f)$ *out of the first* $2k$ *derivatives if the evaluation point is not* 1 *or* -1. *Alltogether we need the value* $f(0.1)$ *and the first* 17 *derivatives at the point* 0.1. *In figure 20 we have shown the absolute difference of the correct derivatives and the approximated derivatives in dependence of the order of the used Taylor expansion. The orders are color-coded, where the failure made by using the second order Taylor expansion is depicted in blue and the* 20*-th order in bright green. Only odd orders are used. We see a similar increase of the made failure of approximating derivatives, after the ninth derivative. Note that the derivatives and therefore also the failures keep increasing after the* 10*-th derivative even though they should be zero. In Figure 20 we used a logarithmic scale for the vertical axis.*

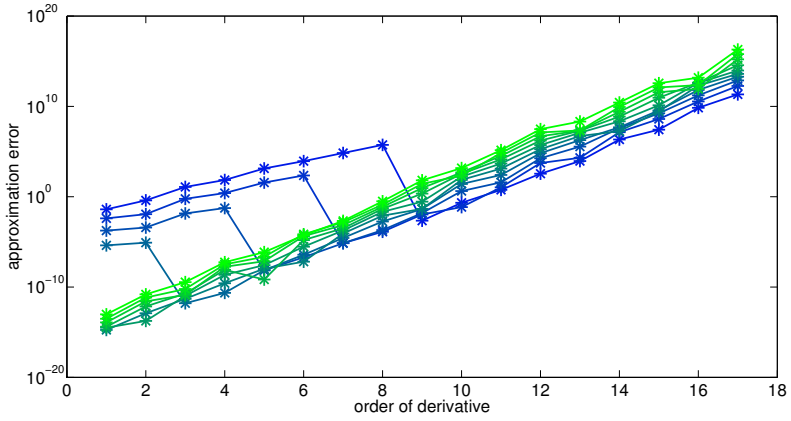

Figure 20: Absolute values of the differences between the actual derivatives $f^{(k)}(0.1)$, $k = 0, \ldots, 17$ and the approximated derivatives. The different used orders for the Taylor approximation are color-coded, with the second order approximation depicted in blue and the 20-th order approximation depicted in bright green.

Note that the value $f(0.1)$ does not have to be approximated. That is why we do not observe a difference in Figure 20 at the point zero.

Indeed, the algorithm fails to detect all three active Legendre polynomials correctly, for any approximation order. Instead of $P_2(x)$ the method finds $P_3(x)$. Since we cannot detect the active eigenfunctions in f accurately in this simple example, we have to discard the approach of approximating derivatives of a polynomial out of equidistant sampling points via Taylor approximation as a suitable preprocessing tool for our generalized Prony Method applied to the Sturm-Liouville operator.

Conclusion

In this thesis we have generalized the classical Prony Method that works for exponential functions to eigenfunctions of linear operators. This meta method enables us not only to develop Prony Methods for other function systems than the exponential functions, but also to further generalize the classical Prony Method due to the freedom of choice in the functional F. We have shown examples of different linear operators and their eigenfunctions, the impact of the functional F to the method and possible applications of the method. Furthermore we generalized numerical implementations of the classical Prony Method to our generalized Prony Method and performed comparative tests. Finally we gave an outlook to future spheres of activities.

References

[1] R. G. Baraniuk, A lecture on compressed sensing, IEEE Signal Proc. Magazine, vol. 24, No. 4, July, pp. 118-121, (2007).

[2] D. Batenkov and Y. Yomdin, On the accuracy of solving confluent Prony systems, preprint (2012).

[3] M. Ben-Or, P. Tiwari, A deterministic algorithm for sparse multivariate polynomial interpolation, *Proc. Twentieth Annual ACM Symp. Theory Comput.*, New York, N.Y., ACM Press, pp. 301 - 309, (1988).

[4] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, M. J. Strauss, Combining geometry and combinatorics: A unified approach to sparse signal recovery. In Communication, Control, and Computing, 46th Annual Allerton Conference on (pp. 798-805), IEEE (2008).

[5] G. Beylkin. On the fast Fourier transform of functions with singularities. Appl. Comput. Harmon. Anal., 2, pp. 363 – 381, (1995).

[6] G. Beylkin, R. Cramer, G. I. Fann, R. J. Harrison, Multiresolution separated representations of singular and weakly singular operators, *Appl. Comput. Harmon. Anal.* **23**, 235–253, (2007).

[7] F. Boßmann, G. Plonka, T. Peter, O. Nemitz, T. Schmitte, Sparse deconvolution methods for ultrasonic NDT, *Journal of Nondestructive Evaluation* **31**(3), pp. 225–244, (2012).

[8] L. Brand, The companion matrix and its properties, *Amer. Math. Monthly* Vol. **71**, No. 6, June (1964)

[9] E. J. Candès, J. Romberg. Sparsity and incoherence in compressive sampling. Inverse Problems, **23**, pp. 969-985, (2007).

[10] E. J. Candès, M. Wakin, An introduction to compresssive sampling, IEEE signal proc. magazin, vol **25**, No. 2, pp. 21-30, March (2008).

[11] E. J. Candès, The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique, **346**(9), pp. 589-592, (2008)

[12] E. J. Candès, C. Fernandez-Granda, Towards a mathematical theory of super-resolution, *Comm. Pure Appl. Math.*, to appear.

[13] T. Corner, C. Leierson, R. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, Cambridge Massachusetts, (2001)

[14] S. Demirli, J. Saniie, Model-based estimation of ultrasonic echos. Part I: Analysis and algorithms, *IEEE Trans. Ultrason. Ferroelect. Freq. Control* **48**(3), pp. 787–802, (2001).

[15] D. L. Donoho, Compressed Sensing, IEEE Transactions on information theory, vol. 52, No. 4, pp.1289 April (2006).

[16] A. Dress, J. Grabmeier, The interpolation problem for k-sparse polynomials and character sums, *Advances in Appl. Math.* **12**, pp. 57–75, (1991).

[17] A. J. W. Duijndam and M. A. Schonewille. Nonuniform fast Fourier transform. Geophysics, **64**, pp. 539 – 551, (1999).

[18] A. Dutt and V. Rokhlin, Fast Fourier transforms for nonequispaced data. SIAM J. Sci. Stat. Comput., **14**, pp. 1368 – 1393, (1993).

[19] F. Filbir, H. Mhaskar, J. Prestin, On the problem of parameter estimation in exponential sums, *Constr. Approx.* **35**, pp. 323–343, (2012).

[20] S. Foucart, H. Rauhut, Restricted Isometry Property. In A Mathematical Introduction to Compressive Sensing pp. 133-174, Springer New York, (2013).

[21] Gelfond, Alexander Osipovich, Differenzenrechnung, VEB Deutscher Verlag der Wissenschaften Berlin, (1958).

[22] M. Giesbrecht, G. Labahn, W. Lee, On the Equivalence Between Prony's and Ben-Or's/Tiwari's Methods, Unversity of Waterloo Tech Report, **23**, (2002).

[23] M. Giesbrecht, G. Labahn, W. Lee, Symbolic-numeric sparse interpolation of multivariate polynomials, J. Symb. Comput. **44**(8), pp. 943–959, (2009).

[24] G. Golub, C. van Loan, Matrix Computations, The John Hopkins University Press, Baltimore (1983).

[25] G. Golub, P. Milanfar, J. Varah, A stable numerical method for inverting shape from moments. SIAM Journal on Scientific Computing, **21**(4), pp. 1222-1243, (1999).

[26] D. Grigoriev, M. Karpinski, M. Singer, The interpolation problem for k-sparse sums of eigenfunctions of operators, *Advances in Appl. Math.* **12**, pp. 76-81, (1991).

[27] M. Hanke, One shot inverse scattering via rational approximation, *SIAM J. Imaging Science* **5**(1), pp. 465Ð-482, (2012).

[28] Y. Hua, T. K. Sarkar, Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise, IEEE Trans. Acoust. Speech Signal Process. **38**, pp. 814 - 824, (1990).

[29] J. I. Jackson, C. H. Meyer, D. G. Nishimura, A. Macovski. Selection of a convolution function for Fourier inversion using gridding. IEEE Trans. Med. Imag., **10**, pp. 473 – 478, (1991).

[30] E. Kaltofen and W.-s. Lee, Early termination in sparse interpolation algorithms, J. Symb. Comput. **36**, pp. 365–400, (2003).

[31] J. Keiner, S. Kunis, D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. ACM Trans. Math. Software, **36**, Article 19, pp. 1 – 30, (2009).

[32] Y. N. Lakshman, B. D. Saunders, Sparse polynomial interpolation in non-standard bases., SIAM J. Comput. **24**, pp. 387 - 397, (1995).

[33] C. Lanczos, Applied Analysis, Sir Isaac Pitman & Sons, (1957).

[34] T. Peter, G. Plonka, A Generalized Prony Method for Reconstruction of Sparse Sums of Eigenfunctions of Linear Operators, Inverse Problems **29**, no. 2, (2013).

[35] T. Peter, G. Plonka, R. Rosça, Representation of Sparse Legendre Expansions, Journal of Symbolic Computation, (2012).

[36] T. Peter, D. Potts, M.Tasche, Nonlinear approximation by sums of exponentials and translates, *SIAM J. Sci. Comput.* **33**(4), pp. 1920–1944, (2011).

[37] V. F. Pisarenko, The Retrieval of Harmonics from a Covariance Function. Geophysical Journal of the Royal Astronomical Society, **33**, pp. 347Ð366, (1973).

[38] G. Plonka and M. Wischerhoff, How many Fourier samples are needed for real function reconstruction?,*J. Appl. Math. Comput.*, to appear.

[39] D. Potts, M. Tasche, Parameter estimation for exponential sums by approximate Prony method, Signal Process. **90**, pp. 1631 - 1642, (2010).

[40] D. Potts, M. Tasche, Nonlinear approximation by sums of nonincreasing exponentials, *Appl. Anal.* **90**, pp. 609–626, (2011).

[41] D. Potts, M. Tasche, Parameter estimation for multivariate exponential sums, preprint, Technical University Chemnitz, (2012).

[42] D. Potts, M. Tasche, Sparse polynomial interpolation in Chebyshev bases, preprint, Technical University Chemnitz, (2012).

[43] D. Potts, M. Tasche, Parameter estimation for nonincreasing exponential sums by Prony-like methods, *Linear Algebra and its Applications*, (2012).

[44] D. Potts, M. Tasche, Reconstruction of sparse Legendre and Gegenbauer expansions, preprint (2013).

[45] Baron de Prony, Gaspard-Clair-François-Marie Riche, 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. de l'École Polytechnique, **1**, pp. 24–76, (1795).

[46] B.D. Rao, K. V. S. Hari, Performance analysis of root-MUSIC. Acoustics, Speech and Signal Processing, IEEE Transactions on, **37** (12), pp. 1939-1949, (1989).

[47] B. Recht, M. Fazel, P. Parrilo, Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization, SIAM review **52**, no. 3, pp. 471-501, (2010).

[48] R. Roy, T. Kailath, ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques, IEEE Transactions on Acoustic Speech. and Signal Processing, **37**, No. 7, July (1989).

[49] R. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Trans. on Antennas and Propagation, **34**, No. 3, pp. 276 - 280, March (1986).

[50] F. Schöpfer, F. Binder, A. Wöstehoff, T. Schuster, S. von Ende, S. Föll, R. Lammering, Accurate determination of dispersion curves of guided waves in plates by applying the matrix pencil method to laser vibrometer measurement data, *Technische Mechanik* 2012, to appear.

[51] R. Sérul, Programming for Mathematicians, Springer (2000)

[52] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. Adv. Comput. Math., **9**, pp. 337 – 353, (1998).

[53] P. Stoica, R. Moses, Introduction to Spectral analysis, Prentice Hall Upper Saddle River, NJ 07458, (1997).

[54] M. Vetterli, P. Marziliano, and T. Blu, Sampling signals with finite rate of innovation, *IEEE Transactions on Signal Processing* **50**, pp. 1417 - 1428, June (2002).

[55] T. Yanai, G. I. Fann, Z. Gan, R.J. Harrison, G. Beylkin, Multiresolution quantum chemistry: Analytic derivatives for Hartree-Fock and density functional theory, *J. Chem. Phys.* **121**(7), pp. 2866–2876, (2004).

[56] R.E. Zippel, Probabilistic algorithms for sparse polynomials, Ph.D, Thesis. Massachusetts Institute of Technology, Cambridge, USA, (1979).

Curriculum Vitae

Dipl.-Math. Thomas Peter

Address	Institute for Numerical and Applied Mathematics University of Göttingen Lotzestraße 16-18 37083 Göttingen / Germany
Tel.: (office)	+49/551/394515
Email:	t.peter@math.uni-goettingen.de
Homepage:	http://na.math.uni-goettingen.de/index.php?section=peter

---

Personal Details

Gender	Male
Date of birth	13th of July, 1982
Place of birth	Rostock, Germany
Citizenship	German

Academic Education

07/2010-09/2013	Ph.D. student of mathematics at the University of Göttingen (Germany). Supervisor: G. Plonka-Hoch.
10/2002-04/2010	Student of mathematics (for diploma) at the University of Rostock (Germany), with specialization in numerics.
01/2007-07-2007	Exchange student of mathematics at the University of Lund (Sweden).
10/2004-09/2007	Student of informatics (for diploma) at the University of Rostock (Germany).

Civillian Service

09/2001-06/2002	Employed at the Environmental Agency in Rostock.

Schooling

09/1996-07/2001	Secondary school 'Christophorus-Gymnasium' in Rostock (Germany).
09/1993-07/1996	Secondary school 'Goethe-Gymnasium' in Rostock (Germany).

Research Experience

Since 07/2010	Scholarship holder of the Research Training Group 1023: Identification in Mathematical Models at the Institute for Numerical and Applied Mathematics, University of Göttingen (Germany).

Teaching Experience

Numerics II, summer term 2011, tutorial. Institute for Numerical and Applied Mathematics, University of Göttingen (Germany).

Methods in signal and image processing, summer term 2011, tutorial. Institute for Numerical and Applied Mathematics, University of Göttingen (Germany).

Numerics I, winter term 2010/2011, tutorial. Institute for Numerical and Applied Mathematics, University of Göttingen (Germany).

Publications

[1] T. Peter, D. Potts, M. Tasche, Nonlinear approximation by sums of exponentials and translates, SIAM J. Sci. Comput. **33**, pp. 1920-1947, (2011).

[2] F. Boßmann, G. Plonka, T. Peter, O. Nemitz, T. Schmitte, Sparse deconvolution methods for ultrasonic NDT, Journal of Nondestructive Evaluation **31**(3), pp. 225-244, (2012).

[3] T. Peter, G. Plonka, D. Rosça, Representation of sparse Legendre expansions, Journal of Symbolic Computation, **50**, pp. 159-169, (2013).

[4] T. Peter, G. Plonka, A generalized Prony method for reconstruction of sparse sums of eigenfunctions of linear operators, Inverse Problems **29**, (2013).

Participation in Conferences and Workshops

2013	Colloquium (talk), June 27 at MIC in Lübeck
	Mecklenburger Workshop über Approximationsmethoden und schnelle Algorithmen (talk), June 20-23 in Hasenwinkel
	Workshop: Structured Function Systems and Applications February 24- March 02 in Oberwolfach

	23. Rhein-Ruhr-Workshop (poster) February 01-02 in Bestwig
2012	Advances in Mathematical Image Processing (talk), September 04-06 in Göttingen
	Colloquium (talk), June 27 at the Institute for Mathematics in Lübeck
	33. Norddeutsches Kolloquium über Angewandte Analysis und Numerische Mathematik, May 04-05 in Rostock
	GAMM Annual Meeting 2012 (talk), March 26-30 in Darmstadt
	22. Rhein-Ruhr-Workshop (talk), February 03-04 in Bestwig
2011	21. Rhein-Ruhr-Workshop (talk), February 04-05 in Königswinter