# Reconstruction of polygonal shapes from sparse Fourier samples

Marius Wischerhoff* Gerlind Plonka†

16th June 2015

In this paper, we want to reconstruct polygonal shapes in the real plane from as few Fourier samples as possible, that is, we want to recover an original polygonal domain $D$ with $N$ vertices by using sparse sampling values of the Fourier transform of the characteristic function of the polygonal domain. We consider only simply-connected polygons, i.e. polygons with non-intersecting edges. For this purpose, we need to reconstruct the vertices of the polygon. In the case of non-convex polygons, we also need to reconstruct the order of the vertices to determine the correct boundary line segments. The method presented here is based on the Prony method.

**Key words**: polygonal shape reconstruction, sparse Fourier reconstruction, Prony method, unit-height polygons

**Mathematics Subject Classification**: 94A12, 42B10, 65D20, 65H10, 41A63

## 1 Introduction

In several scientific areas, such as radio astronomy, computed tomography, and magnetic resonance imaging, [1], the reconstruction of structured functions from the knowledge of samples of their Fourier transform is a common problem. For the analysis of the examined object, it is important to reconstruct the underlying original signal as exactly as possible. In this paper,

*Institute for Numerical and Applied Mathematics, Göttingen University, Lotzestr. 16-18, 37083 Göttingen, Germany. Email: m.wischerhoff@math.uni-goettingen.de

†Institute for Numerical and Applied Mathematics, Göttingen University, Lotzestr. 16-18, 37083 Göttingen, Germany. Email: plonka@math.uni-goettingen.de

we aim at unique recovery of polygonal domains in the real plane from a smallest possible set of Fourier data.

One main ingredient of our reconstruction approach is the *Prony method*, [12, pp. 457–462]. Within the past few years, the Prony method has been increasingly applied in the field of parameter estimation. It enables the determination of exponential sums using only few sampling values.

Several numerically stable variants of the original Prony method, [5], have been derived. Potts and Tasche have developed the *Approximate Prony method* in [21, 22], which works also in the case of noisy measurements. Another stabilization of the Prony method is proposed in [8], where the possibly perturbed sampling values are not used directly, but a windowed average of their autocorrelation sequence is used instead. This approach is motivated by the application of operators of the form $\sum_{k=0}^{N} g\left(\frac{k}{N}\right) \widehat{f}(k) \exp(\mathrm{i}\, k\bullet)$ with a suitable filter $g$ in [15] and [16], where such operators have been applied in order to detect singularities of piecewise smooth functions $f$.

There are also well-known parameter identification methods in signal processing, such as the *ESPRIT method*, [25], the *Matrix pencil method*, [13], and the *MUSIC method*, [26]. In [24], it has been pointed out that these methods are equivalent to the Prony method such that they can be seen as so-called *Prony-like methods*, see also [18, Chapter 3].

The problem of reconstructing polygonal shapes in the real plane from as few Fourier samples as possible using the Prony method has been briefly introduced in [27]. The reconstruction is based on the observation that the Fourier transform of the characteristic function $\mathbf{1}_D$ of a polygonal domain $D$ can be represented as an exponential sum where the exponential parameters (frequencies) contain the information on the vertices while the corresponding coefficients cover further information on the slopes of the edges of $D$. In the paper on hand, we will examine this problem in detail, and we will show that we are able to reconstruct convex and non-convex polygonal domains $D$ with $N$ vertices by taking only $3N$ samples of the Fourier transform of $\mathbf{1}_D$ on three lines through the origin. For this purpose, we use an adaptive sampling scheme. We consider two predetermined lines, while the third sampling line is chosen dependently on the results obtained by employing the samples from the first two lines. The adaptive sampling idea has also been applied in our recent paper [20], where we presented a reconstruction method for sparse expansions of non-uniform translates of a known low-pass filter function from a small number of Fourier samples in the bivariate case. This approach can also be generalized to $d$-variate functions with $d > 2$. A similar idea of transferring a multidimensional problem into several one-dimensional problems by considering different sampling lines has been pursued in [23]. But there, in

contrast to our reconstructing scheme with a fixed, small number of sampling lines, it is not directly clear how many lines are needed since they are not chosen adaptively. Due to the solvability of a linear system, it is decided if further sampling liens are needed.

Considering convex polygons, it is sufficient to know the vertices in order to obtain a reconstruction of the polygon. But in the case of non-convex polygons with at least four vertices, there are always several distinct polygons which have the same vertices. Thus, we have to reconstruct the order of the vertices in order to know which vertices are connected by line segments. We use an approach consisting of two steps. First, we compute the vertices by applying the Prony method to Fourier samples taken on three lines through the origin in the frequency domain. The obtained sparse exponential sum provides not only the vertices (as exponential parameters) but also the corresponding coefficients which we use in the second step of our approach to determine the order of the vertices.

The problem of reconstructing polygons from some given data has also been examined in [7, 9, 17]. In these papers, the given data are complex moments, i.e. integral moments of the analytic power function $f(z) = z^k$ over the characteristic function of the polygonal domain $D$ with $z = x + \mathrm{i}\, y$, $x, y \in \mathbb{R}$. The given data are of the form

$$k(k-1) \int_D z^{k-2} \,\mathrm{d}(x,y) = \sum_{j=1}^{N} a_j z_j^k, \quad k = 0, \ldots, 2N-1, \tag{1}$$

where $z_j$ are the vertices of the polygon in the complex plane, and Prony-like methods are employed in order to determine the vertices. In [9, 17], it is discussed that, even if besides the vertices also the coefficients $a_j$ in (1) are given, the interior of the polygon is not always uniquely determined. Also in cases of a unique solution, the step of ordering the vertices to determine the interior of the polygon yields the problem of deciding on the right configuration of up to at most $2^{N-1}$ possible scenarios for the sides of the polygon, see [9, Subsection 3.2], that is, it leads to a problem in computational geometry and graph theory, see [6]. In contrast to these approaches, we will present an iterative algorithm to determine the correct order of the vertices of $D$.

Finally, we want to mention that there are approaches to reconstruct convex polytopes from moments, [3, 10]. The paper [4] describes an idea to reconstruct more general shapes of compact objects from moments.

This paper is organized as follows: Section 2 provides an overview of the Prony method in such a formulation as we need later on in Section 3 for

the reconstruction of polygonal shapes in the real plane from sparse Fourier data. We illustrate the proposed reconstruction method with some numerical experiments in Section 4 showing the applicability for exact input data.

## 2 Prony method

We briefly summarize the Prony method, which we will apply in the form presented below.

Let us consider an exponential sum where the coefficients are real numbers, i.e. a trigonometric function $P : \mathbb{R} \to \mathbb{C}$ of the form

$$P(\omega) = \sum_{j=1}^{N} c_j \, \mathrm{e}^{-\mathrm{i}\,\omega T_j} \tag{2}$$

with $N \in \mathbb{N}$, non-zero coefficients $c_j \in \mathbb{R}$, and real-valued frequencies $T_j$ in ascending order, i.e. $-\infty < T_1 < T_2 < \ldots < T_N < \infty$.

We want to compute the frequencies $T_1, \ldots, T_N$ as well as the coefficients $c_1, \ldots, c_N$ from the sampling values $P(\ell h)$ for $\ell = 0, \ldots, N$ where $h$ is assumed to be a positive constant with $hT_j \in (-\pi, \pi]$ for all $j \in \{1, \ldots, N\}$. For this purpose, the Prony method can be applied as follows.

First, we consider the so-called Prony polynomial $\Lambda : \mathbb{C} \to \mathbb{C}$ defined by

$$\Lambda(z) := \prod_{j=1}^{N} \left( z - \mathrm{e}^{-\mathrm{i}\,hT_j} \right) = \sum_{\ell=0}^{N} \lambda_\ell z^\ell, \tag{3}$$

which possesses the values $\mathrm{e}^{-\mathrm{i}\,hT_j}$ for $j = 1, \ldots, N$ with the unknown frequencies $T_j$ from (2) as zeros. The leading coefficient of $\Lambda$ in the monomial representation on the right-hand side of (3) is given as $\lambda_N = 1$ by definition of $\Lambda$. Then we obtain the following for $m = 0, \ldots, N$:

$$\begin{aligned} \sum_{\ell=0}^{N} \lambda_\ell P\big(h(\ell - m)\big) &= \sum_{\ell=0}^{N} \lambda_\ell \sum_{j=1}^{N} c_j \, \mathrm{e}^{-\mathrm{i}\,h(\ell-m)T_j} \\ &= \sum_{j=1}^{N} c_j \, \mathrm{e}^{\mathrm{i}\,hmT_j} \sum_{\ell=0}^{N} \lambda_\ell \, \mathrm{e}^{-\mathrm{i}\,h\ell T_j} \\ &= \sum_{j=1}^{N} c_j \, \mathrm{e}^{\mathrm{i}\,hmT_j} \, \Lambda(\mathrm{e}^{-\mathrm{i}\,hT_j}) \overset{(3)}{=} 0, \end{aligned} \tag{4}$$

which means that the coefficient vector $\boldsymbol{\lambda} := (\lambda_0, \ldots, \lambda_N)^{\mathrm{T}}$ is the solution of the linear system

$$\boldsymbol{T}_{N+1}\boldsymbol{\lambda} = \mathbf{0} \tag{5}$$

with the Toeplitz matrix

$$\boldsymbol{T}_{N+1} := \Big(P\big(h(\ell - m)\big)\Big)_{m,\ell=0}^{N} \in \mathbb{C}^{(N+1)\times(N+1)}.$$

All $(N^2 + 2N + 1)$ entries of $\boldsymbol{T}_{N+1}$ are given by the $2N + 1$ function values $P(\ell h)$, $\ell = -N, \ldots, -1, 0, 1, \ldots, N$. Since

$$P(-\ell h) = \overline{P(\ell h)}, \quad \ell = 1, \ldots, N, \tag{6}$$

the Toeplitz matrix $\boldsymbol{T}_{N+1}$ is completely determined by the sampling values $P(\ell h)$, $\ell = 0, \ldots, N$.

The further approach is based on matrix analysis. With the Vandermonde matrix

$$\boldsymbol{V}_{N,N+1} := \big(\exp(-\mathrm{i}\, hkT_j)\big)_{j=1,k=0}^{N},$$

we find

$$\boldsymbol{T}_{N+1} = \boldsymbol{V}_{N,N+1}^{*} \cdot \operatorname{diag}(c_1, c_2, \ldots, c_N) \cdot \boldsymbol{V}_{N,N+1} \quad \text{and} \quad \operatorname{rank}(\boldsymbol{T}_{N+1}) = N.$$

Thus, the eigenvector $\boldsymbol{\lambda}$ of $\boldsymbol{T}_{N+1}$ corresponding to the eigenvalue 0 is uniquely determined by (5) and $\lambda_N = 1$.

The next step consists of the computation of the zeros $z_j := \mathrm{e}^{-\mathrm{i}\, hT_j}$, $j = 1, \ldots, N$, of the polynomial $\Lambda$, which is determined by the coefficient vector $\boldsymbol{\lambda}$ of $\Lambda$. Using the assumptions about $h$ (see also Remarks 2.2, 1.) and the principal branch of the complex logarithm, we get the frequencies $T_1, \ldots, T_N$ by the evaluation $T_j = -\operatorname{Im}\big(\ln(z_j)\big)/h$ for $j = 1, \ldots, N$.

Finally, we obtain the coefficients $c_j$, $j = 1, \ldots, N$, from the linear Vandermonde-type system

$$P(\ell h) = \sum_{j=1}^{N} c_j\, \mathrm{e}^{-\mathrm{i}\, \ell h T_j}, \quad \ell = 0, \ldots, N.$$

In summary, we have the following algorithm:

**Algorithm 2.1.**

- **Input:**
  - $P(\ell h)$, $\ell = 0, \ldots, N$;
  - step size $h$ with $hT_j \in (-\pi, \pi]$ for all $j \in \{1, \ldots, N\}$.

- **Computation:**
  1. Compute $P(-\ell h)$, $\ell = 1, \dots, N$, using (6).
  2. Construct the Toeplitz matrix $\boldsymbol{T}_{N+1} := \Big(P\big(h(\ell - m)\big)\Big)_{m,\ell=0}^{N}$.
  3. Solve the system $\boldsymbol{T}_{N+1}\boldsymbol{\lambda} = \boldsymbol{0}$ where $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \dots, \lambda_{N-1}, 1)^{\mathrm{T}}$.
  4. Consider the polynomial $\Lambda(z) := \sum_{\ell=0}^{N} \lambda_\ell z^\ell$, and compute all its zeros $z_j := \mathrm{e}^{-\mathrm{i}hT_j}$, $j = 1, \dots, N$.
  5. Determine the frequencies $T_1, \dots, T_N \in \big(-\frac{\pi}{h}, \frac{\pi}{h}\big]$ by
  $$T_j = \frac{-\operatorname{Im}\big(\ln(z_j)\big)}{h}, \quad j = 1, \dots, N.$$
  6. For the frequencies $T_j$, $j = 1, \dots, N$, compute the corresponding coefficients $c_j$, $j = 1, \dots, N$, as least squares solution to the Vandermonde-type system
  $$\sum_{j=1}^{N} c_j \, \mathrm{e}^{-\mathrm{i}\,\ell h T_j} = P(\ell h), \quad \ell = 0, \dots, N.$$
- **Output:** Sequences $(T_j)_{j=1}^N$ and $(c_j)_{j=1}^N$, determining $P$ in (2).

**Remarks 2.2.**

1. In order to compute the frequencies $T_j$ for $j = 1, \dots, N$ uniquely, we need to ensure that $hT_j \in (-\pi, \pi]$ since the function $\omega \mapsto \mathrm{e}^{-\mathrm{i}\omega}$ is $2\pi$-periodic. Otherwise, we will not be able to extract the values $T_j$ from the zeros $z_j = \mathrm{e}^{-\mathrm{i}hT_j}$ of $\Lambda$ on the unit circle uniquely.
2. While the frequencies $T_j$ are not known, we only need to find a suitable upper bound for $|T_j|$ in order to fix a suitable step size $h$.
3. In applications, also the number $N$ of terms in (2) is usually unknown. Having given at least an upper bound $M \geq N$ and $M + 1$ sampling values $P(\ell h)$ for $\ell = 0, \dots, M$, we can also apply the above procedure (replacing $N$ by $M$) and obtain $N$ by examining the rank of $\boldsymbol{T}_{M+1}$ numerically. In this case, (5) cannot longer be solved uniquely, but each eigenvector corresponding to the eigenvalue 0 will serve for the determination of the zeros of $\Lambda$ on the unit circle and hence of $T_j$, $j = 1, \dots, N$, see [21], for example.

**Remark 2.3.** The Prony method as shown here is a method for parameter estimation in exponential sums where the exponential sum is univariate. In the next section, we want to use this method for solving a parameter estimation problem in bivariate exponential sums. For this purpose, we reduce the two-dimensional problem to some one-dimensional problems in order to be able to apply the Prony method as presented here.

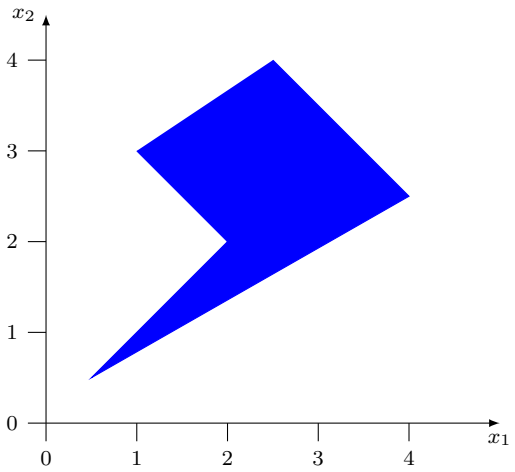


**Figure 1** *Example for a function of the form* (7) *with* 5 *vertices.*

## 3 Reconstruction of polygonal shapes in the space $\mathbb{R}^2$

Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ of the special form

$$f(\boldsymbol{x}) := \mathbf{1}_D(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{7}$$

where $\mathbf{1}_D$ is the characteristic function of the domain $D \subset \mathbb{R}^2$. Here, $D$ is a polygonal domain determined by the $N$ vertices $\boldsymbol{v}_j \in \mathbb{R}^2$, $j = 1, \ldots, N$, which are numbered anticlockwise. We always assume the polygon to be non-degenerated such that three neighbouring vertices do not lie on the same line. In the following, we will refer to the function $f$ as *unit-height polygon.*

We gain to reconstruct the domain $D$ from sparse Fourier samples. The Fourier transform of $f$ is given by

$$\widehat{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^2} \mathbf{1}_D(\boldsymbol{x})\, \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle}\, \mathrm{d}\boldsymbol{x} = \int_D \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle}\, \mathrm{d}\boldsymbol{x}, \qquad \boldsymbol{\xi} \in \mathbb{R}^2. \tag{8}$$

In order to reconstruct the polygonal domain from Fourier samples, we have to extract information about $D$ from these samples. Therefore, we need a representation of $\widehat{f}$ where information about its vertices and its edges can be easily extracted.

We employ a formula by Komrska, [14], and transfer it to our setting.

**Proposition 3.1** (Fourier transform of a unit-height polygon)**.**
*Let $D$ be a polygonal domain in $\mathbb{R}^2$ with the $N$ vertices $\boldsymbol{v}_j = (v_{j,1}, v_{j,2})^{\mathrm{T}}$,*

$j = 1, \dots, N$, *which are numbered anticlockwise with* $\boldsymbol{v}_{N+1} := \boldsymbol{v}_1$. *For* $\boldsymbol{\xi} \neq \boldsymbol{0}$, *the Fourier transform of the function* $f : \mathbb{R}^2 \ni \boldsymbol{x} \mapsto \mathbf{1}_D(\boldsymbol{x})$ *is given by*

$$\widehat{f}(\boldsymbol{\xi}) = \int_D \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle}\, \mathrm{d}\boldsymbol{x} = \frac{1}{\|\boldsymbol{\xi}\|_2^2} \sum_{j=1}^{N} \frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_j\rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j\rangle} \Big( \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_j\rangle} - \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1}\rangle} \Big) \quad (9)$$

*with*

$$\boldsymbol{n}_j := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} (\boldsymbol{v}_{j+1} - \boldsymbol{v}_j), \quad j = 1, \dots, N,$$

*and the convention*

$$\frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_j\rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j\rangle} \Big( \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_j\rangle} - \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1}\rangle} \Big) = \mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{n}_j\rangle\, \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_j\rangle}$$

*if* $\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j\rangle = 0$ *for some* $j \in \{1, \dots, N\}$.

The proof of this proposition follows the lines in [14].

Now we have a representation of the Fourier transform of the unit-height polygon where the vertices of the polygon occur. We aim to derive a theory for the reconstruction of the polygonal domain $D$ from sparse Fourier samples.

We want to assume that no edge of the polygon $D$ is parallel to the $x_1$-axis or the $x_2$-axis in the plane. Then it holds that

$$\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j\rangle \neq 0 \quad \text{for all } j \in \{1, \dots, N\}$$

for vectors $\boldsymbol{\xi}$ of the form

$$\boldsymbol{\xi} = (\xi_1, 0)^{\mathrm{T}} \quad \text{or} \quad \boldsymbol{\xi} = (0, \xi_2)^{\mathrm{T}} \quad \text{with} \quad \xi_1, \xi_2 \neq 0.$$

Using this assumption, we can rephrase (9) as

$$\widehat{f}(\boldsymbol{\xi}) = \frac{1}{\|\boldsymbol{\xi}\|_2^2} \sum_{j=1}^{N} \left( \frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_j\rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j\rangle} - \frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_{j-1}\rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_j - \boldsymbol{v}_{j-1}\rangle} \right) \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_j\rangle} \quad (10)$$

for vectors $\boldsymbol{\xi}$ of the form $\boldsymbol{\xi} = (\xi_1, 0)^{\mathrm{T}}$ or $\boldsymbol{\xi} = (0, \xi_2)^{\mathrm{T}}$ with $\xi_1, \xi_2 \neq 0$, where we use the conventions $\boldsymbol{v}_0 := \boldsymbol{v}_N$, $\boldsymbol{n}_0 := \boldsymbol{n}_N$, and $\boldsymbol{v}_{N+1} = \boldsymbol{v}_1$.

By considering the function

$$g(\boldsymbol{\xi}) := \|\boldsymbol{\xi}\|_2^2 \widehat{f}(\boldsymbol{\xi}), \quad (11)$$

we obtain a bivariate exponential sum with the vertices $\boldsymbol{v}_j$ of the polygon $D$ as exponents. Assuming that the vertex coordinates $v_{j,1}$, $j = 1, \dots, N$,

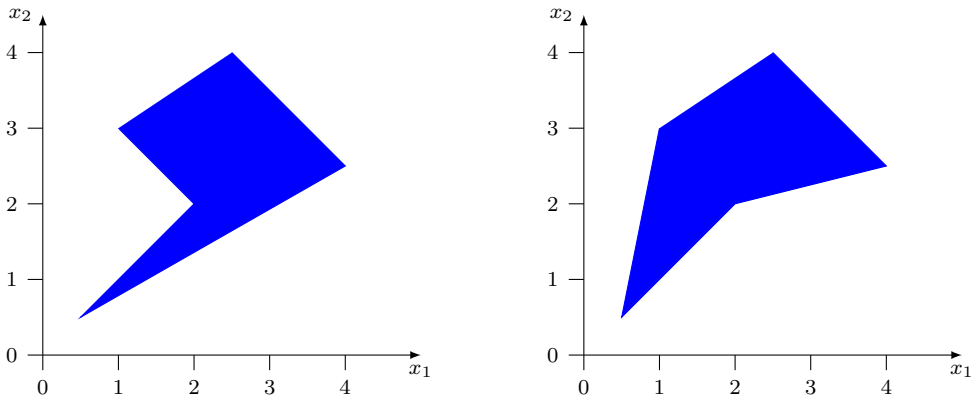


**Figure 2** *Example for two different unit-height polygons with the same* 5 *vertices.*

as well as the coordinates $v_{j,2}$, $j = 1, \ldots, N$, are pairwise different, we will be able to compute the vertices by applying the Prony method to sampling values from three straight lines through the origin where the third line is determined adaptively.

The assumption that the vertex components $v_{j,1}$ and $v_{j,2}$ are distinct in the $x_1$- and $x_2$-direction of the Cartesian coordinate system respectively also ensures that no edge of the polygon is parallel to the $x_1$- or $x_2$-axis.

However, it does not suffice to know the vertices of $D$, see Figure 2, since the polygon can be non-convex. Thus, we also need to determine which vertices have to be connected by line segments in order to reconstruct the original polygonal domain.

The ordering of the computed vertices will be done with the help of the coefficients which are obtained by applying the Prony method to three univariate problems, i.e. to the sampling values from three straight lines in the plane.

The proof of the following theorem is constructive and provides us also with an algorithmic scheme for the reconstruction of polygonal domains.

**Theorem 3.2** (Reconstruction of polygonal domains)**.**
*Let $f$ be a unit-height polygon as defined in (7), i.e. $f : \mathbb{R}^2 \to \mathbb{R}$ with*

$$f(\boldsymbol{x}) := \mathbf{1}_D(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2,$$

*where $D$ is a non-degenerated polygon in $\mathbb{R}^2$ with the vertices $\boldsymbol{v}_j = (v_{j,1}, v_{j,2})^{\mathrm{T}}$ for $j = 1, \ldots, N$, which are numbered anticlockwise. Assume that the vertex components $v_{j,1}$ and $v_{j,2}$ are distinct in the $x_1$- and $x_2$-direction of the*

*Cartesian coordinate system respectively. Further, assume that the constant $h > 0$ satisfies the condition $h\|\boldsymbol{v}_j\|_2 < \pi$ for all $j \in \{1, \dots, N\}$. Then we get the following reconstruction result:*

> *The polygon $D$ can be uniquely recovered from the $3N$ Fourier samples $\widehat{f}(\boldsymbol{\xi})$ for*
>
> $$\boldsymbol{\xi}^{\mathrm{T}} \in \big\{ \big(\ell h, 0\big), \big(0, \ell h\big), \big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big) \mid \ell = 1, \dots, N \big\}$$
>
> *where $\vartheta \in (0,1) \setminus \{\frac{1}{2}\}$ needs to be chosen suitably.*

**Proof.** The proof consists of five major parts. In the first two parts, we use the Prony method (applied to sampling values of $\widehat{f}$ from two straight lines in the frequency domain) in order to compute the coordinate values $v_{j,1}$ and $v_{j,2}$ of the vertices $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$.

But we are faced with the problem of combining these coordinate values to points in the plane which are the original vertices. Thus, in the third part, we determine a *candiate set* of points that are possible vertices of $D$. Dependently on this candidate set, we determine a third sampling line. We apply the Prony method to sampling values from this third line in order to uniquely determine the original vertices in the fourth part of the proof.

In the fifth and last part, we establish the right order of the computed vertices, for which we use the coefficients that we have obtained by the application of the Prony method.

<u>*Part 1:*</u>
The representation of the Fourier transform of $f$ in (10) yields

$$g(\boldsymbol{\xi}) = \sum_{j=1}^{N} \left( \frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_j \rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j \rangle} - \frac{\langle \boldsymbol{\xi}, \boldsymbol{n}_{j-1} \rangle}{\langle \boldsymbol{\xi}, \boldsymbol{v}_j - \boldsymbol{v}_{j-1} \rangle} \right) \mathrm{e}^{-\,\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{v}_j \rangle} \tag{12}$$

with $g$ as in (11) and the conventions

$$\boldsymbol{v}_{N+1} = \boldsymbol{v}_1, \quad \boldsymbol{v}_0 = \boldsymbol{v}_N, \quad \boldsymbol{n}_0 = \boldsymbol{n}_N. \tag{13}$$

Taking vectors $\boldsymbol{\xi}$ of the form $\boldsymbol{\xi} = (\xi_1, 0)^{\mathrm{T}}$, $\xi_1 \neq 0$, we obtain

$$g(\xi_1, 0) = \sum_{j=1}^{N} \left( \frac{\xi_1 n_{j,1}}{\xi_1 (v_{j+1,1} - v_{j,1})} - \frac{\xi_1 n_{j-1,1}}{\xi_1 (v_{j,1} - v_{j-1,1})} \right) \mathrm{e}^{-\,\mathrm{i}\,\xi_1 v_{j,1}},$$

where

$$n_{j,1} = v_{j+1,2} - v_{j,2}, \quad j = 0, \dots, N.$$

Thus, we have the univariate exponential sum

$$g(\xi_1, 0) = \sum_{j=1}^{N} a_j \, \mathrm{e}^{-\,\mathrm{i}\,\xi_1 v_{j,1}} \tag{14}$$

with the coefficients

$$a_j := \frac{v_{j+1,2} - v_{j,2}}{v_{j+1,1} - v_{j,1}} - \frac{v_{j,2} - v_{j-1,2}}{v_{j,1} - v_{j-1,1}}, \quad j = 1, \ldots, N. \tag{15}$$

Observe that the coefficient $a_j$ for $j \in \{1, \ldots, N\}$ is the difference of the slopes of the two line segments connecting the vertex $\boldsymbol{v}_j$ with its neighbours $\boldsymbol{v}_{j-1}$ and $\boldsymbol{v}_{j+1}$. By assumption, the polygon $D$ is non-degenerated. In particular, $a_j$ is non-zero and well defined. Hence, the values of the slopes of the edges are finite.

In order to obtain the vertex coordinates $v_{j,1}$ and the corresponding coefficients $a_j$, $j = 1, \ldots, N$, we apply the Prony method to the univariate function $g(\cdot, 0)$ in (14). Note that we have

$$\begin{aligned} g(0,0) &= \sum_{j=1}^{N} a_j = \sum_{j=1}^{N} \left( \frac{v_{j+1,2} - v_{j,2}}{v_{j+1,1} - v_{j,1}} - \frac{v_{j,2} - v_{j-1,2}}{v_{j,1} - v_{j-1,1}} \right) \\ &= \frac{v_{N+1,2} - v_{N,2}}{v_{N+1,1} - v_{N,1}} - \frac{v_{1,2} - v_{0,2}}{v_{1,1} - v_{0,1}} = 0 \end{aligned} \tag{16}$$

by (13). Further, the coefficients $a_j$ are real-valued, and we have $h|v_{j,1}| \leq h\|\boldsymbol{v}_j\|_2 < \pi$ for all $j \in \{1, \ldots, N\}$ by assumption. We apply the Prony method from Section 2 using the function values

$$g(\ell h, 0), \quad \ell = 0, \ldots, N,$$

which are given by (16) and by

$$g(\ell h, 0) = |\ell h|^2 \, \widehat{f}(\ell h, 0), \quad \ell = 1, \ldots, N.$$

The application of the Prony method then yields the set $\{v_{1,1}, \ldots, v_{N,1}\}$ of frequency values together with the set $\{a_1, \ldots, a_N\}$ of corresponding coefficients. We consider the ordered set $\{\alpha_1, \ldots, \alpha_N\} = \{v_{1,1}, \ldots, v_{N,1}\}$ with

$$\alpha_1 < \alpha_2 < \ldots < \alpha_N$$

and corresponding coefficients $a_{\alpha,j}$ with $\{a_{\alpha,1}, \ldots, a_{\alpha,N}\} = \{a_1, \ldots, a_N\}$. Observe that the order of the vertex components is still unknown and has to be determined later.

*Part 2:*
Analogously, we compute a set of values containing all vertex coordinates $v_{j,2}$ and corresponding coefficients by applying the Prony method to the univariate function

$$g(0,\xi_2) = \sum_{j=1}^{N} b_j \, \mathrm{e}^{-\,\mathrm{i}\,\xi_2 v_{j,2}} \tag{17}$$

where the coefficients $b_j$ are defined by

$$b_j := \frac{v_{j,1} - v_{j+1,1}}{v_{j+1,2} - v_{j,2}} - \frac{v_{j-1,1} - v_{j,1}}{v_{j,2} - v_{j-1,2}}, \quad j = 1, \ldots, N. \tag{18}$$

We obtain values

$$\beta_1 < \beta_2 < \ldots < \beta_N$$

with corresponding coefficients $b_{\beta,1}, \ldots, b_{\beta,N}$ where we have

$$\{\beta_1, \ldots, \beta_N\} = \{v_{1,2}, \ldots, v_{N,2}\} \quad \text{and} \quad \{b_{\beta,1}, \ldots, b_{\beta,N}\} = \{b_1, \ldots, b_N\}.$$

As before, the correct order of the vertex components $v_{j,2}$ is still unknown.

*Part 3:*
Now we compute the original vertices $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$. We consider the Cartesian product of the sets $\{\alpha_1, \ldots, \alpha_N\}$ and $\{\beta_1, \ldots, \beta_N\}$ as a set of *candidate points* for the original vertices,

$$K := \big\{ (\alpha_k, \beta_\ell)^{\mathrm{T}} : \ k = 1, \ldots, N, \ \ell = 1, \ldots, N \big\}.$$

Obviously, we have $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\} \subset K$.

In order to determine the $N$ original vertices, we apply the Prony method to Fourier samples from a third sampling line through the origin.

For this purpose, we choose a parameter $\vartheta \in (0,1) \backslash \{\frac{1}{2}\}$ such that the orthogonal projections of all candidate points in $K$ onto the line $x_2 = \tan(\vartheta\pi) x_1$ are pairwise different, see Figure 3. This parameter $\vartheta$ determines the third sampling line in the frequency domain. We use equispaced sampling locations on this third sampling line $\omega_2 = \tan(\vartheta\pi)\omega_1$ with step size $h$, see Figure 3, and take the $N$ Fourier samples

$$\widehat{f}\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big), \quad \ell = 1, \ldots, N.$$

Since the orthogonal projections of all points in $K$ onto the line $x_2 = \tan(\vartheta\pi)x_1$ are distinct, no possible edge of the polygon $D$, i.e. possible connection between two points in $K$, is perpendicular to the line $x_2 = \tan(\vartheta\pi)x_1$. Thus, we have

$$\langle \boldsymbol{\xi}, \boldsymbol{v}_{j+1} - \boldsymbol{v}_j \rangle \neq 0 \quad \text{for all } j \in \{1, \ldots, N\} \tag{19}$$

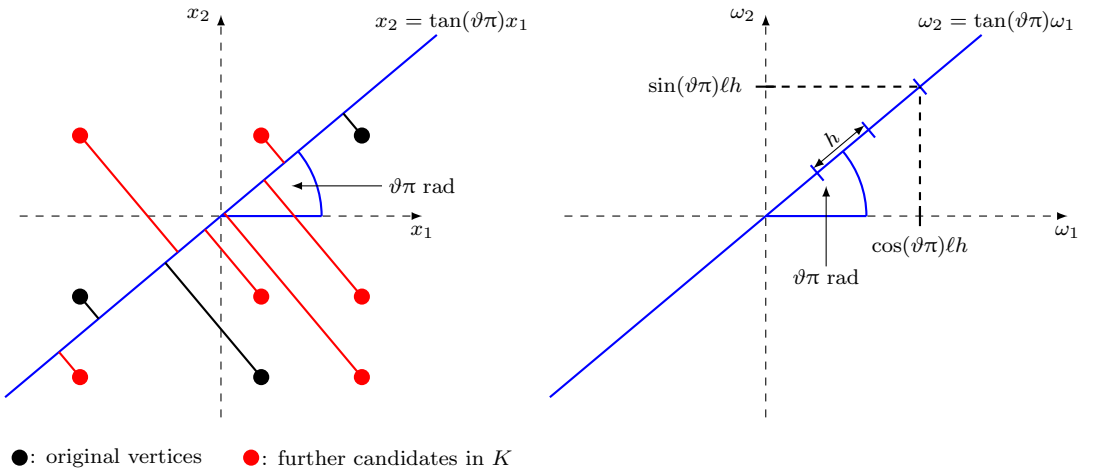


**Figure 3** Left: *Determination of the parameter $\vartheta$ in the time domain.* Right: *Third sampling line in the frequency domain with sampling locations (displayed example: $\ell = 3$).*

for $\boldsymbol{\xi} = \big(\cos(\vartheta\pi)\xi_1, \sin(\vartheta\pi)\xi_1\big)^{\mathrm{T}}$ with $\xi_1 \neq 0$, and we can use the representation of $g$ in (11) with $\widehat{f}$ in (10) along this third sampling line,

$$g\big(\cos(\vartheta\pi)\xi_1, \sin(\vartheta\pi)\xi_1\big) = \sum_{j=1}^{N} c_j \, \mathrm{e}^{-\mathrm{i}\,\xi_1 \left(\cos(\vartheta\pi)v_{j,1} + \sin(\vartheta\pi)v_{j,2}\right)} \tag{20}$$

where the coefficients $c_j$ for $j = 1, \ldots, N$ are given by

$$\begin{aligned} c_j := {} & \frac{\cos(\vartheta\pi)(v_{j+1,2} - v_{j,2}) + \sin(\vartheta\pi)(v_{j,1} - v_{j+1,1})}{\cos(\vartheta\pi)(v_{j+1,1} - v_{j,1}) + \sin(\vartheta\pi)(v_{j+1,2} - v_{j,2})} \\ & - \frac{\cos(\vartheta\pi)(v_{j,2} - v_{j-1,2}) + \sin(\vartheta\pi)(v_{j-1,1} - v_{j,1})}{\cos(\vartheta\pi)(v_{j,1} - v_{j-1,1}) + \sin(\vartheta\pi)(v_{j,2} - v_{j-1,2})}. \end{aligned} \tag{21}$$

Note that these coefficients are non-zero and well defined due to (19). This can be seen similarly as in the cases of the coefficients $a_j$ and $b_j$ in *Parts 1* and *2* of this proof since the coefficient $c_j$ for $j \in \{1, \ldots, N\}$ is also a difference of slopes of two neighbouring line segments.

In order to determine the frequency values $\cos(\vartheta\pi)v_{j,1} + \sin(\vartheta\pi)v_{j,2}$ and the coefficients $c_j$ for all $j \in \{1, \ldots, N\}$ in (20), we apply the Prony method. Observe that the assumption that $h\|\boldsymbol{v}_j\|_2 < \pi$ for all $j \in \{1, \ldots, N\}$ is satisfied. Indeed, the rotated vectors

$$\begin{pmatrix} \cos(\vartheta\pi) & \sin(\vartheta\pi) \\ -\sin(\vartheta\pi) & \cos(\vartheta\pi) \end{pmatrix} \boldsymbol{v}_j, \quad j = 1, \ldots, N,$$

fulfil the same norm condition. Thus, we have

$$\begin{aligned} & h\big|\cos(\vartheta\pi)v_{j,1}+\sin(\vartheta\pi)v_{j,2}\big| \\ & \quad = h\sqrt{\big(\cos(\vartheta\pi)v_{j,1}+\sin(\vartheta\pi)v_{j,2}\big)^2} \\ & \quad \le h\sqrt{\big(\cos(\vartheta\pi)v_{j,1}+\sin(\vartheta\pi)v_{j,2}\big)^2+\big(-\sin(\vartheta\pi)v_{j,1}+\cos(\vartheta\pi)v_{j,2}\big)^2} \\ & \quad = \left\| \begin{pmatrix} \cos(\vartheta\pi) & \sin(\vartheta\pi) \\ -\sin(\vartheta\pi) & \cos(\vartheta\pi) \end{pmatrix} \boldsymbol{v}_j \right\|_2 < \pi \end{aligned}$$

for $j = 1, \ldots, N$. Hence, we can apply the Prony method as described in Section 2, where we use the function values

$$g\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big) = |\ell h|^2 \widehat{f}\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big), \quad \ell = 1, \ldots, N.$$

Further, we have $g(0,0) = 0$ by (20) and (13).

We obtain the ordered frequency values

$$\gamma_1 < \gamma_2 < \ldots < \gamma_N$$

with corresponding coefficients $c_{\gamma,1}, \ldots, c_{\gamma,N}$, where

$$\begin{aligned} & \{\gamma_1, \ldots, \gamma_N\} \\ & \quad = \big\{\big(\cos(\vartheta\pi)v_{1,1}+\sin(\vartheta\pi)v_{1,2}\big), \ldots, \big(\cos(\vartheta\pi)v_{N,1}+\sin(\vartheta\pi)v_{N,2}\big)\big\} \end{aligned}$$

and $\{c_{\gamma,1}, \ldots, c_{\gamma,N}\} = \{c_1, \ldots, c_N\}$.

*Part 4:*

The previous results now enable us to compute the original vertices $\boldsymbol{v}_j$, $j = 1, \ldots, N$, by comparison of the set $K$ of candidate points with the set $\{\gamma_1, \ldots, \gamma_N\}$. We determine all points $(\alpha_k, \beta_\ell)^{\mathrm{T}}$ in the set $K$ for which there exist indices $j \in \{1, \ldots, N\}$ such that $\cos(\vartheta\pi)\alpha_k + \sin(\vartheta\pi)\beta_\ell = \gamma_j$. Then the set

$$\widetilde{G} := \big\{(\alpha_k, \beta_\ell)^{\mathrm{T}} \in K \mid \exists\, j \in \{1, \ldots, N\} :\ \cos(\vartheta\pi)\alpha_k + \sin(\vartheta\pi)\beta_\ell = \gamma_j\big\}$$

contains all $N$ original vertices $\boldsymbol{v}_j$, $j = 1, \ldots, N$, of the polygon $D$ and it holds that

$$\big|\widetilde{G}\big| = N;$$

that is, the set $\widetilde{G}$ contains only the vertices $\boldsymbol{v}_j$ and no other elements. We sort the elements of $\widetilde{G}$ in an arbitrary order such that we have

$$\widetilde{G} = \big\{\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_N\big\} \quad \text{with} \quad \widetilde{\boldsymbol{v}}_j = \big(\widetilde{\alpha}_j, \widetilde{\beta}_j\big)^{\mathrm{T}} \text{ for } j = 1, \ldots, N.$$

*Part 5:*
Recall that each element $\widetilde{\boldsymbol{v}}_j$ of $\widetilde{G}$ has three corresponding coefficient values

$$\widetilde{a}_j, \quad \widetilde{b}_j, \quad \text{and} \quad \widetilde{c}_j,$$

which are the coefficients obtained by applying the Prony method to the three problems (14), (17), and (20).

In this part, we establish the right order of the vertices of $D$. Using the coefficient values $\widetilde{a}_j$, $\widetilde{b}_j$, and $\widetilde{c}_j$, we can compute the slopes of the polygon's edges such that we are able to determine a predecessor and a successor for each vertex in the set $\widetilde{G}$.

*Part 5a:* First, let us take a look at the original edges of the polygon $D$. We define

$$m_j := \frac{v_{j+1,2} - v_{j,2}}{v_{j+1,1} - v_{j,1}}, \quad j = 1, \ldots, N,$$

describing the slope of the line which contains the edge connecting $\boldsymbol{v}_j$ and $\boldsymbol{v}_{j+1}$. Observe that $m_j$ is well defined and non-zero by assumption. Using this definition of $m_j$, the coefficients $a_j$, $b_j$, and $c_j$ in (15), (18), and (21) respectively, corresponding to the vertex $\boldsymbol{v}_j$ for $j = 1, \ldots, N$, can be written as follows:

$$a_j = \frac{v_{j+1,2} - v_{j,2}}{v_{j+1,1} - v_{j,1}} - \frac{v_{j,2} - v_{j-1,2}}{v_{j,1} - v_{j-1,1}} = m_j - m_{j-1}, \tag{22}$$

$$b_j = \frac{v_{j,1} - v_{j+1,1}}{v_{j+1,2} - v_{j,2}} - \frac{v_{j-1,1} - v_{j,1}}{v_{j,2} - v_{j-1,2}} = -\frac{1}{m_j} + \frac{1}{m_{j-1}}, \tag{23}$$

$$c_j = \frac{m_j - m_{j-1}}{[\cos(\vartheta\pi) + \sin(\vartheta\pi)m_j] \cdot [\cos(\vartheta\pi) + \sin(\vartheta\pi)m_{j-1}]}. \tag{24}$$

The representations (22), (23), and (24) can now be used to compute the slope $m_j$ if the coefficients $a_j$, $b_j$, and $c_j$ are known. For $j = 1, \ldots, N$, we obtain the following system of equations:

$$\begin{aligned} a_j &= m_j - m_{j-1}, \\ b_j &= -\frac{1}{m_j} + \frac{1}{m_{j-1}}, \\ c_j &= \frac{m_j - m_{j-1}}{[\cos(\vartheta\pi) + \sin(\vartheta\pi)m_j] \cdot [\cos(\vartheta\pi) + \sin(\vartheta\pi)m_{j-1}]}, \end{aligned}$$

with $m_0 := M_N$. As the solution to this system, we find

$$m_{j-1} = \frac{a_j}{2c_j \sin(\vartheta\pi)\cos(\vartheta\pi)} - \frac{a_j}{2b_j}\tan(\vartheta\pi) - \frac{a_j}{2} - \cot(\vartheta\pi) \tag{25}$$

and

$$m_j = \frac{a_j}{2c_j \sin(\vartheta\pi)\cos(\vartheta\pi)} - \frac{a_j}{2b_j}\tan(\vartheta\pi) + \frac{a_j}{2} - \cot(\vartheta\pi). \tag{26}$$

Observe that the slopes $m_{j-1}$ and $m_j$ of the edges connecting $\boldsymbol{v}_j$ with its predecessor and successor respectively can be computed separately for each $j$ by using only the coefficient values corresponding to the considered vertex $\boldsymbol{v}_j$.

Further, note that the computation of $m_{j-1}$ and $m_j$ in (25) and (26) is well defined since the values $b_j$ and $c_j$ are non-zero.

Moreover, observe that we have $\sin(\vartheta\pi) \neq 0$ and $\cos(\vartheta\pi) \neq 0$ since $\vartheta \in (0,1) \setminus \{\frac{1}{2}\}$. Therefore, also $\tan(\vartheta\pi)$ and $\cot(\vartheta\pi)$ are well defined.

*Part 5b:* Let us turn back to the set

$$\widetilde{G} = \{\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_N\} \tag{27}$$

of the vertices of the polygonal domain $D$, where we have to establish which elements have to be connected by edges in order to uniquely reconstruct $D$. Since we have three corresponding values $\widetilde{a}_j$, $\widetilde{b}_j$, and $\widetilde{c}_j$ for each element $\widetilde{\boldsymbol{v}}_j$ in $\widetilde{G}$, we can use the formulae (25) and (26) in order to determine a predecessor and a successor for each vertex.

We start with one vertex of the convex hull of $\widetilde{G}$. Without loss of generality, let this be $\widetilde{\boldsymbol{v}}_1$. Then initialize the set $G$ of ordered vertices of $D$ to

$$G = \{\boldsymbol{v}_1\} \quad \text{with} \quad \boldsymbol{v}_1 := \widetilde{\boldsymbol{v}}_1.$$

Further, we update the set $\widetilde{G}$ of the not yet ordered vertices,

$$\widetilde{G} := \widetilde{G} \setminus \{\boldsymbol{v}_1\} = \{\widetilde{\boldsymbol{v}}_2, \ldots, \widetilde{\boldsymbol{v}}_N\}.$$

Using (26), we compute the slope $m_1$ of the edge that connects the vertex $\boldsymbol{v}_1$ with its successor $\boldsymbol{v}_2$ by

$$m_1 = \frac{\widetilde{a}_1}{2\widetilde{c}_1 \sin(\vartheta\pi)\cos(\vartheta\pi)} - \frac{\widetilde{a}_1}{2\widetilde{b}_1}\tan(\vartheta\pi) + \frac{\widetilde{a}_1}{2} - \cot(\vartheta\pi).$$

The line containing the edge between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is then given by the equation

$$x_2 = m_1 \cdot \big(x_1 - v_{1,1}\big) + v_{1,2}.$$

Now we determine all points $\widetilde{\boldsymbol{v}}_j = (\widetilde{v}_{j,1}, \widetilde{v}_{j,2})^{\mathrm{T}}$ in $G$ on this line, i.e. the points satisfying

$$\widetilde{v}_{j,2} = m_1 \cdot \big(\widetilde{v}_{j,1} - v_{1,1}\big) + v_{1,2} \tag{28}$$

and hence obtain the set

$$S_1 := \{\widetilde{\boldsymbol{v}}_j \in \widetilde{G} \mid \widetilde{\boldsymbol{v}}_j \text{ fulfils (28)}\}$$

of possible successors of $\boldsymbol{v}_1$. Now we have to establish which point in $S_1$ is actually the successor $\boldsymbol{v}_2$. For this purpose, we have to consider the following cases:

(1a) $|S_1| = 1$: If $S_1$ contains only one element, i.e.

$$S_1 = \{\widetilde{\boldsymbol{v}}_{j_1}\} \quad \text{for some } j_1 \in \{2, \ldots, N\},$$

then the successor $\boldsymbol{v}_2$ is given by $\widetilde{\boldsymbol{v}}_{j_1}$, and we set

$$G = \{\boldsymbol{v}_1, \boldsymbol{v}_2\} \quad \text{with} \quad \boldsymbol{v}_2 := \widetilde{\boldsymbol{v}}_{j_1}$$

and $\widetilde{G} := \widetilde{G} \setminus \{\boldsymbol{v}_2\}$.

(1b) $|S_1| \geq 2$: Remember that $\boldsymbol{v}_1$ is a vertex of the convex hull of $\widetilde{G}$. Thus, if $S_1$ contains at least two elements, all these elements can only lie in the same direction from $\boldsymbol{v}_1$, and only the point in $S_1$ which is nearest to $\boldsymbol{v}_1$ is a possible choice for $\boldsymbol{v}_2$, see Figure 4. Otherwise, we would have a degenerate case, but such a case is excluded by assumption. Having given the only possible choice for the successor $\boldsymbol{v}_2$, we proceed as described in the case (1a).

Now we repeat the approach explained above in order to determine the successor $\boldsymbol{v}_3$ of the vertex $\boldsymbol{v}_2$. The line which contains the edge connecting $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ is described by

$$x_2 = m_2 \cdot \big(x_1 - v_{2,1}\big) + v_{2,2} \tag{29}$$

where the slope $m_2$ is computed using (26). The set of possible successors of $\boldsymbol{v}_2$ is given by

$$S_2 := \{\widetilde{\boldsymbol{v}}_j \in \widetilde{G} \mid \widetilde{\boldsymbol{v}}_j \text{ fulfils (29)}\}.$$

(2a) $|S_2| = 1$: If $S_2$ contains only one element $\widetilde{\boldsymbol{v}}_{j_2}$, then $\widetilde{\boldsymbol{v}}_{j_2}$ is the successor of $\boldsymbol{v}_2$, and we set

$$G = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\} \quad \text{with} \quad \boldsymbol{v}_3 := \widetilde{\boldsymbol{v}}_{j_2}$$

and $\widetilde{G} := \widetilde{G} \setminus \{\boldsymbol{v}_3\}$.

The case $|S_2| \geq 2$ is a bit different from the case (1b) since $\boldsymbol{v}_2$ may lie inside the convex hull of $D$.

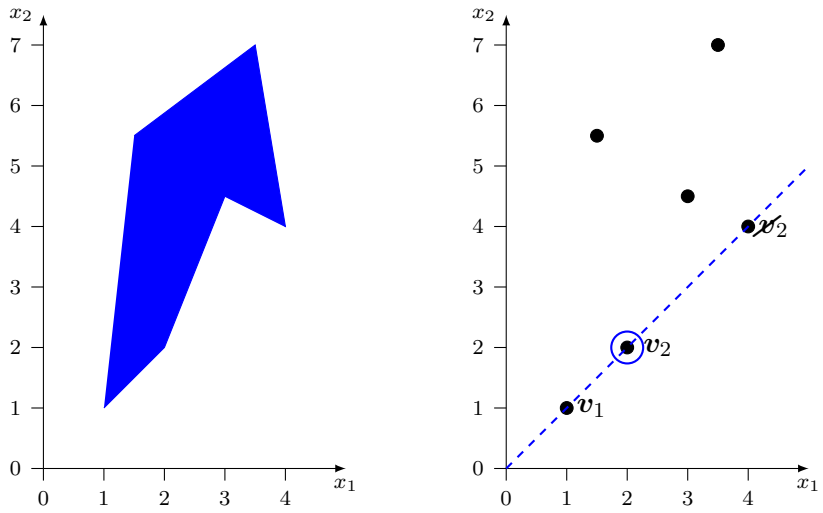


**Figure 4** Left: *Original unit-height polygon.* Right: *Determination of the order of the computed vertices. Start with* $\boldsymbol{v}_1 := (1,1)^{\mathrm{T}}$. *Computation of* $m_1$ *accordingly to* (26) *leads to* $(2,2)^{\mathrm{T}}$ *and* $(4,4)^{\mathrm{T}}$ *as candidates for* $\boldsymbol{v}_2$. *Choice of* $(4,4)^{\mathrm{T}}$ *would lead to a degenerate case. Hence,* $\boldsymbol{v}_2 = (2,2)^{\mathrm{T}}$.

(2b) $|S_2| \geq 2$: Note that we only know the line containing the edge between $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, but we do not know in which direction we have to go on this line, starting at $\boldsymbol{v}_2$, in order to reach the successor $\boldsymbol{v}_3$. For each of the two directions, only the point in $S_2$ which is the nearest neighbour of $\boldsymbol{v}_2$ is a possible choice for $\boldsymbol{v}_3$. Otherwise, we would have a degenerate case. Thus, we obtain a new set $\widetilde{S}_2$ which contains at most two points that fulfil (29) and hence are possible choices for the successor $\boldsymbol{v}_3$. If $\left|\widetilde{S}_2\right| = 1$, then we follow the lines in the case (2a).

Consider now the case $\left|\widetilde{S}_2\right| = 2$, i.e.

$$\widetilde{S}_2 = \left\{\widetilde{\boldsymbol{v}}_{j_2}, \widetilde{\boldsymbol{v}}_{j_3}\right\} \quad \text{for some } j_2, j_3 \in \{2, \ldots, N\} \setminus \{j_1\}.$$

For each *candidate point* $\boldsymbol{v}_{3,c,j}$ in $\widetilde{S}_2$ ($j = j_2, j_3$), we compute the slope $m_{3-1,c,j}$ of the line containing the edge between this point and its predecessor by using formula (25). The successor $\boldsymbol{v}_3$ is now given by the point $\boldsymbol{v}_{3,c,j}$ for which we have

$$m_2 = m_{3-1,c,j}, \tag{30}$$

see also Figure 5, and we proceed as described in the case (2a).

We use this approach iteratively in order to determine the order of the remaining vertices until we have computed a successor for every element in

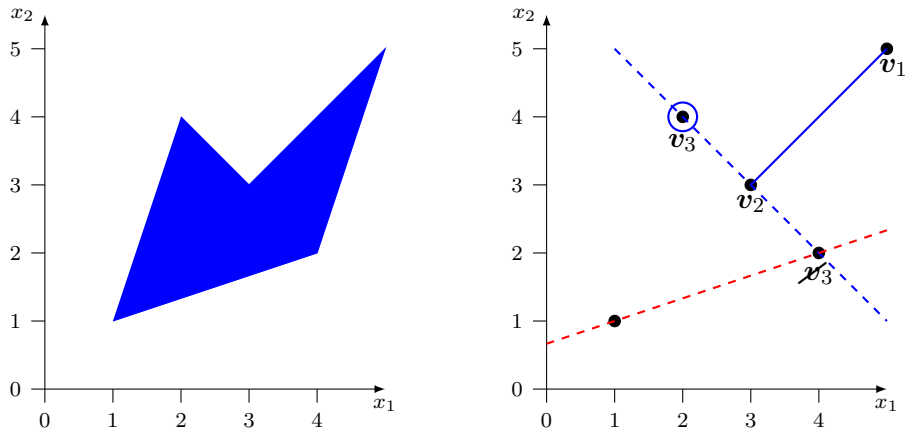


**Figure 5** Left: *Original unit-height polygon.* Right: *Determination of the order of the computed vertices. Starting with* $\boldsymbol{v}_1 := (5,5)^{\mathrm{T}}$ *leads to* $\boldsymbol{v}_2 = (3,3)^{\mathrm{T}}$. *Computation of* $m_2$ *accordingly to* (26) *leads to* $(2,4)^{\mathrm{T}}$ *and* $(4,2)^{\mathrm{T}}$ *as candidates for* $\boldsymbol{v}_3$ *(blue, dashed line).* Equation (25) *yields that the predecessor of* $(2,4)^{\mathrm{T}}$ *lies on the blue, dashed line, and that the predeccessor of* $(4,2)^{\mathrm{T}}$ *lies on the red, dashed line. Thus,* $\boldsymbol{v}_3 = (2,4)^{\mathrm{T}}$.

the set given in (27) such that the successor of the last considered vertex is the vertex $\boldsymbol{v}_1$, with which we have started.

If (30) holds for both candidate points at some stage of the iteration, we have to choose arbitrarily one of those points as the next successor vertex. If this choice is the wrong one, the algorithm will also terminate when a computed successor is equal to the first considered vertex $\boldsymbol{v}_1$. But in this case we will not have determined a successor for all elements in the set $\widetilde{G}$ in (27) such that we have to turn back to the stage of the iteration where we had to choose a successor arbitrarily and continue the iteration by choosing the other candidate point.

In this manner, we uniquely reconstruct the polygonal domain $D$. This concludes the proof. □

**Remarks 3.3.**

1. It may happen that an incorrect, arbitrary choice is not discovered instantly as being wrong, but it is possible that other arbitrary choices have to be made until the algorithm terminates without determining successors for all vertices. In this case, one has to go back to the stages of the iteration where an arbitrary choice has been made, and where the at that time not considered candidate has not been used as a successor afterwards. The edges constructed there have to be erased, and one has to proceed with the unused candidates.

2. In order to avoid the problems mentioned in the first remark, one can determine sets of possible successors and sets of possible predecessors for each vertex $\widetilde{\boldsymbol{v}}_j$ in the set $\widetilde{G}$ given in (27). Starting with sets which contain only one possible successor or predecessor, one can construct all edges of the polygon step by step. For this purpose, one has to update the sets of possible successors and predecessors every time when an edge is constructed.

   Note that at least each of the vertices of the convex hull of $\widetilde{G}$ has only one possible successor and one possible predecessor, see the case (1b) in the proof of Theorem 3.2.

3. In *Part 3* of the proof of Theorem 3.2, we have to choose a parameter $\vartheta \in (0,1) \setminus \{\frac{1}{2}\}$ such that the orthogonal projections of all candidate points in $K$ onto the line $x_2 = \tan(\vartheta\pi)x_1$ are pairwise different. In order to improve the robustness of the reconstruction method, $\vartheta$ should be chosen in such a way that the minimal distance between two orthogonal projections of candidate vectors from $K$ onto the line $x_2 = \tan(\vartheta\pi)x_1$ is maximized. With $\boldsymbol{u} := \big(\cos(\vartheta\pi), \sin(\vartheta\pi)\big)^{\mathrm{T}}$, the orthogonal projection $\mathrm{P}(\boldsymbol{v})$ of $\boldsymbol{v} \in K$ onto this line is given by

   $$\mathrm{P}(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{u}\rangle \boldsymbol{u} = \big(\cos(\vartheta\pi)v_1 + \sin(\vartheta\pi)v_2\big)\boldsymbol{u},$$

   and the distance between two projections $\mathrm{P}(\boldsymbol{v})$ and $\mathrm{P}(\boldsymbol{w})$, $\boldsymbol{v}, \boldsymbol{w} \in K$, $\boldsymbol{v} \neq \boldsymbol{w}$ is given by $\|\mathrm{P}(\boldsymbol{v}) - \mathrm{P}(\boldsymbol{w})\|_2$, for which we have

   $$\|\mathrm{P}(\boldsymbol{v}) - \mathrm{P}(\boldsymbol{w})\|_2^2 = \big(\langle \boldsymbol{v} - \boldsymbol{w}, \boldsymbol{u}\rangle\big)^2$$

   since $\|\boldsymbol{u}\|_2 = 1$. Thus, in order to choose $\vartheta$ as mentioned, we need to maximize the minimal distance between two projections with respect to $\vartheta$, that is, we have to solve the max-min problem

   $$\max_{\vartheta \in (0,1)\setminus\{\frac{1}{2}\}} \min_{\substack{\boldsymbol{v},\boldsymbol{w}\in K \\ \boldsymbol{v}\neq\boldsymbol{w}}} \big(\langle \boldsymbol{v} - \boldsymbol{w}, \boldsymbol{u}\rangle\big)^2.$$

4. The scheme for the reconstruction of polygonal domains from sparse Fourier data as proposed in Theorem 3.2 can also be used if the number $N$ of the vertices of the polygon $D$ is not known a priori. In that case, we need an upper bound $M \geq N$ and $3M$ sampling values of the Fourier transform of $\mathbf{1}_D$, compare Remarks 2.2, 3.

We summarize the proposed reconstruction scheme in the following algorithm:

**Algorithm 3.4** (Reconstruction of polygonal domains)**.**

- **Input:**
  - Step size $h > 0$ with $h\|\boldsymbol{v}_j\|_2 < \pi$ for $j = 1, \dots, N$;
  - Fourier samples $\widehat{f}(\boldsymbol{\xi})$ for

$$\boldsymbol{\xi}^{\mathrm{T}} \in \big\{\big(\ell h, 0\big), \big(0, \ell h\big), \big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big)\big\}, \quad \ell = 1, \dots, N,$$

    where $\vartheta \in (0,1) \setminus \{\frac{1}{2}\}$ needs to be chosen suitably, see Step 6 of the computation.

- **Computation:**
  1. Compute the values $g(\ell h, 0) = |\ell h|^2 \widehat{f}(\ell h, 0)$ for $\ell = 1, \dots, N$ and set $g(0,0) = 0$.
  2. Use Algorithm 2.1 in order to compute the parameters $v_{j,1}$ and $a_j$ of $g$ in (14).
  3. Compute the values $g(0, \ell h) = |\ell h|^2 \widehat{f}(0, \ell h)$ for $\ell = 1, \dots, N$ and set $g(0,0) = 0$.
  4. Use Algorithm 2.1 in order to compute the parameters $v_{j,2}$ and $b_j$ of $g$ in (17).
  5. Compute the Cartesian product $\{v_{1,1} \dots, v_{N,1}\} \times \{v_{1,2}, \dots, v_{N,2}\}$ as the set of possible candidates for the true vertices, i.e.

$$K := \big\{(v_{k,1}, v_{\ell,2})^{\mathrm{T}} : \ k = 1, \dots, N, \ \ell = 1, \dots, N\big\}.$$

  6. Choose a parameter $\vartheta \in (0,1) \setminus \{\frac{1}{2}\}$ such that the orthogonal projections of all candidate points in $K$ onto the line $x_2 = \tan(\vartheta\pi)x_1$ are pairwise different.
  7. Acquire the Fourier samples $\widehat{f}\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big)$, $\ell = 1, \dots, N$. Then compute

$$g\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big) = |\ell h|^2 \widehat{f}\big(\cos(\vartheta\pi)\ell h, \sin(\vartheta\pi)\ell h\big)$$

     for $\ell = 1, \dots, N$ and set $g(0,0) = 0$.
  8. Use Algorithm 2.1 in order to compute the parameters of $g$ in (20), namely $\gamma_j = \big(\cos(\vartheta\pi)v_{j,1} + \sin(\vartheta\pi)v_{j,2}\big)$ and $c_j$.
  9. Determine the true vertices $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ by comparison of $K$ with the set $\{\gamma_1, \dots, \gamma_N\}$; that is, determine all points $(\alpha_k, \beta_\ell)^{\mathrm{T}}$ in the set $K$ for which there exist indices $j \in \{1, \dots, N\}$ such that

$$\cos(\vartheta\pi)\alpha_k + \sin(\vartheta\pi)\beta_\ell = \gamma_j.$$

     This results in the set $\widetilde{G}$ of all true vertices.

10. Establish the right order of the elements in $\widetilde{G}$ by computing successors and predecessors for each vertex. For this purpose, consider the explanations following (27) as well as Remarks 3.3, 1. and 2.

- **Output:**

  Set $G = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ of ordered vertices; that is, the boundary $\partial D$ of the polygon $D$ is determined by the closed polygonal chain $[\boldsymbol{v}_1 — \boldsymbol{v}_2 — \boldsymbol{v}_3 — \ldots — \boldsymbol{v}_{N-1} — \boldsymbol{v}_N — \boldsymbol{v}_1]$.

**Remark 3.5.**
In the proof of Theorem 3.2 and in the above algorithm, we rely upon the assumption that the coordinates of the vertices of the polygons to be reconstructed are pairwise different in both dimensions.

Our reconstruction method uses the orthogonal projections of the vertices onto the $x_1$- and the $x_2$-axis in the time domain, which are computed by using Fourier samples from corresponding sampling lines in the frequency domain, i.e. the $\omega_1$- and the $\omega_2$-axis.

The assumption that the vertex coordinates $v_{j,1}$, $j = 1, \ldots, N$, as well as the coordinates $v_{j,2}$, $j = 1, \ldots, N$, are pairwise different is needed since the computed orthogonal projections onto the $x_1$- and the $x_2$-axis have to be distinct such that we can use the formula (10) for the Fourier transform of the unit-height polygons.

One may be able to generalize formula (10) such that this restriction is not necessary. Then we could reconstruct general polygons, i.e. polygons without any restrictions on the vertex coordinates, using our proposed reconstruction scheme.

Without such a generalized formula, we need to take much more Fourier samples for the reconstruction of general polygons. By using Fourier data sampled on distinct lines $\widetilde{l}_k := \{\lambda \boldsymbol{u}_k \,|\, \lambda \in \mathbb{R}\}$, $k = 1, \ldots, r$, in the frequency domain where $\boldsymbol{u}_k$ are unit vectors, we obtain the orthogonal projections of the vertices onto the distinct lines $l_k := \{\lambda \boldsymbol{u}_k \,|\, \lambda \in \mathbb{R}\}$, $k = 1, \ldots, r$, in the time domain. For all $k \in \{1, \ldots, r\}$, these orthogonal projections are given by

$$\langle \boldsymbol{u}_k, \boldsymbol{v}_j \rangle \boldsymbol{u}_k, \quad j = 1, \ldots, N.$$

Thus, we have to ensure that we have always two distinct lines on which the projections are pairwise different. For this purpose, we need $\frac{N(N-1)}{2} + 2$ pairwise different lines $l_k$ and $\widetilde{l}_k$ in the time domain and the frequency domain respectively, see [2, §3], where Buhmann and Pinkus use this approach for a similar problem.

We take the usual coordinate axes and consider the vertical and horizontal axes of $\frac{N(N-1)}{2}$ rotated versions of the usual Cartesian coordinate system

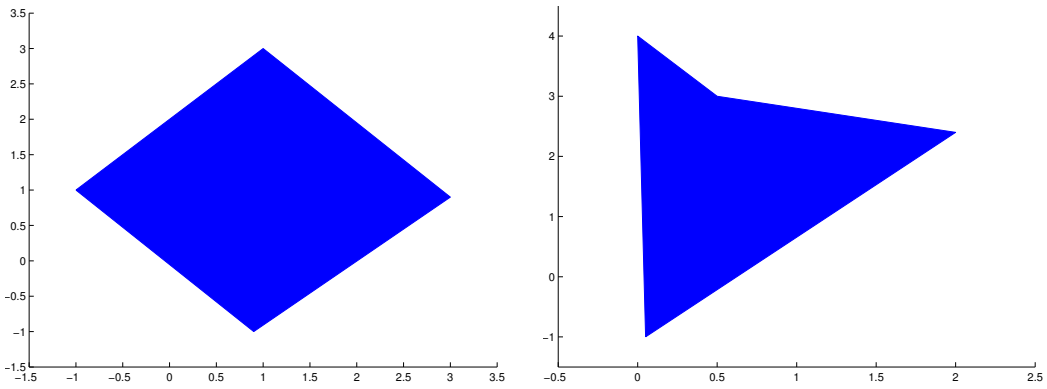


**Figure 6** Left: *Unit-height polygon determined by* $(\boldsymbol{v}_j)_{j=1}^4$ *given in* Table 1. Right: *Unit-height polygon determined by* $(\boldsymbol{v}_j)_{j=1}^4$ *given in* Table 2.

as further sampling lines. Let us assume that the orthogonal projections of the vertices are pairwise different onto the lines $l_1$ and $l_2$. Then we can use the Fourier samples from the lines $\widetilde{l}_1$ and $\widetilde{l}_2$ as initializations for Algorithm 3.4. Observe that we have to use an appropriate coordinate transformation, corresponding to the coordinate system spanned by $l_1$ and $l_2$, in order to continue with Algorithm 3.4.

Altogether, we need Fourier samples from $\frac{N(N-1)}{2} + 3$ sampling lines. But we will only use the Fourier data from three lines. Therefore, it would be of high interest to have an approach similar to our proposed reconstruction method where only three sampling lines are needed. This means that another representation for the Fourier transform of a unit-height polygon is required.

## 4 Numerical results

In this section, we want to illustrate the developed reconstruction method with a few numerical examples. We use simulated Fourier data on the $x_1$-axis, on the $x_2$-axis, and on a third, adaptively chosen sampling line.

Observe in the following examples that some vertices have nearly the same first or second coordinate. But nevertheless we are able to recover the original vertices, and the maximal reconstruction error for the vertex coordinates has an order of magnitude equal to $-7$.

Further, note that concave polygons are considered in the second, third, and fourth example such that the step of determining the order of the computed vertices is very important in order to recover the original shape.

| $j$ | $v_{j,1}$ | $\lvert v_{j,1} - v^*_{j,1}\rvert$ | $v_{j,2}$ | $\lvert v_{j,2} - v^*_{j,2}\rvert$ |
|---|---|---|---|---|
| 1 | -1 | $8.882 \cdot 10^{-16}$ | 1 | $7.649 \cdot 10^{-14}$ |
| 2 | 1 | $1.91 \cdot 10^{-14}$ | 3 | $4.441 \cdot 10^{-16}$ |
| 3 | 3 | $2.665 \cdot 10^{-15}$ | 0.9 | $8.737 \cdot 10^{-14}$ |
| 4 | 0.9 | $8.626 \cdot 10^{-14}$ | -1 | $2.22 \cdot 10^{-16}$ |

**Table 1** *Vertices of the unit-height polygon displayed in* Figure 6 (left) *and approximate reconstruction errors. The sampling step size is* $h = 0.7$*, and the measure of the angle between the first and the third sampling line is* $64°$*.*

| $j$ | $v_{j,1}$ | $\lvert v_{j,1} - v^*_{j,1}\rvert$ | $v_{j,2}$ | $\lvert v_{j,2} - v^*_{j,2}\rvert$ |
|---|---|---|---|---|
| 1 | 0.05 | $2.732 \cdot 10^{-12}$ | 0 | $9.992 \cdot 10^{-16}$ |
| 2 | 0 | $2.273 \cdot 10^{-12}$ | 4 | $3.109 \cdot 10^{-15}$ |
| 3 | 0.5 | $2.224 \cdot 10^{-12}$ | 3 | $2.665 \cdot 10^{-15}$ |
| 4 | 2 | $1.776 \cdot 10^{-15}$ | 2.4 | 0 |

**Table 2** *Vertices of the unit-height polygon displayed in* Figure 6 (right) *and approximate reconstruction errors. The sampling step size is* $h = 0.7$*, and the measure of the angle between the first and the third sampling line is* $10°$*.*

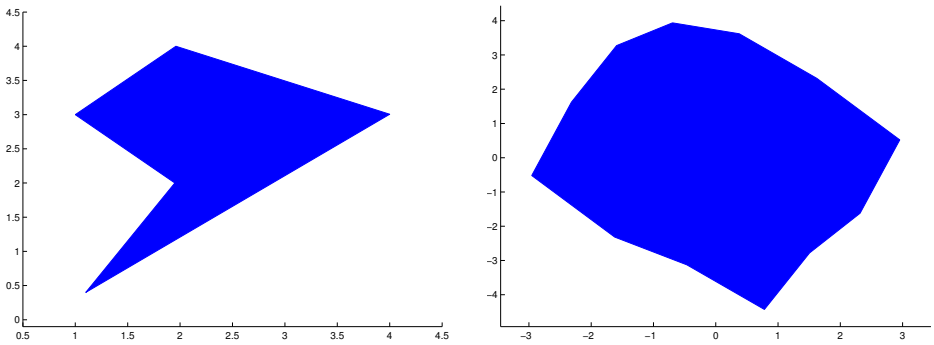

**Figure 7** Left: *Unit-height polygon determined by* $(\boldsymbol{v}_j)_{j=1}^5$ *given in* Table 3. Right: *Unit-height polygon with* 12 *vertices. The approximate reconstruction errors are given in* Table 4.

| $j$ | $v_{j,1}$ | $\lvert v_{j,1} - v^*_{j,1}\rvert$ | $v_{j,2}$ | $\lvert v_{j,2} - v^*_{j,2}\rvert$ |
|---|---|---|---|---|
| 1 | 1 | $1.601 \cdot 10^{-9}$ | 3 | $2.792 \cdot 10^{-7}$ |
| 2 | 1.95 | $2.677 \cdot 10^{-7}$ | 2 | $5.788 \cdot 10^{-12}$ |
| 3 | 1.1 | $5.524 \cdot 10^{-9}$ | 0.4 | $1.197 \cdot 10^{-13}$ |
| 4 | 4 | $1.403 \cdot 10^{-13}$ | 3.005 | $1.68 \cdot 10^{-7}$ |
| 5 | 1.96 | $4.96 \cdot 10^{-7}$ | 4 | $7.994 \cdot 10^{-13}$ |

**Table 3** *Vertices of the unit-height polygon displayed in* Figure 7 (left) *and approximate reconstruction errors. The sampling step size is* $h = 0.4$*, and the measure of the angle between the first and the third sampling line is* $45°$*.*

| $j$ | $\lvert v_{j,1} - v^*_{j,1}\rvert$ | $\lvert v_{j,2} - v^*_{j,2}\rvert$ | $j$ | $\lvert v_{j,1} - v^*_{j,1}\rvert$ | $\lvert v_{j,2} - v^*_{j,2}\rvert$ |
|---|---|---|---|---|---|
| 1 | $1.014 \cdot 10^{-11}$ | $1.11 \cdot 10^{-14}$ | 7 | $1.94 \cdot 10^{-12}$ | $2.665 \cdot 10^{-15}$ |
| 2 | $1.399 \cdot 10^{-10}$ | $2.132 \cdot 10^{-14}$ | 8 | $3.758 \cdot 10^{-11}$ | $9.77 \cdot 10^{-15}$ |
| 3 | $8.304 \cdot 10^{-14}$ | $2.887 \cdot 10^{-15}$ | 9 | $1.052 \cdot 10^{-13}$ | 0 |
| 4 | $4.441 \cdot 10^{-16}$ | $9.992 \cdot 10^{-16}$ | 10 | $8.882 \cdot 10^{-16}$ | $1.11 \cdot 10^{-16}$ |
| 5 | $3.161 \cdot 10^{-10}$ | $8.882 \cdot 10^{-16}$ | 11 | $8.695 \cdot 10^{-11}$ | $4.885 \cdot 10^{-15}$ |
| 6 | $3.238 \cdot 10^{-11}$ | $9.326 \cdot 10^{-16}$ | 12 | $6.564 \cdot 10^{-12}$ | $3.197 \cdot 10^{-14}$ |

**Table 4** *Vertices of the unit-height polygon displayed in* Figure 7 (right) *and reconstruction errors. The sampling step size is* $h = 0.7$*, and the measure of the angle between the first and the third sampling line is* $56.75°$*.*

## 5 Conclusion and Outlook

In this paper, we have asked how to reconstruct polygonal shapes in the real plane by means of a smallest possible set of Fourier data. Answering this question, we have derived a novel algorithm for the unique reconstruction of polygonal shapes, based upon the Prony method. It suffices to take only $3N$ Fourier samples to recover polygons with $N$ vertices, where we have emphasized that this does not only works for convex polygons but also for non-convex polygons.

We have illustrated our proposed approach with numerical experiments where we use exact (simulated) Fourier data. However, in the case of noisy measurements, the performance of the reconstruction can be greatly improved if a larger number of Fourier data is available, see [8, 19, 21]. In particular, for small data sets we recommend the preprocessing step of data filtering presented in [8].

It would be of great interest to generalize, and maybe combine, the approaches for the reconstruction of non-uniform translates in [20, Section 4.2], which can be generalized to a $d$-variate setting, and the reconstruction of polygonal shapes in Section 3 of this paper such that more general functions and shapes can be considered. For example, the *shape from moments problem*, see [7, 9, 17], is extended from polygons to algebraic curves in [11].

In this context, the question arises if an approach for a $d$-dimensional setting can be combined with the reconstruction of polygonal shapes such that we can transfer it to the reconstruction of polytopes. This is, for example, treated in [3, 10], where integral moments of a $d$-dimensional polytope are used, together with the Prony method, in order to reconstruct the polytope. But there are considered only convex polytopes. Thus, it would be interesting to extend the theory developed in this paper.

## Acknowledgement

The research in this paper is partially funded by the project PL 170/16-1 of the German Research Foundation (DFG). This is gratefully acknowledged.

## References

[1] R. N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, New York, 3rd edition, 2000.

[2] M. Buhmann and A. Pinkus. On a Recovery Problem. *Ann. Numer. Math.*, 4:129–142, 1997.

[3] M. Collowald, A. Cuyt, E. Hubert, W.-S. Lee, and O. S. Celis. Numerical reconstruction of convex polytopes from directional moments. *Adv Comput Math*, 2015. DOI 10.1007/s10444-014-9401-0.

[4] A. Cuyt, G. Golub, P. Milanfar, and B. Verdonk. Multidimensional integral inversion, with applications in shape reconstruction. *SIAM J. Sci. Comput.*, 27(3):1058–1070, 2005.

[5] G. C. F. M. R. de Prony. 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. *Journal de l'École Polytechnique*, 1:24–76, 1795. Quoted from [18].

[6] S. Durocher. Graph-Theoretic and Geometric Algorithms Associated with Moment-based Polygon Reconstruction. Master's thesis, Department of Computer Science, University of British Columbia, 1999.

[7] M. Elad, P. Milanfar, and G. H. Golub. Shape from Moments — An Estimation Theory Perspective. *IEEE Trans. Signal Process.*, 52(7):1814–1829, 2004.

[8] F. Filbir, H. N. Mhaskar, and J. Prestin. On the Problem of Parameter Estimation in Exponential Sums. *Constr. Approx.*, 35(3):323–343, 2012.

[9] G. H. Golub, P. Milanfar, and J. Varah. A stable numerical method for inverting shape from moments. *SIAM J. Sci. Comput.*, 21(4):1222–1243, 1999.

[10] N. Gravin, J. Lasserre, D. V. Pasechnik, and S. Robins. The Inverse Moment Problem for Convex Polytopes. *Discrete Comput. Geom.*, 48(3):596–621, 2012.

[11] B. Gustafsson, C. He, P. Milanfar, and M. Putinar. Reconstructing planar domains from their moments. *Inverse Problems*, 16(4):1053–1070, 2000.

[12] F. B. Hildebrand. *Introduction to numerical analysis*. Dover Publications, New York, 2nd edition, 1987. Unabridged, slightly corrected republication.

[13] Y. Hua and T. K. Sarkar. Matrix Pencil Method for Estimatin Parameter of Exponentially Damped/Undamped Sinusoids in Noise. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(5):814–824, 1990.

[14] J. Komrska. Simple derivation of formulas for Fraunhofer diffraction at polygonal apertures. *J. Opt. Soc. Amer.*, 72(10):1382–1384, 1982.

[15] H. N. Mhaskar and J. Prestin. On the detection of singularities of a periodic function. *Adv. Comput. Math.*, 12(2–3):95–131, 2000.

[16] H. N. Mhaskar and J. Prestin. On Local Smoothness Classes of Periodic Functions. *J. Fourier Anal. Appl.*, 11(3):353–373, 2005.

[17] P. Milanfar, G. C. Verghese, W. C. Karl, and A. S. Willsky. Reconstructing Polygons from Moments with Connections to Array Processing. *IEEE Trans. Signal Process.*, 43(2):432–443, 1995.

[18] T. Peter. *Generalized Prony Method*. Der Andere Verlag, Uelvesbüll, 2014.

[19] T. Peter, D. Potts, and M. Tasche. Nonlinear approximation by sums of exponentials and translates. *SIAM J. Sci. Comput.*, 33(4):1920–1947, 2011.

[20] G. Plonka and M. Wischerhoff. How many Fourier samples are needed for real function reconstruction? *J. Appl. Math. Comput.*, 42(1–2):117–137, 2013.

[21] D. Potts and M. Tasche. Parameter estimation for exponential sums by approximate Prony method. *Signal Process.*, 90(5):1631–1642, 2010.

[22] D. Potts and M. Tasche. Nonlinear approximation by sums of nonincreasing exponentials. *Appl. Anal.*, 90:609–626, 2011.

[23] D. Potts and M. Tasche. Parameter estimation for multivariate exponential sums. *Electron. Trans. Numer. Anal.*, 40:204–224, 2013.

[24] D. Potts and M. Tasche. Parameter estimation for nonincreasing exponential sums by prony-like methods. *Linear Algebra Appl.*, 439(4):1024–1039, 2013.

[25] R. Roy and T. Kailath. ESPRIT — Estimation of Signal Parameters Via Rotational Invariance Techniques. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(7):984–995, 1989.

[26] R. O. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Trans. Antennas Propagat.*, 34(3):276–280, 1986.

[27] M. Wischerhoff. Real function reconstruction from sparse fourier samples. *PAMM. Proc. Appl. Math. Mech.*, 13(1):491–492, 2013.