Numer Algor
DOI 10.1007/s11075-017-0370-5

ORIGINAL PAPER

Deterministic sparse FFT for M-sparse vectors

Gerlind Plonka[1] · Katrin Wannenwetsch[1] · Annie Cuyt[2] · Wen-shin Lee[2]

Received: 24 January 2017 / Accepted: 26 June 2017

Abstract In this paper, we derive a new deterministic sparse inverse fast Fourier transform (FFT) algorithm for the case that the resulting vector is sparse. The sparsity needs not to be known in advance but will be determined during the algorithm. If the vector to be reconstructed is M-sparse, then the complexity of the method is at most $\mathcal{O}(M^2 \log N)$ if $M^2 < N$ and falls back to the usual $\mathcal{O}(N \log N)$ algorithm for $M^2 \geq N$. The method is based on the divide-and-conquer approach and may require the solution of a Vandermonde system of size at most $M \times M$ at each iteration step j if $M^2 < 2^j$. To ensure the stability of the Vandermonde system, we propose to employ a suitably chosen parameter σ that determines the knots of the Vandermonde matrix on the unit circle.

Keywords Sparse signals · Vandermonde matrices · Discrete Fourier transform · Sparse FFT

✉ Gerlind Plonka
plonka@math.uni-goettingen.de

Katrin Wannenwetsch
k.wannenwetsch@math.uni-goettingen.de

Annie Cuyt
annie.cuyt@uantwerpen.be

Wen-shin Lee
wen-shin.lee@uantwerpen.be

[1] Institute for Numerical and Applied Mathematics, University of Göttingen, Lotzestr. 16-18, 37083 Göttingen, Germany

[2] Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium

Published online: 05 July 2017

Mathematics Subject Classification (2010) 65T50 · 42A38

1 Introduction

Usual fast Fourier transform (FFT) algorithms require $\mathcal{O}(N \log N)$ operations for the discrete Fourier transform of length N. But assuming that some further a priori information about the resulting vector is available, the question arises, whether this computation can be done even faster.

Let us assume that $\mathbf{x} = (x_k)_{k=0}^{N-1} \in \mathbb{C}^N$ is a vector of length $N = 2^J$, and denote by $\widehat{\mathbf{x}} := \mathbf{F}_N \mathbf{x} \in \mathbb{C}^N$ its discrete Fourier transform, where $\mathbf{F}_N := (\omega_N^{jk})_{j,k=0}^{N-1}$ with $\omega_N := \mathrm{e}^{-2\pi \mathrm{i}/N}$ is the Fourier matrix of order N. In this paper, we will derive a stable deterministic algorithm to reconstruct $\mathbf{x}$ from $\widehat{\mathbf{x}}$ using the assumption that $\mathbf{x}$ is an M-sparse vector. The proposed algorithm uses only at most $M \log N$ components of the vector $\widehat{\mathbf{x}} = (\widehat{x}_k)_{k=0}^{N-1}$. We do not assume that the possible sparsity $M < N$ is known in advance. Applying an iterative procedure, we will adaptively choose the components $\widehat{x}_k$ being used for the reconstruction of $\mathbf{x}$. The number of needed values at each level will depend on the sparsity of the periodization of $\mathbf{x}$ found so far and thus is always at most M. In order to compute a new periodization of $\mathbf{x}$ of double length from the preceding one, we have to solve a Vandermonde system of size at most $\mathcal{O}(M)$, where the system matrix is a special partial matrix of the Fourier matrix. The arithmetical complexity will be $\mathcal{O}(\min\{M^2 \log N, N \log N\})$, where the $\mathcal{O}$ constant is small. Particularly, if no (exploitable) sparsity of $\mathbf{x}$ is recognized, then we fall back to the usual inverse FFT.

While the sparse FFT algorithm is described here for the inverse transform, the idea can be simply transferred to the case when $\mathbf{x} \in \mathbb{C}^N$ is given and $\widehat{\mathbf{x}} \in \mathbb{C}^N$ has to be computed and is a priori known to be sparse.

Sparse FFT methods can be applied in many different applications, where it is a priori known that the resulting signal in time/space or frequency domain is (approximately) sparse, as, e.g., for computing cross-correlation signals for GPS systems [9] or pattern matching problems, see, e.g., [12].

Our proposed sparse FFT algorithm to compute the sparse vector $\mathbf{x}$ is completely deterministic and exact if for each nonzero (significant) component x_k of $\mathbf{x}$ the sums $x_{k \bmod 2^j}^{(j)} := \sum_{\ell=0}^{N/2^j-1} x_{k+2^j\ell}$ do not vanish for all $j = 0, \ldots, J-1$. For randomly chosen signals, this condition is satisfied with high probability and it is obviously true, if, e.g., all nonzero components of $\mathbf{x}$ lie in the same quadrant of the complex plane.

In recent years, different approaches have been suggested to derive sparse FFT algorithms. Usual assumptions on the signal to be recovered are, e.g., sparsity or a small amount of significant signal components. Often, further a priori knowledge is used, as, e.g., that components to be recovered are from a finite range of real values (see, e.g., [8]), or that the significant components of $\mathbf{x}$ are clustered, see [2, 18, 19].

There exist deterministic [1, 2, 10, 11, 15, 17–19] and randomized methods [8, 16, 20] for sparse FFT. We also refer to the recent review [7] that describes some basic

principles of sparse FFT algorithms. Randomized methods are usually faster but do not always produce correct results.

Compared to other deterministic sparse FFT algorithms based on combinatorial approaches, see [1, 2, 10, 11, 15], our method has the advantage that the recovery of $\mathbf{x}$ only employs components of the DFT vector $\widehat{\mathbf{x}} \in \mathbb{C}^N$ and is therefore directly comparable to the usual discrete Fourier transform. The reconstruction based on Prony's method in [17] may still suffer from occurring numerical instabilities. In [5, 6], the ill-conditioning is alleviated with high probability by a random redistribution of the nodes on the unit circle. All previous sparse FFT approaches (except for [19]) need a priori knowledge on the sparsity M or need to run repeatedly for different guesses of M, while our approach automatically recognizes a possible sparsity of the resulting vector.

Indeed, our algorithm can be seen as a generalization of [19], where non-negative vectors with short support have been computed by (inverse) sparse FFT. However, this generalization is essential, the transfer from one short support to general sparsity of a vector requires new ideas for stable recovery of $\mathbf{x}$.

The paper is structured as follows. In Section 2, we derive the main algorithm that is based on a multi-scale reconstruction technique. In order to determine a well-conditioned coefficient matrix to compute the next periodization $\mathbf{x}^{(j+1)} \in \mathbb{C}^{2^{j+1}}$ of $\mathbf{x}$ at level j, we restrict ourselves to matrices with a Vandermonde structure that are determined by the indices of the nonzero entries found so far and on one further parameter σ that we have to choose suitably. Section 3 is devoted to some general considerations on the problem to find a suitable σ and also answers the question which improvement can be expected by employing only one single parameter σ. Assuming that the indices $0 \le n_1 < \ldots < n_{M_j} < 2^j$ of nonzero entries are known, we show that σ should be chosen in a way such that the new knots $\omega_{2^j}^{\sigma n_k}$ determining the Vandermonde matrix are well distributed on the unit circle. This can be achieved by maximizing the minimal distance between two neighboring values σn_k on the 2^j-periodic interval. It is also shown that, in rare cases, even the optimal parameter only leads to the minimal distance $2^j/M^2$ while the optimal distance in the case of M equidistant values is $2^j/M$. Section 4 provides further ideas on efficient computation of σ. We present some insights on how to determine σ and propose two approaches to find a suitable σ with a computational effort not exceeding $\mathcal{O}(M^2)$. Moreover, we show that if the sparsity M_j of the periodized vector does not change compared to the previous iteration step, σ_j can just be taken as $2\sigma_{j-1}$. The ideas are illustrated by different examples. In Section 5, numerical experiments are presented.

2 Multi-scale reconstruction

Assume that $\mathbf{x} \in \mathbb{C}^N$ is M-sparse, where $0 \le M \le N$, i.e., $\mathbf{x}$ possesses M significant nonzero components. Let $\widehat{\mathbf{x}} = \mathbf{F}_N \mathbf{x} = (\widehat{x}_k)_{k=0}^{N-1}$ be the discrete Fourier transform of $\mathbf{x}$. Assume further that $N := 2^J$ with some $J > 0$. We want to derive an iterative

stable procedure to reconstruct $\mathbf{x}$ from adaptively chosen Fourier entries of $\widehat{\mathbf{x}}$. For that purpose, we consider the periodized vectors as follows:

$$\mathbf{x}^{(j)} = (x_k^{(j)})_{k=0}^{2^j-1} := \left(\sum_{\ell=0}^{2^{J-j}-1} x_{k+2^j\ell} \right)_{k=0}^{2^j-1} \in \mathbb{C}^{2^j}, \qquad j = 0, \ldots, J. \tag{2.1}$$

Hence, $\mathbf{x}^{(0)} = \sum_{k=0}^{N-1} x_k$ is the sum of all components, $\mathbf{x}^{(1)} = (\sum_{\ell=0}^{N/2-1} x_{2\ell}, \sum_{\ell=0}^{N/2-1} x_{2\ell+1})^T$, and $\mathbf{x}^{(J)} = \mathbf{x}$. As already mentioned in the introduction, we assume that no cancellation appears in the periodic vectors, i.e., for each significant component $x_k \neq 0$ of $\mathbf{x}$, we have the following:

$$x^{(j)}_{k \bmod 2^j} \neq 0 \qquad \text{for all } j = 0, \ldots, J-1. \tag{2.2}$$

Numerically, we suppose that $|x^{(j)}_{k \bmod 2^j}| > \epsilon$ for a fixed shrinkage constant ϵ. Throughout the paper, we assume that Eq. (2.2) is satisfied.

We recall the following relationship from [18] for the discrete Fourier transform of the vectors $\mathbf{x}^{(j)}$, showing that the components of $\widehat{\mathbf{x}}^{(j)}$ are already given by components of $\widehat{\mathbf{x}}$.

Lemma 2.1 *For the vectors* $\mathbf{x}^{(j)} \in \mathbb{C}^{2^j}$, $j = 0, \ldots, J$, *in* Eq. (2.1), *we have the discrete Fourier transform*

$$\widehat{\mathbf{x}}^{(j)} := \mathbf{F}_{2^j}\, \mathbf{x}^{(j)} = \left(\widehat{x}_{2^{J-j}k}\right)_{k=0}^{2^j-1},$$

where $\widehat{\mathbf{x}} = (\widehat{x}_k)_{k=0}^{N-1} = \mathbf{F}_N\, \mathbf{x}$ *is the discrete Fourier transform of* $\mathbf{x} \in \mathbb{C}^N$.

Idea of the algorithm Assume now that $\mathbf{x} \in \mathbb{C}^N$ is M-sparse but the sparsity $0 \leq M \leq N$ is not known a priori.

Step 0. We start by considering $\mathbf{x}^{(0)}$. Obviously,

$$\mathbf{x}^{(0)} = \sum_{k=0}^{N-1} x_k = \widehat{x}_0.$$

From Eq. (2.2), we can conclude that for $\widehat{x}_0 = 0$ the vector $\mathbf{x}$ is the zero-vector, i.e., it is 0-sparse.

Step 1. Having found $\mathbf{x}^{(0)} = \widehat{x}_0 \neq 0$, we proceed and consider $\mathbf{x}^{(1)}$. Obviously, we have $\mathbf{x}^{(1)} = (x_0^{(1)}, x_1^{(1)})^T$, where $x_0^{(1)} + x_1^{(1)} = \mathbf{x}^{(0)} = \widehat{x}_0$ is already known. Choosing now the Fourier component $\widehat{x}_1^{(1)} = \widehat{x}_{N/2} = x_0^{(1)} - x_1^{(1)}$ and using $x_1^{(1)} = \mathbf{x}^{(0)} - x_0^{(1)}$, we obtain $\widehat{x}_{N/2} = 2x_0^{(1)} - \mathbf{x}^{(0)}$, i.e.,

$$x_0^{(1)} = \frac{1}{2}\left(\mathbf{x}^{(0)} + \widehat{x}_{N/2}\right), \qquad x_1^{(1)} = \frac{1}{2}\left(\mathbf{x}^{(0)} - \widehat{x}_{N/2}\right) = \mathbf{x}^{(0)} - x_0^{(1)}.$$

If $x_0^{(1)} = 0$, we can conclude that all even components of $\mathbf{x}$ vanish, and we do not need to consider them further. If $x_1^{(1)} = 0$, it follows analogously that all odd components of $\mathbf{x}$ are zero.

Step $j+1$. Assume now that we have computed $\mathbf{x}^{(j)} \in \mathbb{C}^{2^j}$ at the j-th level of iteration, and let $M_j \le 2^j$ be the found sparsity of $\mathbf{x}^{(j)}$. Obviously, we have $M_j \le M$. Further, let

$$0 \le n_1^{(j)} < n_2^{(j)} < \ldots < n_{M_j}^{(j)} \le 2^j - 1$$

be the indices of the corresponding nonzero components of $\mathbf{x}^{(j)}$. Observe that generally for $\mathbf{x}^{(j+1)} = \left(x_k^{(j+1)}\right)_{k=0}^{2^{j+1}-1}$ we have

$$x_k^{(j+1)} + x_{k+2^j}^{(j+1)} = x_k^{(j)}, \qquad k = 0, \ldots, 2^j - 1. \tag{2.3}$$

Hence, in order to compute now $\mathbf{x}^{(j+1)}$, we only need to consider the $2M_j$ components $x_{n_k}^{(j+1)}$ and $x_{n_k+2^j}^{(j+1)}$ for $k = 1, \ldots, M_j$ as candidates for nonzero entries in $\mathbf{x}^{(j+1)}$ while all other components of $\mathbf{x}^{(j+1)}$ can be assumed to be zero. Moreover, Eq. (2.3) provides already M_j conditions on these values, so that we need only M_j suitably chosen further Fourier data to recover $\mathbf{x}^{(j+1)}$. In particular, we have

Theorem 2.2 *Let $\mathbf{x}^{(j)}$, $j = 0, \ldots, J$, be the vectors defined in* Eq. (2.1) *satisfying* Eq. (2.2). *Then, for each $j = 0, \ldots, J-1$, we have the following: If $\mathbf{x}^{(j)} \in \mathbb{C}^{2^j}$ is M_j-sparse with support indices $0 \le n_1^{(j)} < n_2^{(j)} < \ldots < n_{M_j}^{(j)} \le 2^j - 1$, then the vector $\mathbf{x}^{(j+1)}$ can be uniquely recovered from $\mathbf{x}^{(j)}$ and M_j components $\widehat{x}_{k_1}, \ldots, \widehat{x}_{k_{M_j}}$ of $\widehat{\mathbf{x}} = \mathbf{F}_N \mathbf{x}$, where the indices $k_1, \ldots, k_{M_j}$ are taken from the set $\{2^{J-j-1}(2\ell+1) : \ell = 0, \ldots 2^j - 1\}$ such that the matrix*

$$\left(\omega_N^{k_p n_r^{(j)}}\right)_{p,r=1}^{M_j} = \left(\mathrm{e}^{-2\pi \mathrm{i} k_p n_r^{(j)}/N}\right)_{p,r=1}^{M_j} \in \mathbb{C}^{M_j \times M_j}$$

is invertible.

Proof Using the vector notation $\mathbf{x}_0^{(j+1)} := \left(x_k^{(j+1)}\right)_{k=0}^{2^j-1}$ and $\mathbf{x}_1^{(j+1)} := \left(x_k^{(j+1)}\right)_{k=2^j}^{2^{j+1}-1}$, we have

$$\mathbf{x}^{(j)} = \mathbf{x}_0^{(j+1)} + \mathbf{x}_1^{(j+1)} \tag{2.4}$$

such that we only need to compute $\mathbf{x}_0^{(j+1)}$ in order to recover $\mathbf{x}^{(j+1)}$. By Lemma 2.1, we find

$$\left(\widehat{x}_{2^{J-j-1}k}\right)_{k=0}^{2^{j+1}-1} = \widehat{\mathbf{x}}^{(j+1)} = \mathbf{F}_{2^{j+1}} \begin{pmatrix} \mathbf{x}_0^{(j+1)} \\ \mathbf{x}_1^{(j+1)} \end{pmatrix} = \mathbf{F}_{2^{j+1}} \begin{pmatrix} \mathbf{x}_0^{(j+1)} \\ \mathbf{x}^{(j)} - \mathbf{x}_0^{(j+1)} \end{pmatrix}$$

$$= \left(\omega_{2^{j+1}}^{k\ell}\right)_{k=0,\ell=0}^{2^{j+1}-1,2^j-1} \mathbf{x}_0^{(j+1)} + \left((-1)^k \omega_{2^{j+1}}^{k\ell}\right)_{k=0,\ell=0}^{2^{j+1}-1,2^j-1} \left(\mathbf{x}^{(j)} - \mathbf{x}_0^{(j+1)}\right). \tag{2.5}$$

We simply observe that the even indexed entries $\widehat{x}_{2\ell}^{(j+1)} = \widehat{x}_{\ell}^{(j)} = \widehat{x}_{2^J-j\,\ell}$ do not further contribute to the recovery of the vector $\mathbf{x}_0^{(j+1)}$ but are determined already from $\mathbf{x}^{(j)}$ that is known from the previous step. Let now $0 \le n_1^{(j)} < n_1^{(j)} < \ldots < n_{M_j}^{(j)} \le 2^j - 1$ be the indices of the nonzero entries of $\mathbf{x}^{(j)}$. Then, by Eq. (2.4) also $\mathbf{x}_0^{(j+1)}$ can have nonzero entries only at these components. We restrict the vectors accordingly to

$$\widetilde{\mathbf{x}}_0^{(j+1)} := \left(x_{n_r^{(j)}}^{(j+1)}\right)_{r=1}^{M_j} \in \mathbb{C}^{M_j}, \qquad \widetilde{\mathbf{x}}^{(j)} := \left(x_{n_r^{(j)}}^{(j)}\right)_{r=1}^{M_j} \in \mathbb{C}^{M_j}.$$

Further, let $k_1, \ldots, k_{M_j}$ be pairwise different indices from $\{2^{J-j-1}(2\ell+1) : \ell = 0, \ldots 2^j - 1\}$, i.e., we have $k_p := 2^{J-j-1}(2\kappa_p + 1)$ with $\kappa_p \in \{0, \ldots, 2^j - 1\}$ for $p = 1, \ldots, M_j$. We now restrict the system in Eq. (2.5) to the M_j equations corresponding to these indices $k_1, \ldots, k_{M_j}$ and find

$$\widehat{\mathbf{z}}^{(j+1)} := \begin{pmatrix} \widehat{x}_{k_1} \\ \vdots \\ \widehat{x}_{k_{M_j}} \end{pmatrix} = \begin{pmatrix} \widehat{x}_{2\kappa_1+1}^{(j+1)} \\ \vdots \\ \widehat{x}_{2\kappa_{M_j}+1}^{(j+1)} \end{pmatrix} = \mathbf{A}^{(j+1)}\,\widetilde{\mathbf{x}}_0^{(j+1)} - \mathbf{A}^{(j+1)}\left(\widetilde{\mathbf{x}}^{(j)} - \widetilde{\mathbf{x}}_0^{(j+1)}\right), \tag{2.6}$$

where

$$\mathbf{A}^{(j+1)} = \left(\omega_N^{k_p n_r^{(j)}}\right)_{p,r=1}^{M_j} = \left(\omega_{2^j}^{\kappa_p n_r^{(j)}}\right)_{p,r=1}^{M_j} \operatorname{diag}\left(\omega_{2^{j+1}}^{n_1^{(j)}}, \ldots, \omega_{2^{j+1}}^{n_{M_j}^{(j)}}\right). \tag{2.7}$$

If $\mathbf{A}^{(j+1)}$ resp. $(\omega_{2^j}^{\kappa_p n_r^{(j)}})_{p,r=1}^{M_j}$ is invertible, it follows from Eq. (2.6) that

$$\mathbf{A}^{(j+1)}\widetilde{\mathbf{x}}_0^{(j+1)} = \frac{1}{2}\left(\widehat{\mathbf{z}}^{(j+1)} + \mathbf{A}^{(j+1)}\widetilde{\mathbf{x}}^{(j)}\right), \tag{2.8}$$

and we can recover $\widetilde{\mathbf{x}}_0^{(j+1)}$ by solving this $(M_j \times M_j)$ equation system. Hence, the components of $\mathbf{x}^{(j+1)} \in \mathbb{C}^{2^{j+1}}$ are given by

$$x_\ell^{(j+1)} = \begin{cases} \left(\widetilde{\mathbf{x}}_0^{(j+1)}\right)_k & \text{for } \ell = n_k^{(j)}, \\ \left(\widetilde{\mathbf{x}}^{(j)}\right)_k - \left(\widetilde{\mathbf{x}}_0^{(j+1)}\right)_k & \text{for } \ell = n_k^{(j)} + 2^j, \\ 0 & \text{else.} \end{cases}$$

□

Theorem 2.2 yields that we essentially have to solve a linear system in Eq. (2.8) of size M_j in order to compute $\mathbf{x}^{(j+1)}$ from $\mathbf{x}^{(j)}$. We summarize our findings in the following Algorithm, where we use the conventional FFT at each step as long as this is more efficient than solving the system in Eq. (2.8).

Algorithm 2.3 *(Reconstruction of a vector from Fourier measurements)*

Input : $N = 2^J$ *(length of the vector* $\mathbf{x}$*);*
possible access to Fourier values $\widehat{x}_k$, $k = 0, \ldots, N-1$*;*
shrinkage constant ϵ.

Set $M := 0$ *and* $K := \{0\}$. *Choose the Fourier value* $\widehat{x}_0$.
If $|\widehat{x}_0| < \epsilon$, *then* $\mathbf{x} = \boldsymbol{0}$ *and* $I^{(J)} = \emptyset$.
If $|\widehat{x}_0| \geq \epsilon$, *then*

1. *Set* $M := 1$, $I^{(0)} := \{0\}$ *and* $\widetilde{\mathbf{x}}^{(0)} = \widehat{x}_0$.
2. *For* $j = 0$ *to* $J-1$ *do*
   If $M^2 \geq 2^j$, *then*
   Choose $\widehat{\boldsymbol{z}}^{(j+1)} := \left(\widehat{x}^{(j+1)}_{2p+1}\right)_{p=0}^{2^j-1} = \left(\widehat{x}_{2^{J-j-1}(2p+1)}\right)_{p=0}^{2^j-1} \in \mathbb{C}^M$ *and solve the linear system*

$$\mathbf{F}_{2^j} diag\Big((\omega_{2^{j+1}}^k)_{k=0}^{2^j-1}\Big)\mathbf{x}_0^{(j+1)} = \frac{1}{2}\left(\widehat{\boldsymbol{z}}^{(j+1)} + \mathbf{F}_{2^j} diag\Big((\omega_{2^{j+1}}^k)_{k=0}^{2^j-1}\Big)\mathbf{x}^{(j)}\right)$$

   using an FFT algorithm
   else

   2.1 *Choose* M *indices* $k_p = 2^{J-j-1}(2\kappa_p + 1)$ *with* $\kappa_p \in \{0, \ldots, 2^j - 1\}$ *for* $p = 1, \ldots, M$ *such that*

$$\mathbf{A}^{(j+1)} := \left(\omega_N^{k_p \ell}\right)_{p=1,\ldots,M;\, \ell \in I^{(j)}}$$

   is well-conditioned and set $K := K \cup \{k_1, \ldots, k_M\}$.
   2.2 *Choose the Fourier values* $\widehat{\boldsymbol{z}}^{(j+1)} := (\widehat{x}_{k_p})_{p=1}^M \in \mathbb{C}^M$ *and solve the linear system*

$$\mathbf{A}^{(j+1)}\widetilde{\mathbf{x}}_0^{(j+1)} = \frac{1}{2}\left(\widehat{\boldsymbol{z}}^{(j+1)} + \mathbf{A}^{(j+1)}\widetilde{\mathbf{x}}^{(j)}\right).$$

   2.3 *Set* $\widetilde{\mathbf{x}}_1^{(j+1)} := \widetilde{\mathbf{x}}^{(j)} - \widetilde{\mathbf{x}}_0^{(j+1)}$ *and* $\widetilde{\mathbf{x}}^{(j+1)} := \left(\left(\widetilde{\mathbf{x}}_0^{(j+1)}\right)^T, \left(\widetilde{\mathbf{x}}_1^{(j+1)}\right)^T\right)^T$.
   end (if)
   2.4 *Determine the set of active indices* $I^{(j+1)} \subset \left(I^{(j)} \cup (I^{(j)} + 2^j)\right)$ *by deleting all indices in* $I^{(j)} \cup (I^{(j)} + 2^j)$ *that correspond to entries in* $\widetilde{\mathbf{x}}^{(j+1)}$ *with modulus being smaller than* ϵ. *Set* $M := \#I^{(j+1)}$ *being the number of nonzero entries of* $\mathbf{x}^{(j+1)}$.

   end (do)

Output : $I^{(J)}$, *the set of active indices in* $\mathbf{x}$ *with* $M = \#I^{(J)}$*;*
$\widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^{(J)} = (x_k)_{k \in I^{(J)}}$, *the vector restricted to nonzero entries;*
K, *the index set of used Fourier values from* $\widehat{\mathbf{x}}$.

Note that the matrices $\mathbf{A}^{(j+1)}$ are just restrictions of the Fourier matrices $\mathbf{F}_N$ to the columns $n_1^{(j)}, \ldots n_{M_j}^{(j)}$ and the rows $k_1, \ldots, k_{M_j}$. Equivalently, they can be represented by Eq. (2.7) using a matrix product, where one factor is the restriction of $\mathbf{F}_{2^j}$ to the columns $n_1^{(j)}, \ldots, n_{M_j}^{(j)}$ and the rows $\kappa_1, \ldots, \kappa_{M_j}$, and the other factor is a unitary diagonal matrix. Observe that we can always choose $k_p = 2^{J-j-1}(2\kappa_p + 1)$ with $\kappa_p = p - 1$, for $p = 1, \ldots, M_j$ to ensure invertibility.

Example 2.4 Assume that we want to recover the 5-sparse vector $\mathbf{x} \in \mathbb{C}^{64}$ with $x_k = 1$ for $k \in I^{(6)} := \{1, 5, 6, 13, 59\}$. For the periodizations of $\mathbf{x}$, we find the index sets and the sparsities

$$\begin{aligned} I^{(0)} &= \{0\}, & M_0 &= 1; \\ I^{(1)} &= \{0, 1\}, & M_1 &= 2; \\ I^{(2)} &= \{1, 2, 3\}, & M_2 &= 3; \\ I^{(3)} &= \{1, 3, 5, 6\}, & M_3 &= 4; \\ I^{(4)} &= \{1, 5, 6, 11, 13\}, & M_4 &= 5; \\ I^{(5)} &= \{1, 5, 6, 13, 27\}, & M_5 &= 5. \end{aligned}$$

For $j = 0, 1, 2$, we have $M_j^2 \geq 2^j$ and therefore just apply the FFT of length 2^j to recover $\mathbf{x}^{(3)} = (0, 1, 0, 1, 0, 2, 1, 0)^T$. Although $M_3^2 = 2^4$, we apply for $j \geq 3$ the new approach for illustration. To recover $\mathbf{x}^{(4)} \in \mathbb{C}^{16}$, the index set of possible candidates for nonzero entries is $I^{(3)} \cup (I^{(3)} + 8) = \{1, 3, 5, 6, 9, 11, 13, 14\}$. We simply choose the indices $k_p = 2^{J-j-1}(2\kappa_p+1)$ with $\kappa_p = p-1$, for $p = 1, \ldots, M_j$ at each level. This choice relates to taking just the first M_j rows of $\mathbf{F}_{2^j}$ in Eq. (2.7). Here, we get for $j = 3$ the product of the restriction of $\mathbf{F}_8$ to the first four rows and the four columns $n_r^{(3)}$ from the set $I^{(3)}$ and a unitary diagonal matrix,

$$\mathbf{A}^{(4)} = \left(\omega_8^{(p-1)n_r^{(3)}}\right)_{p,r=1}^{4} \cdot \operatorname{diag}\left(\omega_{16}^1, \omega_{16}^3, \omega_{16}^5, \omega_{16}^6\right)$$

with condition number 2.69 to recover $\mathbf{x}^{(4)}$. Similarly, we find at the next iteration steps

$$\mathbf{A}^{(5)} = \left(\omega_{16}^{(p-1)n_r^{(4)}}\right)_{p,r=1}^{5} \cdot \operatorname{diag}\left(\omega_{32}^1, \omega_{32}^5, \omega_{32}^6, \omega_{32}^{11}, \omega_{32}^{13}\right)$$

with condition number 3.93 to recover $\mathbf{x}^{(5)}$ and

$$\mathbf{A}^{(6)} = \left(\omega_{32}^{(p-1)n_r^{(5)}}\right)_{p,r=1}^{5} \cdot \operatorname{diag}\left(\omega_{64}^1, \omega_{64}^5, \omega_{64}^6, \omega_{64}^{13}, \omega_{64}^{27}\right)$$

with $\operatorname{cond} \mathbf{A}^{(6)} = 35.57$ to recover $\mathbf{x}^{(6)}$. Thus, we have employed only the Fourier entries $\widehat{x}_{8k}$, $k = 0, \ldots, 7$, in the first three iteration steps ($j = 0, 1, 2$) to recover $\mathbf{x}^{(3)}$, the entries $\widehat{x}_{4(2k+1)}$, $k = 0, 1, 2, 3$, at level $j = 3$, $\widehat{x}_{2(2k+1)}$, $k = 0, 1, 2, 3, 4$, at level $j = 4$, and $\widehat{x}_{2k+1}$, $k = 0, 1, 2, 3, 4$, at level $j = 5$. Summing up, we have to employ 22 of the 64 Fourier components to recover $\mathbf{x}$, while the arithmetical complexity is governed by solving the three equation systems of size 4 resp. 5 with coefficient matrices $\mathbf{A}^{(4)}$, $\mathbf{A}^{(5)}$, and $\mathbf{A}^{(6)}$.

While the condition numbers of $\mathbf{A}^{(j)}$ in the small example above are moderate, the condition of $\mathbf{A}^{(j)}$ can be a serious problem for numerical stability in other examples. Recovering the 5-sparse vector $\mathbf{x} \in \mathbb{C}^{2048}$ with the same nonzero entries, i.e., $I^{(11)} := \{1, 5, 6, 13, 59\}$, we obtain with the procedure above at the last level $\mathbf{A}^{(10)} = \left(\omega_{1024}^{(p-1)n_r^{(10)}}\right)_{p,r=1}^{5} \cdot \operatorname{diag}\left(\omega_{2048}^{1}, \omega_{2048}^{5}, \omega_{2048}^{6}, \omega_{2048}^{13}, \omega_{2048}^{27}\right)$ with condition number $2.29 \cdot 10^7$. Therefore, it remains to answer the following question:

How should the $M = M_j$ indices $k_1, \ldots, k_{M_j}$ be chosen in dependence on the found index set $I^{(j)} = \{n_1^j, \ldots, n_{M_j}^{(j)}\}$ at the j-th iteration step such that the matrix $\mathbf{A}^{(j+1)}$ has a small condition number leading to a numerically stable algorithm?

We restrict the adaptive search for suitable indices $k_1, \ldots, k_{M_j}$ in a special way such that at each level j the first factor of the obtained coefficient matrix $\mathbf{A}^{(j+1)}$ in Eq. (2.7) is a Vandermonde matrix with knots on the unit circle. We introduce one parameter $\sigma \in \{1, \ldots, 2^j - 1\}$ such that the first factor of $\mathbf{A}^{(j+1)}$ is the restriction of $\mathbf{F}_{2^j}$ to the fixed columns $n_1^{(j)}, \ldots, n_{M_j}^{(j)}$ from the index set $I^{(j)}$ and the rows $0, \sigma, 2\sigma, \ldots (M_j - 1)\sigma$. More precisely, we consider $k_p = 2^{J-j-1}(2\kappa_p + 1)$ from Theorem 2.2 with $\kappa_p := \sigma(p-1) \bmod 2^j$ for $p = 1, \ldots, M_j$. The idea is now to choose $\sigma = \sigma_j \in \{1, \ldots, 2^j - 1\}$ such that $\mathbf{A}^{(j+1)}$ in Eq. (2.7) is well-conditioned. Observe that $\mathbf{A}^{(j+1)}$ is of the form

$$\mathbf{A}^{(j+1)} = \mathbf{V}_{M_j} \operatorname{diag}(\omega_{2^{j+1}}^{n_1^{(j)}}, \ldots, \omega_{2^{j+1}}^{n_{M_j}^{(j)}}),$$

with the Vandermonde matrix

$$\mathbf{V}_{M_j} := \mathbf{V}_{M_j}\left(\omega_{2^j}^{\sigma n_1^{(j)}}, \ldots, \omega_{2^j}^{\sigma n_{M_j}^{(j)}}\right) := \left(\omega_{2^j}^{\sigma(p-1)n_r^{(j)}}\right)_{p,r=1}^{M_j} = \left(\omega_{2^j}^{\kappa_p n_r^{(j)}}\right)_{p,r=1}^{M_j},$$

being determined by the knots $\omega_{2^j}^{\sigma n_1^{(j)}}, \ldots, \omega_{2^j}^{\sigma n_{M_j}^{(j)}}$. In Sections 3 and 4, we will discuss the problem of efficiently finding a suitable parameter σ and ensuring a stable reconstruction of $\mathbf{x}^{(j)}$ at each level in more detail.

For $\mathbf{A}^{(j+1)}$ in Eq. (2.7) being determined by a Vandermonde matrix, we can estimate the complexity of Algorithm 2.3. For a stable computation of the equation system in step 2.2, we may apply a QR decomposition to $\mathbf{A}^{(j+1)}$ with complexity of $\mathcal{O}(M^2)$ arithmetical operations as suggested, e.g., in [4]. Thus, as long as $M_j^2 \geq 2^j$, the FFT of length 2^j is more efficient than solving this system. As suggested in the algorithm, we employ the FFT if $M_j^2 \geq 2^j$ since the $\mathcal{O}$-constants of the FFT are very small. Algorithm 2.3 requires at most

$$\mathcal{O}(\min\{M^2(\lfloor \log M^2 \rfloor + \lceil \log_2(N/M^2) \rceil), N \log N)\}) = \mathcal{O}(\min\{M^2 \log_2 N, N \log N)\})$$

arithmetical operations, where the first $L = \lfloor \log M^2 \rfloor$ steps of the iteration use $\mathcal{O}(L2^L)$ operations while the remaining steps require $\mathcal{O}(M^2(J - L)) = \mathcal{O}(M^2(\log N - \log M^2))$ operations. A more detailed analysis of the arithmetical complexity is given in Section 5.

Remark 2.5 In case that the condition number of the quadratic Vandermonde matrix $\mathbf{V}_{M_j}\left(\omega_{2^j}^{\sigma n_1^{(j)}}, \ldots, \omega_{2^j}^{\sigma n_{M_j}^{(j)}}\right)$ is not small enough, we can add further lines and use a rectangular Fourier matrix. This means that we apply the rectangular Vandermonde matrix $\mathbf{V}_{M_j', M_j}^{(j)}(\sigma) = \left(\omega_{2^j}^{\sigma k n_p^{(j)}}\right)_{k=0,p=1}^{M_j'-1, M_j}$ with an improved condition number. The algorithm in [4] still provides a QR decomposition for rectangular Vandermonde matrices with complexity $\mathcal{O}(M^2)$ as long as $M_j' \le cM_j$ with a fixed constant c being independent of M_j.

3 Adaptive approach for stable reconstruction

Let us now consider the question, how to find an optimal $\sigma = \sigma_j$ at each iteration step in order to ensure a well-conditioned Vandermonde system. For simplicity, we neglect the subscripts j in this section and reformulate the problem as follows. Let $0 \le n_1 < n_2 < \ldots < n_M < N$ be a known set of indices. We want to find an optimal parameter σ such that the $M' \times M$ Vandermonde matrix

$$\mathbf{V}_{M',M}(\sigma) := \left(\omega_N^{\sigma n_k (p-1)}\right)_{p=1,k=1}^{M',M}$$

(with $N > M' \ge M$) determined by the knots $\omega_N^{\sigma n_k}$, $k = 1, \ldots, M$, has a suitably bounded condition number. At the same time, M' should stay in the same size as M in order to reduce the costs for solving a corresponding Vandermonde system.

It is well-known that the Vandermonde matrix $\mathbf{V}_{M,M}$ is invertible if and only if the support indices $(\sigma n_k \bmod N)$ are pairwise distinct for $k = 1, \ldots, M$. Thus, we can choose $\sigma = 1$ to ensure invertibility of $\mathbf{V}_{M,M}$. This choice is non-adaptive, it is not related to the knowledge of the index set $\{n_1, \ldots, n_M\}$. However, as seen in the previous section, this can lead to bad condition numbers.

We aim at deriving suitable conditions for the parameter σ to ensure a good condition of $\mathbf{V}_{M',M}$.

Indeed, the condition of the matrix $\mathbf{V}_{M',M}$ strongly depends on the distribution of the M values $\omega_N^{\sigma n_k}$ on the unit circle. The condition number of $\mathbf{V}_{M,M}$ can be even one, if and only if the values $\omega_N^{\sigma n_k}$ are equidistantly distributed on the unit circle, i.e., if M is a divisor of N and

$$\{\omega_N^{\sigma n_k} : k = 1, \ldots, M\} = \{c\, \omega_M^r : r = 1, \ldots, M\},$$

where c is a unitary constant, see [3].

Recall that the condition number of an $(M' \times M)$ matrix $\mathbf{V}_{M',M}(\sigma)$ based on the spectral norm is determined by

$$\kappa_2(\mathbf{V}_{M',M}(\sigma)) := \frac{\max\limits_{\mathbf{u}\in\mathbb{C}^M, \|\mathbf{u}\|_2=1} \|\mathbf{V}_{M',M}(\sigma)\,\mathbf{u}\|_2}{\min\limits_{\mathbf{u}\in\mathbb{C}^M, \|\mathbf{u}\|_2=1} \|\mathbf{V}_{M',M}(\sigma)\,\mathbf{u}\|_2}.$$

In order to bound the condition number of $\mathbf{V}_{M',M}$, an observation by Moitra [13] comes to our help. We slightly modify his result and give a different proof that directly adapts Hilbert's inequality in [14].

Theorem 3.1 *Let $0 \le n_1 < n_2 < \ldots < n_M < N$ be a given set of indices. For a given $\sigma \in \{1, \ldots, N-1\}$ let*

$$d_\sigma := \min_{1 \le k < \ell \le M} (\pm\sigma(n_\ell - n_k)) \bmod N \tag{3.1}$$

be the smallest (periodic) distance between two indices σn_ℓ and σn_k, and assume that $d_\sigma > 0$. Then, the condition number $\kappa_2(\mathbf{V}_{M',M}(\sigma))$ of the Vandermonde matrix $V_{M',M}(\sigma) := \left(\omega_N^{\sigma n_k \ell}\right)_{\ell=0,k=1}^{M'-1,M}$ satisfies

$$\kappa_2(\mathbf{V}_{M',M}(\sigma))^2 \le \frac{M' + N/d_\sigma}{M' - N/d_\sigma}, \tag{3.2}$$

provided that $M' > \frac{N}{d_\sigma}$.

Proof 1. Assume that $\tilde{n}_k := \frac{\sigma n_k \bmod N}{N}$ for $k = 1, \ldots, M$. By assumption, the values $\tilde{n}_k$ are distinct numbers in $[0, 1)$ and the minimal (cyclic) distance between two of these values is d_σ / N. Considering the matrix $(\mathbf{V}_{M',M}(\sigma))^* \mathbf{V}_{M',M}(\sigma) = (b_{k,\ell})_{k,\ell=1}^{M}$ we find

$$b_{k,\ell} = \sum_{r=0}^{M'-1} \mathrm{e}^{-2\pi \mathrm{i}(\tilde{n}_\ell - \tilde{n}_k) r} = \begin{cases} \frac{1 - \mathrm{e}^{-2\pi \mathrm{i}(\tilde{n}_\ell - \tilde{n}_k) M'}}{1 - \mathrm{e}^{-2\pi \mathrm{i}(\tilde{n}_\ell - \tilde{n}_k)}} & \tilde{n}_k \ne \tilde{n}_\ell, \\ M' & \tilde{n}_k = \tilde{n}_\ell, \end{cases}$$

i.e., we have

$$b_{k,\ell} = \mathrm{e}^{-2\pi \mathrm{i}(\tilde{n}_\ell - \tilde{n}_k)(M'-1)/2} D_{M'}\left(2\pi(\tilde{n}_\ell - \tilde{n}_k)\right),$$

where

$$D_{M'}(x) = \begin{cases} \frac{\sin(M'x/2)}{\sin(x/2)} & x \ne 0 \\ M' & x = 0 \end{cases}$$

denotes the Dirichlet kernel. Hence, the symmetric and positive semidefinite matrix

$$\mathbf{B}_M = \left(D_{M'}(2\pi(\tilde{n}_\ell - \tilde{n}_k))\right)_{\ell,k=1}^{M}$$

possesses the same eigenvalues as $(\mathbf{V}_{M',M}(\sigma))^* \mathbf{V}_{M',M}(\sigma)$ since

$$(\mathbf{V}_{M',M}(\sigma))^* \mathbf{V}_{M',M}(\sigma) = \operatorname{diag}\left(\mathrm{e}^{2\pi \mathrm{i} \tilde{n}_k (M'-1)/2}\right)_{k=1}^{M} \mathbf{B}_M \operatorname{diag}\left(\mathrm{e}^{-2\pi \mathrm{i} \tilde{n}_\ell (M'-1)/2}\right)_{\ell=1}^{M}.$$

Let us first consider the Frobenius norm $\|\mathbf{V}_{M',M}(\sigma)\|_F$. Since $\mathbf{B}_M(\ell, \ell) = M'$ for all $\ell = 1, \ldots, M$, it follows that

$$\|\mathbf{V}_{M',M}(\sigma)\|_F^2 = \operatorname{tr}(\mathbf{V}_{M',M}(\sigma))^* \mathbf{V}_{M',M}(\sigma)) = \operatorname{tr} \mathbf{B}_M = MM',$$

such that the spectral norm is bounded by $\|\mathbf{V}_{M',M}(\sigma)\|_2 \le \|\mathbf{V}_{M',M}(\sigma)\|_F = \sqrt{M'M}$.

2. We consider now for arbitrary $\mathbf{u} \in \mathbb{C}^M$

$$\begin{aligned}\mathbf{u}^T \mathbf{B}_M \overline{\mathbf{u}} &= \sum_{k=1}^{M} \sum_{\ell=1}^{M} u_k \overline{u}_\ell \, D_{M'}(2\pi(\tilde{n}_k - \tilde{n}_\ell)) \\ &= M' \sum_{k=1}^{M} |u_k|^2 + \sum_{\substack{k,\ell=1 \\ k \neq \ell}}^{M} u_k \overline{u}_\ell \, \frac{\sin(M'\pi(\tilde{n}_k - \tilde{n}_\ell))}{\sin(\pi(\tilde{n}_k - \tilde{n}_\ell))}.\end{aligned}$$

We recall the following result by Montgomery and Vaughan, see Theorem 1 in [14]. Let $0 \le x_1 < x_2 < \ldots < x_R < 1$ and $\delta = \min\{|(x_k - x_\ell) \bmod 1| : k, \ell = 1, \ldots, R, k \neq \ell\}$. Then

$$\left| \sum_{\substack{k,\ell=1 \\ k \neq \ell}}^{R} \frac{u_k \overline{u}_\ell}{\sin(\pi(x_k - x_\ell))} \right| \le \frac{1}{\delta} \sum_{k=1}^{R} |u_k|^2. \tag{3.3}$$

Using $\sin(M'\pi(\tilde{n}_k - \tilde{n}_\ell)) = \frac{1}{2\mathrm{i}}(\mathrm{e}^{M'\pi \mathrm{i}(\tilde{n}_k - \tilde{n}_\ell)} - \mathrm{e}^{-M'\pi \mathrm{i}(\tilde{n}_k - \tilde{n}_\ell)})$, we now apply the Eq. (3.3) twice, with u_k replaced by $u_k \mathrm{e}^{M'_j \pi \mathrm{i} \tilde{n}_k}$ and $u_k \mathrm{e}^{-M'_j \pi \mathrm{i} \tilde{n}_k}$, respectively. Thus, we obtain with $\delta = d_\sigma / N$

$$\left| \sum_{\substack{k,\ell=1 \\ k \neq \ell}}^{M} u_k \overline{u}_\ell \, \frac{\sin(M'\pi(\tilde{n}_k - \tilde{n}_\ell))}{\sin(\pi(\tilde{n}_k - \tilde{n}_\ell))} \right| \le \frac{N}{d_\sigma} \|\mathbf{u}\|_2.$$

This observation yields now

$$\begin{aligned}\|V_{M',M}(\sigma)\|_2^2 &= \max_{\mathbf{u} \in \mathbb{C}^M, \|\mathbf{u}\|_2 = 1} \mathbf{u}^T \mathbf{B}_M \overline{\mathbf{u}} \le M' + \frac{N}{d_\sigma}, \\ \|(V_{M',M}(\sigma))^{-1}\|_2^2 &= \left(\min_{\mathbf{u} \in \mathbb{C}^M, \|\mathbf{u}\|_2 = 1} \mathbf{u}^T \mathbf{B}_M \overline{\mathbf{u}} \right)^{-1} \le \left(M' - \frac{N}{d_\sigma} \right)^{-1}.\end{aligned}$$

Thus, we find the condition of $V_{M',M}(\sigma)$ as given in Eq. (3.2). □

The observation in Theorem 3.1 leads us to the problem to optimize for given indices $0 \le n_1 < n_2 < \ldots < n_M < N$ the parameter σ such that d_σ in Eq. (3.1) is maximized. Before presenting an algorithm to compute the optimal $\widetilde{\sigma}$, that satisfies

$$d_{\widetilde{\sigma}} := \max_{\sigma \in \{1, \ldots, N-1\}} d_\sigma \tag{3.4}$$

with d_σ defined in Eq. (3.1), we want to answer the question, how small $d_{\widetilde{\sigma}}$ can happen to be.

Theorem 3.2 *Let N be of the form $N = 2^J$, $J \in \mathbb{N}$, and $d = d_{\widetilde{\sigma}} := \max_{\sigma \in \{1, \ldots, N-1\}} d_\sigma$ with d_σ defined in Eq. (3.1) be the distance obtained for the optimally chosen parameter $\widetilde{\sigma}$. Then, we have*

$$\frac{N}{M^2} \le d \le \frac{N}{M}.$$

Proof 1. Considering the M knots $0 \le n_1 < n_2 < \ldots < n_M < N$ and the corresponding knots $\widetilde{\sigma} n_k \bmod N$, the distance d is obviously maximal if the knots $\widetilde{\sigma} n_k$ are equidistantly distributed on the (periodic) interval of length N, i.e., if $d = N/M$.

2. In order to show the lower bound, we apply a counting argument. Let us consider the set D of $M(M-1)$ distances $d_{\ell,k} := (n_\ell - n_k) \bmod N$ for $\ell, k = 1, \ldots, M$, $\ell \neq k$. Assume that ν indices n_ℓ are odd, and $M - \nu$ indices are even. Then, we obtain $2\nu(M-\nu)$ odd distances $d_{\ell,k}$ and $M(M-1) - 2\nu(M-\nu)$ even distances.

We assume now to the contrary that $d = d_{\widetilde{\sigma}} < N/M^2$. Thus, $d_\sigma < N/M^2$ for all $\sigma \in \{1, \ldots, N-1\}$, i.e., for each σ there exists a distance $d^\sigma_{\ell,k} \in D$ with $\sigma d^\sigma_{\ell,k} \bmod N < N/M^2$. We will show that this assumption leads to a contradiction.

For each distance $d_{\ell,k} \in D$, we now determine the largest possible number of odd integers σ such that $\sigma d_{\ell,k} \bmod N < N/M^2$. We distinguish between odd and even distances $d_{\ell,k}$ and consider two cases.

Case 1: If the fixed distance $d_{\ell,k} \in D$ is odd, then $\sigma d_{\ell,k} \bmod N$ is again odd, and for two pairwise different odd integers $\sigma_1, \sigma_2 \in \{1, \ldots, N-1\}$ the corresponding values $\sigma_1 d_{\ell,k} \bmod N$ and $\sigma_2 d_{\ell,k} \bmod N$ are different, since $\sigma_1 d_{\ell,k} = \sigma_2 d_{\ell,k} \bmod N$ yields $(\sigma_1 - \sigma_2) d_{\ell,k} = 0 \bmod N$ with the only solution $\sigma_1 = \sigma_2$.

Observe that there are $\lceil N/(2M^2) + 1/2 \rceil - 1$ (distinct) odd numbers in the interval $[0, N/M^2]$. Thus, there exist at most $\lceil N/(2M^2) + 1/2 \rceil - 1$ pairwise different odd integers σ in $\{1, \ldots, N-1\}$ such that $\sigma d_{\ell,k} \bmod N < N/M^2$.

Since we have at most $2\nu(M-\nu)$ distinct odd distances in D, there can be at most

$$2\nu(M-\nu)\left(\left\lceil \frac{N}{2M^2} + \frac{1}{2} \right\rceil - 1\right)$$

pairwise different odd integers σ in $\{1, \ldots, N-1\}$ such that the condition

$$\sigma d^\sigma_{\ell,k} \bmod N < N/M^2 \tag{3.5}$$

is satisfied with an odd distance $d_{\ell,k}$. Observe that this upper bound can be only achieved if all occurring odd distances $d_{\ell,k}$ in D are pairwise different.

Case 2: Let now $d_{\ell,k}$ be a fixed even distance. Then, there exists a positive integer μ such that $d_{\ell,k} = 2^\mu \tilde{d}_{\ell,k}$ and $\tilde{d}_{\ell,k}$ is odd. Thus, the condition $\sigma d_{\ell,k} \bmod N < \frac{N}{M^2}$ can be simplified to

$$\sigma \tilde{d}_{\ell,k} \bmod \frac{N}{2^\mu} < \frac{N}{2^\mu M^2}.$$

Hence, at most $\lceil N/(2^{\mu+1}M^2) + 1/2 \rceil - 1$ pairwise different odd integers σ in $\{1, \ldots, N-1\}$ can exist such that Eq. (3.5) is satisfied.

Since we have $M(M-1) - 2\nu(M-\nu)$ even distances $d_{\ell,k}$, it follows that at most

$$(M(M-1) - 2\nu(M-\nu))\left(\left\lceil \frac{N}{4M^2} + \frac{1}{2} \right\rceil - 1\right)$$

odd integers σ in $\{1, \ldots, N-1\}$ can exist, such that the condition (3.5) is satisfied with an even distance $d_{\ell,k}$. Observe that this upper bound can be only achieved if all occurring even distances $d_{\ell,k}$ are pairwise different and of the form $d_{\ell,k} = 2\tilde{d}_{\ell,k}$ with some odd $\tilde{d}_{\ell,k}$.

3. We now consider the following cases.

a) For $N > 4M^2$, the number of odd σ satisfying Eq. (3.5) for at least one distance $d_{\ell,k}$ is bounded by

$$\begin{aligned} &2\nu(M-\nu)\left(\frac{N}{2M^2}+\frac{1}{2}\right) + (M(M-1)-2\nu(M-\nu))\left(\frac{N}{4M^2}+\frac{1}{2}\right) \\ &= 2\nu(M-\nu)\frac{N}{4M^2} + M(M-1)\left(\frac{N}{4M^2}+\frac{1}{2}\right) \\ &\le \frac{M^2}{2}\frac{N}{4M^2} + \frac{N}{4} - \frac{N}{4M} + \frac{M^2}{2} - \frac{M}{2} \\ &< \frac{N}{8} + \frac{N}{4} + \frac{N}{8} - \frac{N}{4M} - \frac{M}{2} < \frac{N}{2}, \end{aligned}$$

where we have used that $2\nu(M-\nu) \le \frac{M^2}{2}$ for all $\nu \in \{0, \ldots, M\}$. Hence, not all values σ satisfy the condition (3.5) in this case.

b) For $3M^2 < N \le 4M^2$, we have

$$\left\lceil \frac{N}{2M^2}+\frac{1}{2} \right\rceil - 1 = 2, \qquad \left\lceil \frac{N}{4M^2}+\frac{1}{2} \right\rceil - 1 = 1.$$

Thus, the number of odd integers σ satisfying Eq. (3.5) is bounded by

$$\begin{aligned} &4\nu(M-\nu) + M(M-1) - 2\nu(M-\nu) = 2\nu(M-\nu) + M(M-1) \\ &\le \frac{M^2}{2} + M^2 - M = \frac{3M^2}{2} - M < \frac{N}{2}, \end{aligned}$$

i.e., not all values σ satisfy (3.5) also in this case.

c) For $2M^2 < N \le 3M^2$, it holds that

$$\left\lceil \frac{N}{2M^2}+\frac{1}{2} \right\rceil - 1 = 1, \qquad \left\lceil \frac{N}{4M^2}+\frac{1}{2} \right\rceil - 1 = 1,$$

and therefore the number of odd integers σ satisfying (3.5) is bounded by

$$2\nu(M-\nu) + M(M-1) - 2\nu(M-\nu) = M(M-1) < M^2 < \frac{N}{2}.$$

Hence, also for $2M^2 < N \le 3M^2$, not all values σ satisfy Eq. (3.5).

d) For $M^2 < N \le 2M^2$, we have

$$\left\lceil \frac{N}{2M^2}+\frac{1}{2} \right\rceil - 1 = 1, \qquad \left\lceil \frac{N}{4M^2}+\frac{1}{2} \right\rceil - 1 = 0.$$

and thus, the number of odd integers σ satisfying Eq. (3.5) is bounded by

$$2\nu(M-\nu) \le \frac{M^2}{2} < \frac{N}{2},$$

i.e., also in this case, not all values σ satisfy Eq. (3.5).

e) For $N \leq M^2$, no odd integer satisfies Eq. (3.5).

Thus, the number of odd integers σ for which there exists a $d_{\ell,k}^{\sigma} \in D$ such that Eq. (3.5) holds, is strictly smaller than $N/2$, i.e., there exists at least one odd $\sigma \in \{1, \ldots, N-1\}$ with

$$\sigma d_{\ell,k}^{\sigma} \bmod N \geq \frac{N}{M^2}$$

for all $d_{\ell,k} \in D$. □

Remark 3.3 1. The lower bound $d = N/M^2$ can be indeed achieved if $N = 2^J = 2^{\alpha} M^2$ for some $\alpha \in \mathbb{N}_0$, and if all distances of the form

$$d_{\ell,k} = 2^{\alpha}(2r+1), \qquad r = 0, \ldots, \frac{N}{2^{\alpha+2}} - 1$$

occur. Choosing, e.g., $N = 16$, $M = 4$, $\alpha = 0$, and the four indices $n_1 = 0$, $n_2 = 1$, $n_3 = 3$, $n_4 = 8$, then

$$D := \{d_{\ell,k} : \ell, k = 1, \ldots, 3;\ \ell < k\} = \{1, 2, 3, 5, 7, 8\}$$

contains all odd numbers in $\{0, \ldots, N/2\}$ and we find $d = N/M^2 = 1$.

2. Observe that this case $d = N/M^2$ is very rare. It occurs only for very special choices of indices $\{n_k\}_{k=1}^{M}$ (as well as its shifts $\{n_k + \ell\}_{k=1}^{M}$, $\ell = 0, \ldots, N-1$, and shifted reflections $\{(N - n_k) + \ell\}_{k=1}^{M}$, $\ell = 0, \ldots, N-1$. In the above case $N = 16$, $M = 4$, there are $\binom{N}{4} = 1820$ possibilities to fix four (ordered) indices, where $d = N/M^2 = 1$ only occurs in 128 cases.

4 Efficient parameter computation

In this section, we will derive a method to compute the optimal parameter $\tilde{\sigma}$ such that the optimization problem

$$\tilde{\sigma} = \operatorname*{argmax}_{\sigma \in \{1, \ldots, N-1\}} d_{\sigma} \tag{4.1}$$

with d_{σ} in Eq. (3.1) is solved for a given index set

$$0 \leq n_1 < n_2 < \ldots < n_M < N,$$

where $N = 2^j$ and $M^2 \leq N$. Our considerations will lead to an algorithm providing a suboptimal $\tilde{\sigma}$ ensuring a large distance $d_{\tilde{\sigma}}$. If two or more found values $\tilde{\sigma}$ satisfy the distance criterion with the same distance $d_{\tilde{\sigma}}$, then we will choose from the set of these parameters the one which minimizes the value

$$\left| \sum_{k=1}^{M} \omega_N^{\tilde{\sigma} n_k} \right|$$

thereby enforcing a better distribution of the knots $\omega_N^{\tilde{\sigma} n_k}$ on the unit circle.

As before, let $d_{\ell,k} := |n_\ell - n_k|$ for $\ell, k = 1, \ldots M$, $\ell > k$, and $\tilde{d}_{\ell,k} = N - d_{\ell,k} = -d_{\ell,k} \bmod N$ be the periodic distances modulo N. The set D contains all distinct

values $d_{\ell,k}$ and $\tilde{d}_{\ell,k}$. Clearly, D has at most $M(M-1)$ elements and can be also smaller since distances $d_{\ell,k}$ and $d_{\ell',k'}$ can coincide. Further, $0 \notin D$ since $d_{\ell,k} \neq 0$.

We use σD to denote the set that contains all the distances $d \in D$ multiplied by σ modulo N. For our problem, we look at $\sigma D \pmod N$. Our goal is to seek $\sigma \in \Sigma = \{1, 2, \ldots, N-1\}$ such that

> either the minimum value of σD ($\min \sigma D$) is enlarged to a chosen value;
> or the minimum value of σD ($\min \sigma D$) is maximized.

Our main idea is now to efficiently determine subsets $\Sigma^{(L)}$, $L = 0, 1, 2, \ldots$, such that $\min \sigma D > L$ for all $\Sigma^{(L)}$, i.e., $\sigma d > L$ for all $d \in D$ and all $\sigma \in \Sigma^{(L)} \subset \Sigma$.

For this purpose, we consider the following disjoint subsets $D^{(\ell)}$ of D. Recalling that $N = 2^j$ for some $j > 0$, each $d \in D$ can be uniquely written in the form $d = 2^\ell \bar{d}$ with $\bar{d}$ odd and $\ell \in \{0, 1, \ldots, 2^{j-1}\}$. We write

$$D^{(\ell)} := \{d \in D : d = 2^\ell \bar{d},\ \bar{d} \text{ odd}\}, \qquad \ell = 0, \ldots, j-1, \tag{4.2}$$

such that $D = \cup_{\ell=0}^{j-1} D^{(\ell)}$. Obviously, each $d \in D^{(\ell)}$ possesses $2^\ell - 1$ nonzero divisors modulo 2^j, namely $2^{j-\ell} r$, $r = 1, \ldots, 2^\ell - 1$.

Construction of $\Sigma^{(0)}$ To obtain $\Sigma^{(0)}$, we remove all σ from Σ satisfying $\sigma d = 0 \bmod N$ for some d in D. This is done by fixing the largest index $\ell \leq j-1$ with $D^{(\ell)} \neq \emptyset$ and removing all multiples of $2^{j-\ell}$ from Σ. For example, if D contains the distance $d = N/2 = 2^{j-1}$, then we have to remove all even integers from Σ in order to obtain $\Sigma^{(0)}$. By definition, all remaining values $\sigma \in \Sigma^{(0)}$ ensure that the σn_k are pairwise distinct such that the corresponding Vandermonde matrix V_M is invertible. We simply observe that all odd integers $\sigma \in \{0, \ldots, N-1\}$ are still in $\Sigma^{(0)}$.

Construction of $\Sigma^{(1)}$ To obtain $\Sigma^{(1)}$, we have to remove all $\sigma \in \Sigma^{(0)}$ satisfying $\sigma d = 1 \bmod N$ for some $d \in D$. By construction, we only need to consider $d \in D^{(0)}$ here, since the d's in $D^{(\ell)}$ with $\ell > 0$ are even. Thus, we have to remove the inverse of each $d \in D^{(0)}$ (modulo N) from $\Sigma^{(0)}$ to obtain $\Sigma^{(1)}$.

Construction of $\Sigma^{(L)}$ We can proceed with this idea to obtain the sets $\Sigma^{(L)}$, $L > 1$, by increasing the lower bound of σD. For that purpose, we define the following subsets of $\Sigma^{(0)}$,

$$\begin{aligned}
T^{(0)} &= \{\sigma \in \Sigma^{(0)} : \sigma \cdot d = 1 \bmod N \text{ for a } d \in D^{(0)}\},\\
T^{(1)} &= \{\sigma \in \Sigma^{(0)} : 2^{-1}\sigma \cdot d = 1 \bmod N/2 \text{ for a } d \in D^{(1)}\},\\
&\vdots \qquad\qquad\qquad \vdots\\
T^{(j-1)} &= \{\sigma \in \Sigma^{(0)} : 2^{-j+1}\sigma \cdot d = 1 \bmod N/2^{j-1} \text{ for a } d \in D^{(j-1)}\}.
\end{aligned}$$

We use the convention that $kT^{(\ell)} := \{k\sigma \bmod N : \sigma \in T^{(\ell)}\}$ for $k \in \{1, \ldots, N-1\}$. Then, as already described before, we obtain

$$\Sigma^{(1)} = \Sigma^{(0)} - T^{(0)}$$

enforcing that the minimal distance $\min \sigma D$ is at least 2.

In order to obtain $\Sigma^{(2)}$, we have to remove $2T^{(0)}$ and $T^{(1)}$ from $\Sigma^{(1)}$, since these sets contain parameters σ satisfying $\sigma d = 2 \bmod N$ for some $d \in D^{(0)}$ resp. $D^{(1)}$. Observe that the distances in $D^{(\ell)}$ with $\ell > 1$ need not to be checked since they contain the factor 4 and can never produce a remainder 2. Thus, $\Sigma^{(2)} = \Sigma^{(1)} - 2T^{(0)} - T^{(1)}$.

Generally, for $L \in \{1, \ldots, \lfloor N/M \rfloor\}$ with $L = 2^r \bar{L}$ we get in a similar manner

$$\begin{aligned} \Sigma^{(L)} &= \Sigma^{(L-1)} - L\,T^{(0)} - \frac{L}{2}\,T^{(1)} - \ldots - \frac{L}{2^r}\,T^{(r)} \\ &= \Sigma^{(0)} - \bigcup_{k=1}^{L} kT^{(0)} - \bigcup_{k=1}^{\lfloor L/2 \rfloor} kT^{(1)} - \ldots - \bigcup_{k=1}^{\lfloor L/2^{j-1} \rfloor} kT^{(j-1)}, \end{aligned} \tag{4.3}$$

where the sets vanish if $\lfloor L/2^s \rfloor = 0$ for $s \in \{0, \ldots, j-1\}$.

Example 4.1 We reconsider the example of Section 2. Let $\mathbf{x} \in \mathbb{C}^{64}$ be 5-sparse with $x_k = 1$ for $k \in I^{(6)} := \{1, 5, 6, 13, 59\}$. Assume that we have already computed $\mathbf{x}^{(4)} \in \mathbb{C}^{16}$ with $I^{(4)} = \{1, 5, 6, 11, 13\}$, i.e., $N = 16$, $M = 5$ (disregarding that $M^2 > N$ in this small example case). We find the set of distances $D = D^{(0)} \cup D^{(1)} \cup D^{(2)} \cup D^{(3)}$ with

$$D^{(0)} = \{1, 5, 7, 9, 11, 15\}, \quad D^{(1)} = \{2, 6, 10, 14\}, \quad D^{(2)} = \{4, 12\}, \quad D^{(3)} = \{8\}.$$

Since $8 \in D$ all even values σ have to be removed from $\Sigma = \{1, 2, \ldots, 15\}$ and we obtain $\Sigma^{(0)} = \{1, 3, 5, 7, 9, 11, 13, 15\}$. Now, we compute

$$T^{(0)} = \{1, 3, 7, 9, 13, 15\}, \quad T^{(1)} = T^{(2)} = T^{(3)} = \{1, 3, 5, 7, 9, 11, 13, 15\}.$$

Thus, $\Sigma^{(1)} = \Sigma^{(0)} - T^{(0)} = \{5, 11\}$ and these are the only parameters ensuring the distance $d_\sigma \geq 2$. Indeed, $5I^{(4)} = \{5, 7, 9, 1, 14\}$ with $d_5 = 2$. The corresponding Vandermonde matrix $V_5(5)$ possesses the condition 3.02. Observe that $\sigma = \sigma_4 = 11 = 16 - 5$ gives the same result by "reflection".

In step $j = 5$, having computed $\mathbf{x}^{(5)} \in \mathbb{C}^{32}$ with $I^{(5)} = \{1, 5, 6, 13, 27\}$, i.e., $N = 32$, $M = 5$, we find

$$\begin{aligned} &D^{(0)} = \{1, 5, 7, 11, 21, 25, 27, 31\}, \; D^{(1)} = \{6, 10, 14, 18, 22, 26\}, \\ &D^{(2)} = \{4, 12, 20, 28\}, \qquad D^{(3)} = \{8, 24\}, \qquad D^{(4)} = \emptyset. \end{aligned}$$

To obtain $\Sigma^{(0)}$, we have to remove thus all multiples of 4 and get

$$\Sigma^{(0)} = \{1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 18, 19, 21, 22, 23, 25, 26, 27, 29, 30, 31\}.$$

Further, we find

$$\begin{aligned} &T^{(0)} = \{1, 3, 9, 13, 19, 23, 29, 31\}, \quad T^{(1)} = \{3, 5, 7, 9, 11, 13, 19, 21, 23, 25, 27, 29\}, \\ &T^{(2)} = T^{(3)} = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31\}. \end{aligned}$$

Thus,

$$\Sigma^{(1)} = \Sigma^{(0)} - T^{(0)} = \{2, 5, 6, 7, 10, 11, 14, 15, 17, 18, 21, 22, 25, 26, 27, 30\},$$
$$\Sigma^{(2)} = \Sigma^{(1)} - 2T^{(0)} - T^{(1)} = \{10, 15, 17, 22\},$$
$$\Sigma^{(3)} = \Sigma^{(2)} - 3T^{(0)} = \{10, 15, 17, 22\},$$
$$\Sigma^{(4)} = \Sigma^{(3)} - 4T^{(0)} - 2T^{(1)} - T^{(2)} = \emptyset.$$

Thus, $\sigma_5 = 10$ and $\sigma_5 = 15$ are the optimal parameters in this case achieving both a distance $d_\sigma = 4$ between neighboring knots, while 17 and 22 are the corresponding "reflections".

Application to the iterative procedure In order to apply the above ideas for constructing an optimal σ in the sparse FFT Algorithm 2.3, we simplify the procedure. We distinguish the following two cases: either the number M_j of nonzero values in $\mathbf{x}^{(j)}$ is the same as in the previous step, i.e., $M_j = M_{j-1}$, or it increases, i.e., $M_j > M_{j-1}$. For the first case, we will show in the next theorem that, supposed that a suitable parameter σ_{j-1} has been found already in step $j-1$, then $\sigma_j = 2\sigma_{j-1}$ will be a suitable parameter in step j and moreover, the obtained Vandermonde matrices will coincide. In the second case, we can simplify the above procedure since in this case $D^{(j-1)} = \{2^{j-1}\}$ will appear.

Theorem 4.2 *Let σ_{j-1} be the parameter that has been used in the iterative procedure in order to obtain a well-conditioned Vandermonde matrix $V_{M_{j-1}} = \left(\omega_{2^{j-1}}^{\sigma(j-1)n_r^{(j-1)}}\right)_{p,r=1}^{M_{j-1}}$ in step $j-1$ of Algorithm 2.3 to compute $\mathbf{x}^{(j)}$, where $0 < n_1^{(j-1)} < \ldots < n_{M_{j-1}}^{(j-1)} < 2^{j-1}$ denotes the support of $\mathbf{x}^{(j-1)}$. Then, if $M_j = M_{j-1}$, the parameter $\sigma_j = 2\sigma_{j-1}$ produces the same Vandermonde matrix in the iteration step j, i.e.,*

$$V_{M_j} = \left(\omega_{2^j}^{2\sigma_{j-1}(p-1)n_r^{(j)}}\right)_{p,r=1}^{M_j} = V_{M_{j-1}}.$$

Proof Since $M_j = M_{j-1}$, each support index $n_r^{(j)}$ of $\mathbf{x}^{(j)}$ is related to $n_r^{(j-1)}$ by

$$n_r^{(j)} \in \{n_r^{(j-1)},\ n_r^{(j-1)} + 2^{j-1}\}.$$

Thus,

$$2\sigma_{j-1}(p-1)n_r^{(j)} \bmod 2^j = 2\sigma_{j-1}(p-1)n_r^{(j-1)} \bmod 2^j$$

and hence,

$$\omega_{2^j}^{2\sigma_{j-1}(p-1)n_r^{(j)}} = \omega_{2^j}^{2\sigma_{j-1}(p-1)n_r^{(j-1)}} = \omega_{2^{j-1}}^{\sigma_{j-1}(p-1)n_r^{(j)}},$$

i.e., the entries of V_{M_j} and $V_{M_{j-1}}$ coincide. □

Reconsidering Example 4.1, for $j = 4$ and $j = 5$ we obtained the optimal parameters $\sigma_4 = 5$ and $\sigma_5 = 10$, respectively. In this case, we had $M_4 = M_5 = 5$. Thus, the choice $\sigma_j = 2\sigma_{j-1}$ can be even optimal regarding the optimization problem (4.1).

Let us now consider the second case $M_j > M_{j-1}$. This case can only occur if (at least) one support index $n_k^{(j-1)}$ splits into two new support indices $n_k^{(j)} = n_k^{(j-1)}$ and $n_{k+s}^{(j)} = n_k^{(j-1)} + 2^{j-1}$. Thus, D contains the distance 2^{j-1} and therefore $D^{(j-1)} \neq \emptyset$ which means that the set $\Sigma^{(0)}$ contains only the odd integers in the range $\{1, \ldots, N-1\}$ with $N = 2^j$. Hence, all sets of the form $2kT^{(r)}$, $k \in \mathbb{N}$, $r \in \{0, \ldots j-1\}$ are disjoint from $\Sigma^{(0)}$ and the evaluation of $\Sigma^{(L)}$ with $L = 2^r \bar{L}$ in (4.3) simplifies to

$$\begin{aligned}\Sigma^{(L)} &= \Sigma^{(L-1)} - \bar{L}T^{(r)} \\ &= \Sigma^{(0)} - \bigcup_{k=1}^{\lfloor L/2 \rfloor} (2k-1)T^{(0)} - \bigcup_{k=1}^{\lfloor L/4 \rfloor} (2k-1)T^{(1)} - \ldots - \bigcup_{k=1}^{\lfloor L/2^{j-1} \rfloor} (2k-1)T^{(j-2)}.\end{aligned}$$

Applying the above formula, we can iteratively determine the optimal parameter σ by computing the sets $\Sigma^{(0)}, \Sigma^{(1)}, \ldots$ and choosing $\sigma \in \Sigma^{(L')}$ such that $\Sigma^{(L)} = \emptyset$ for all $L > L'$. But as this means that we have to consider all odd integers in $\{1, \ldots, N/2 - 1\}$, the computation of σ in this way is very expensive.

Therefore, we propose two different methods for a more efficient computation of σ that we describe in the following. The first approach is based on the idea that we can restrict our search to parameters σ which give "sufficiently good" distances. In the second approach, we restrict the number of regarded σ's in advance in order to reduce the computational effort.

First method By Theorem 3.2, we already have $\frac{N}{M^2} \leq d \leq \frac{N}{M}$. Let us assume that there exist odd distances in D, i.e., $D^{(0)} \neq \emptyset$. Now, we fix the largest odd integer being smaller than N/M,

$$\tilde{d} = 2\left\lfloor \frac{N}{2M} + \frac{1}{2} \right\rfloor - 1 < \frac{N}{M},$$

since this would be the optimal distance "$d_{\tilde{\sigma}}$" that we can hope for. We compute all parameters σ satisfying $\sigma d = \tilde{d} \bmod N$ for at least one distance $d \in D$. Since $\tilde{d}$ is odd, we can restrict our search to the elements $d \in D^{(0)}$. For each distance $d \in D^{(0)}$ with $d < N/2$ we apply the following procedure: We compute σ satisfying $\sigma d = \tilde{d} \bmod N$, i.e., $\sigma = \tilde{d} d^{-1} \bmod N$, where d^{-1} has been already computed in $T^{(0)}$. For the obtained σ's, we compute $\sigma I^{(j)}$, order the values of this set by size and determine the minimal distance d_σ between neighboring values. In this computation, we can neglect the parameters $\sigma \in T^{(0)}$ if there is at least one parameter $\sigma \notin T^{(0)}$, see also Example 4.7.

Inspecting all distances d_σ found in this way, we choose the parameter σ that produces the largest minimal distance. If there is more than one σ achieving this largest distance, we choose the σ for which the sum $\left|\sum_{k=1}^{M} \omega_N^{\sigma n_k}\right|$ is minimal. We only have to consider distances in $D^{(0)}$ with $d < N/2$, as the remaining distances $N - d$ give "reflected" sets $(n - \sigma)I^{(j)}$ with the same minimal distances. Thus, the number of relevant distances in $D^{(0)}$ is bounded by $M^2/4$.

Let us first give an example illustrating the computation of σ as above. Afterwards, we extend the procedure for the case when $D^{(0)} = \emptyset$ and summarize the algorithm for the computation of σ.

Example 4.3 Let us assume, we are given $\mathbf{x}^{(7)} \in \mathbb{C}^{128}$ with the set of nonzero indices $I^{(7)} = \{0, 5, 6, 64\}$. In this case, the sparsity changes in the last iteration step, and we have $M_7 = 4 > M_6$. We obtain the sets $D = D^{(0)} \cup D^{(1)} \cup D^{(6)}$ with

$$D^{(0)} = \{1, 5, 59, 69, 123, 127\}, \quad D^{(1)} = \{6, 58, 70, 122\}, \quad D^{(6)} = \{64\}.$$

Further, it follows that

$$T^{(0)} = \{1, 77, 115, 13, 51, 127\}, \quad T^{(1)} = \{43, 107, 53, 117, 11, 75, 21, 85\},$$

where the order of the entries in $T^{(0)}$ and $T^{(1)}$ corresponds to that in $D^{(0)}$ resp. $D^{(1)}$, i.e., we have, e.g., $5^{-1} \bmod 128 = 77$. We choose $\tilde{d} = 2\left\lfloor \frac{1}{2}\left\lceil \frac{N}{M} \right\rceil \right\rfloor - 1 = 31$ as optimal distance value. Considering the elements in $D^{(0)}$, we obtain the following cases:

a) $d_1 = 1$: From $1\sigma = 31 \bmod 128$, we get $\sigma := 31$ and $31I^{(7)} = \{0, 27, 58, 64\}$. Thus, this set gives only a minimal distance 6.
b) $d_2 = 5$: From $5\sigma = 31 \bmod 128$, we get $\sigma = 77 \cdot 31 \bmod 128 = 83 \bmod 128$ and $83I^{(7)} = \{0, 31, 114, 64\}$. Hence, $\sigma = 83$ leads to a minimal distance 14.
c) $d_3 = 59$: From $59\sigma = 31 \bmod 128$, we get $\sigma = 115 \cdot 31 \bmod 128 = 109 \bmod 128$ and $109I^{(7)} = \{0, 33, 14, 64\}$. Thus, $\sigma = 109$ leads to the distance $d = 14$ as well.

Comparing $\sigma = 83$ and $\sigma = 109$, we obtain in the first case $|\omega_{128}^{0} + \omega_{128}^{31} + \omega_{128}^{114} + \omega_{128}^{64}| = 0.8992$ while in the second case $|\omega_{128}^{0} + \omega_{128}^{33} + \omega_{128}^{14} + \omega_{128}^{64}| = 1.7864$. Therefore, we prefer $\sigma = 83$. The corresponding (4×4)-Vandermonde matrix possesses the condition 3.2199 for $\sigma = 83$.

The optimal parameter in this small example is $\sigma = 59$ with the corresponding index set $59I^{(7)} = \{0, 39, 98, 64\}$. This yields the optimal distance $d_\sigma = 25$ for which the Vandermonde matrix achieves the condition 1.4535.

If $D^{(0)} = \emptyset$, then we consider $D^{(1)}$. As mentioned before, all distances $d \in D^{(1)}$ are of the form $d = 2 \cdot \bar{d}$ for some odd $\bar{d} \in \{1, 3, \ldots, N/2 - 1\}$. Therefore, we choose

$$\tilde{d} = 4\left\lfloor \frac{N}{4M} + \frac{1}{2} \right\rfloor - 2,$$

being the largest even integer smaller than $\frac{N}{M}$ that is divisible by 2 but not by 4. Then, we compute for each distance $d \in D^{(1)}$ the parameter σ which achieves $\tilde{d}$, i.e.,

$$\sigma = \frac{\tilde{d}}{2}\left(\frac{d}{2}\right)^{-1} \bmod \frac{N}{2}.$$

As above, from the obtained set of σ's we choose the σ for which the minimal distance between neighboring values in the ordered set $\sigma I^{(j)}$ is maximal and, if there are several possibilities, the one for which the sum $\left|\sum_{k=1}^{M} \omega_N^{\sigma n_k}\right|$ is minimal.

Here again, we can safe computation time, since if not all found σ's are in $T^{(1)}$, we can restrict the computation of these ordered sets $\sigma I^{(j)}$ to $\sigma \notin T^{(1)}$.

If $D^{(0)} = D^{(1)} = \emptyset$, but $D^{(2)} \neq \emptyset$, we proceed similarly choosing the optimal distance

$$\tilde{d} = 8 \left\lfloor \frac{N}{8M} + \frac{1}{2} \right\rfloor - 4$$

and computing the parameters σ by

$$\sigma = \frac{\tilde{d}}{4} \left(\frac{d}{4} \right)^{-1} \bmod \frac{N}{4}$$

for all $d \in D^{(2)}$, etc.

We summarize our first method to find a parameter σ where we try to achieve an optimal distance.

Algorithm 4.4 ***(Computation of σ if $M_j > M_{j-1}$, choosing "optimal" $\tilde{d}$)***

Input: *$N = 2^j$*
Index set $I^{(j)}$ containing M indices $0 \leq n_1 < n_2 < \cdots < n_M < N$.
Output: *$\sigma = \sigma_j$ such that σD contains at least once the "optimal distance" $\tilde{d}$.*

1. *Compute the pairwise distances between all $n_1, \ldots, n_M$ and form D. Form the subsets $D^{(0)}, \ldots, D^{(j-1)}$ according to* Eq. (4.2)*. Fix the smallest integer ℓ such that $D^{(\ell)} \neq \emptyset$. Compute the set $T^{(\ell)}$.*
2. *Compute $\tilde{d} := 2^{\ell+1} \left\lfloor \frac{N}{2^{\ell+1}M} + \frac{1}{2} \right\rfloor - 2^{\ell}$ as "optimal" distance value.*
3. *Set $\Sigma := \emptyset$.*
   For all elements $d \in D^{(\ell)}$ with $d < N/2$ do
   - *Compute*
   $$\sigma = \frac{\tilde{d}}{2^{\ell}} \left(\frac{d}{2^{\ell}} \right)^{-1} \bmod N/2^{\ell}.$$
   - *Form $\Sigma = \Sigma \cup \{\sigma\}$.*
4. *If $\Sigma - T^{(\ell)} \neq \emptyset$ then set $\Sigma = \Sigma - T^{(\ell)}$.*
5. *For all $\sigma \in \Sigma$ do*
   - *Form $\sigma I^{(j)}$. Order the elements of $\sigma I^{(j)}$ by size and compute the smallest distance L' between neighboring values in $\sigma I^{(j)}$.*
6. *Choose the $\sigma \in \Sigma$ that leads to the largest minimal distance L' of neighboring values. If there are several parameters σ achieving the same distance, choose the σ which minimizes the sum $\left| \sum_{k=1}^{M} \omega_N^{\sigma n_k} \right|$.*

The computational effort to find σ using this algorithm is at most $\mathcal{O}(M^2)$. The set D contains at most $M(M-1)$ entries. Using a distance matrix, where the (ℓ, k)th entry is $n_\ell - n_k \bmod N$, we can exploit the computations from the previous iteration steps and only have to change (or add) single rows and corresponding columns if one

entry $n_k^{(j-1)}$ shifts to $n_k^{(j)} = n_k^{(j-1)} + N/2$ or if $M_j > M_{j-1}$ and entries $n_k^{(j-1)}$ split into $n_k^{(j)} = n_k^{(j-1)}$ and $n_{k+1}^{(j)} = n_k^{(j-1)} + N/2$. The computation of $T^{(\ell)}$ also iteratively uses the values found in the previous step, where we exploit that for $n_k \cdot n_k' = 1 \bmod N$ it follows that either $n_k \cdot n_k' = 1 \bmod 2N$ or $n_k \cdot (n_k' + N) = 1 \bmod 2N$.

Second method We present a second idea for determining a suitable parameter σ. This time we limit the set of possible parameters σ in advance. As already seen, only odd parameters σ have to be considered when $M_j > M_{j-1}$. Additionally, our numerical experiments indicate that prime numbers might be a good choice. Hence, the second idea for choosing a suitable σ is to restrict the search to prime numbers. In order to limit the computational effort, we propose here to choose σ among the M larges primes being smaller than $N/2$ which has achieved good results in practice.

We summarize the second algorithm using prime numbers.

Algorithm 4.5 ***(Computation of σ if $M_j > M_{j-1}$, using prime numbers)***

Input: *$N = 2^j$, set Σ of M largest prime numbers smaller than $N/2$, Index set $I^{(j)}$ containing M indices $0 \le n_1 < n_2 < \cdots < n_M < N$.*
Output: *$\sigma = \sigma_j$ prime.*

1. *For all $\sigma \in \Sigma$ do*
   (a) *Compute the set $\sigma I^{(j)}$.*
   (b) *Order the elements of $\sigma I^{(j)}$ by size and compute the smallest distance L' between neighboring values.*
2. *Choose the $\sigma \in \Sigma$ that leads to the largest minimal distance L' of neighboring values. If there are several parameters σ achieving the same distance, choose the σ which minimizes the sum $\left|\sum_{k=1}^{M} \omega_N^{\sigma n_k}\right|$.*

Remark 4.6 If the largest maximal distance L' is too small compared to the optimal distance N/M, we recommend to extend the matrix $V_{M,M}$ to $V_{2M,M}$ by taking just more rows. Keep in mind that taking all N rows would even lead to the optimal condition 1. Observe that the QR decomposition of this rectangular Vandermonde matrix has still a complexity of $\mathcal{O}(M^2)$, see [4].

We finish the section by presenting a larger example.

Example 4.7 Let $N = 2^{14} = 16384$, $M = 17$ and let

$$I = I^{(14)} := \{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 345, 1234, 1235\}$$

be the set of nonzero indices, such that $x_k = 1$ for $k \in I^{(14)}$ and $x_k = 0$ for $k \notin I^{(14)}$. Assume that $\widehat{\mathbf{x}} \in \mathbb{C}^{16384}$ is given.

We apply Algorithm 2.3 to recover $\mathbf{x}$. We summarize our findings at each level in Table 1.

Table 1 Sparsity M_j and obtained index sets at each level for Example 4.7

j	2^j	M_j	Method	Obtained new index set $I^{(j+1)}$
0	1	1	FFT(1)	{0, 1}
1	2	2	FFT(2)	{0, 1, 2, 3}
2	4	4	FFT(4)	{0, 1, 2, 3, 4, 5, 6, 7}
3	8	8	FFT(8)	{0, 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 15}
4	16	13	FFT(16)	{6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 24, 25, 26}
5	32	16	FFT(32)	{6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 25, 56, 57, 58}
6	64	17	FFT(64)	{6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 25, 56, 57, 58}
7	128	17	FFT(128)	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 82, 83, 89}
8	256	17	FFT(256)	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 210, 211, 345}
9	512	17	$\sigma = 88$	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 210, 211, 345}
10	1024	17	$\sigma = 176$	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 345, 1234, 1235}
11	2048	17	$\sigma = 352$	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 345, 1234, 1235}
12	4096	17	$\sigma = 704$	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 345, 1234, 1235}
13	8192	17	$\sigma = 1408$	{6, 7, 8, 9, 10, 11, 12, 13, 56, 57, 58, 79, 80, 81, 345, 1234, 1235}

For $j = 0, \ldots, 8$, the FFT is applied since $2^j < M_j^2$. We obtain $\mathbf{x}^{(9)} \in \mathbb{C}^{512}$ with sparsity $M_9 = 17$. Since $M_9^2 = 289 < 512$, we apply for $j = 9$ the special reconstruction step using the Vandermonde system. Since the number of nonzero entries has not changed since step $j = 6$, we may go back there to find σ. In that case, $M = 17$, and $N = 2^6 = 64$, we cannot achieve a distance better than $\lfloor N/M \rfloor = 3$. By previous observations, each odd σ already provides a distance 1. We find

$$D^{(0)} = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 31, 33, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63\}.$$

Hence, $D^{(0)}$ contains all odd numbers up to 29 and 35. Therefore, only σ with $29\sigma = 1 \bmod 64$ or with $35\sigma = 1 \bmod 64$ need to be considered since all other odd σ's are in $T^{(0)}$ and therefore cannot be in $\Sigma^{(1)}$. We obtain $\Sigma^{(1)} = \{11, 53\}$. With $\sigma = 11$ we find

$$11I^{(6)} = \{2, 4, 6, 13, 15, 17, 19, 24, 35, 37, 40, 46, 48, 51, 57, 59, 62\}.$$

such that $d = 2$. The second parameter $\sigma = 53 = 64 - 11$ provides a "reflected" set modulo 64. Algorithm 4.4 gives us $\sigma = 53$ here, and Algorithm 4.5 gives the result $\sigma = 11$, since we have to check up to 17 primes smaller than 32 including 11. Having found $\sigma_6 = 11$, we can choose $\sigma_9 = 8\sigma_6 = 88$ by Theorem 4.2. We obtain in this case the condition 97.37 of the corresponding Vandermonde matrix, compared to $6.37 \cdot 10^{15}$ for taking just $\sigma = 1$. Solving the equation system yields $\mathbf{x}^{(10)}$.

Since the number M of nonzero entries is already achieved, we can just take $\sigma_{10} = 2\sigma_9 = 176$, $\sigma_{11} = 2\sigma_{10} = 352$, etc. The (17×17)-Vandermonde matrix

applied to reconstruct $\mathbf{x}^{(j)}$ for $j \geq 10$ exactly coincides with the Vandermonde matrix for $j = 9$. Therefore, we can use the QR decomposition that has already been computed in the previous iteration step, thereby further reducing the numerical effort. Summarizing the arithmetical complexity in this case, we observe that we need $\mathcal{O}(2^9 \log 2^9)$ operations to compute $\mathbf{x}^{(9)}$ and afterwards one QR decomposition of the Vandermonde matrix with effort $\mathcal{O}(M^2)$. Finding σ and solving the Vandermonde system for the last four steps then only requires a matrix multiplication and a back substitution with effort $\mathcal{O}(M^2)$. Since $2^9 < 2M^2$, the complete effort is here $\mathcal{O}(M^2 \log \frac{N}{M^2})$.

5 Numerical experiments

First, we present some numerical experiments showing that the two proposed Algorithms 4.4 and 4.5 work well in practice. For different values of N and sparsity M, we consider randomly chosen sets of M indices, compute the minimal distance in the set of distances D as well as the minimal distances in σD for σ achieved in Algorithm 4.4 resp. 4.5. For each fixed pair (N, M), we have computed these distances for 100 randomly chosen index sets of size M in $\{0, \ldots, N-1\}$. The obtained average distances are presented in Table 2.

The findings indicate that the proposed algorithms in any case strongly improve the minimal distance and hence also the matrix condition when they are applied in our reconstruction algorithm. Algorithm 4.4 which aims at achieving a least once an "optimal" distance $\tilde{d}$ performs slightly better than Algorithm 4.5. In return, Algorithm 4.5, which only considers M different primes for σ, provides a very simple and efficient possibility to determine a suitable σ.

Finally, we consider the arithmetical complexity of our new Algorithm 2.3 in more detail and compare the runtime of the algorithm with the FFT in Matlab. In Algorithm 2.3, we compute $\mathbf{x}^{(j+1)}$ from $\mathbf{x}^{(j)}$ for $j = 0, \ldots, J-1$. As long as the found sparsity $M_j \leq M$ satisfies $M_j^2 > 2^j$, we employ the FFT of length 2^j to compute $\mathbf{x}^{(j+1)}$. Thus, for the iteration steps $j = 0, \ldots, L$ with $L = 2\lfloor \log_2 M \rfloor$, we need at each step

Table 2 Average of largest minimal distances achieved by Algorithm 4.4 resp. 4.5 for different values of N and M and 100 randomly chosen index sets

N	M	Minimal distance in D	Minimal distance in σD, σ from Algorithm 4.4	Minimal distance in σD, σ from Algorithm 4.5
$2^7 = 128$	4	8.65	16.66	16.61
$2^{10} = 1024$	4	73.54	138.88	139.09
$2^{10} = 1024$	20	2.96	12.09	9.16
$2^{15} = 32\,768$	4	2124.21	4389.99	4181.90
$2^{15} = 32\,768$	20	72.34	372.96	285.23
$2^{15} = 32\,768$	50	14.93	88.42	57.59

one DFT(2^j) and 2^{j+1} further operations. Using the Sande-Tukey algorithm with $\frac{3}{2}N \log_2 N$ operations for a DFT(N), we require

$$\sum_{j=0}^{L} \left(\frac{3}{2} j 2^j + 2^{j+1} \right) = 3(2^L L + 1) + 2^{L+2} = 2^L(3L+4) + 3 \le M^2(6 \log_2 M + 4) + 3$$

operations, where we disregard the needed comparisons. In particular, for $M^2 > N$, the arithmetical complexity of the algorithm is still comparable with the usual FFT algorithm. For the remaining iteration steps, $j = L+1, \ldots, J-1$, we have to build the partial Fourier matrix $\mathbf{A}^{(j+1)}$. We assume that the powers of ω_N are predefined as for the usual FFT. The evaluation of σ using Algorithm 4.5 requires $2M^2 + \mathcal{O}(M)$ operations and M sortings of vectors of length M. We keep in mind that σ needs not to be computed, if the sparsity does not change any longer during the iteration. The QR factorization of Demeure [4] possesses computational costs of $8.5M^2 + \mathcal{O}(M)$ such that the equation system in (2.8) can be solved using $12.5M^2 + \mathcal{O}(M)$ operations, where the computation of σ is included for each second step. Summing up, we obtain

$$12.5M^2((\log_2 N) - 1 - L) + \mathcal{O}(M) \le 12.5M^2((\log_2 N) - 2(\log_2 M)) + \mathcal{O}(M)$$

operations for the remaining iteration steps, disregarding comparisons and sorting. Together, we have a complexity of $M^2(12.5 \log_2 N - 19 \log_2 M + 4) + \mathcal{O}(M)$ for $M^2 < N$ and $N(3 \log_2 N + 4) + \mathcal{O}(N)$ for $M^2 \ge N$ disregarding comparisons and sorting. The following Table 3 gives a comparison of these complexities for sparse signals with $M = 5,\ 10,\ 20$.

A primitive Matlab implementation of all algorithms in this paper can be found at http://na.math.uni-goettingen.de/index.php?section=gruppe&subsection=software. Our current implementation is less efficient than the arithmetical complexity suggests

Table 3 Complexity of the FFT of length $N = 2^j$ in comparison to Algorithm 2.3 with sparsity $M = 5,\ 10,\ 20$, where sorting and comparison is not taken into account

J	DFT(2^J)	$M = 5$	$M = 10$	$M = 20$
10	15360	2122	6588	18753
11	33792	2435	7838	23753
12	73728	2747	9088	28753
13	1590744	3060	10338	33753
14	344064	3372	11588	38753
15	737280	3685	12838	43753
16	1572864	3997	14088	48753
17	3342336	4310	15338	53753
18	7077888	4622	16588	58753
19	14942208	4935	17838	63753
20	31457280	5247	19088	68753

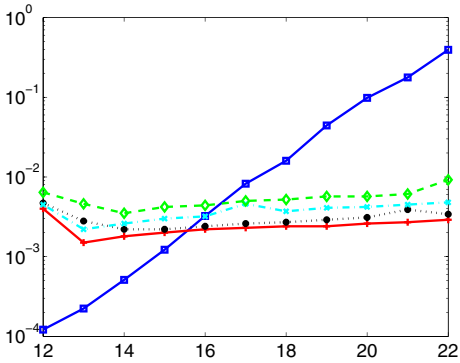

Fig. 1 Runtime comparison (in seconds) of the FFT (*blue line*) and our algorithm with $M = 5$ (*red line*), $M = 10$ (*black dotted line*), $M = 20$ (*cyan dash-dots line*) and $M = 30$ (*green dashed line*) for length $N = 2^j$ with $j = 12, \ldots, 22$

but can still improve the Matlab FFT algorithm for strong sparsity. In Fig. 1, we compare the runtime of the FFT of length 2^j with our algorithm, with different sparsities M. Here, we have used a slight modification of Algorithm 2.3, where M is known in advance such that the first $L = 2\lfloor \log_2 M \rfloor$ steps of Algorithm 2.3 can be replaced by one FFT of length 2^{L+1}. The runtimes in Fig. 1 have been obtained by computing the average runtime for 10 tests with randomly chosen sparse vectors with sparsities $M = 5,\ 10,\ 20,\ 30$.

Let us finish this section with some final remarks. There are several issues still open according to the considered approach that particularly regard a more efficient implementation and a further improvement of the sparse FFT algorithm. As the numerical experiments show, the proposed parameter selection provides sufficiently small condition numbers of the Vandermonde matrix in most of the cases. However, particularly for larger M, the condition number gets too large, such that the algorithm cannot process noisy data in a suitable way. Fortunately, the Algorithms 4.4 and 4.5 for parameter selection give us an essential hint about the condition number of the Vandermonde matrix in the next iteration step, since the largest minimal distance L' of neighboring knots is computed. This knowledge could be used to decide to apply rectangular Vandermonde matrices with more rows at single levels (and more corresponding Fourier values).

Possible improvements in runtime of the algorithm regard for example the case when the sparsity does not change from one level to the next. Then, we can use the same Vandermonde matrix as at the previous level and just directly take the corresponding QR decomposition again for solving the new equation system.

Acknowledgments The authors thank the anonymous referee for valuable suggestions to improve this manuscript. This work is supported by the Deutsche Forschungsgemeinschaft (DFG) in the project PL 170/16-1 and in the framework of the RTG 2088.

References

1. Akavia, A.: Deterministic sparse fourier approximation via approximating arithmetic progressions. IEEE Trans. Inform. Theory **60**(3), 1733–1741 (2014)

2. Bittens, S.: Sparse FFT for functions with short frequency support. Dolomites Res. Notes Approx. **10**, 43–55 (2017)
3. Berman, L., Feuer, A.: On perfect conditioning of Vandermonde matrices on the unit circle. Electron. J. Linear Algebra **16**, 157–161 (2007)
4. Demeure, C.J.: Fast QR factorization of Vandermonde matrices. Linear Algebra Appl. **122–124**, 165–194 (1989)
5. Giesbrecht, M., Labahn, G., Lee, W.-S.: Symbolic-numeric sparse interpolation of multivariate polynomials. J. Symbolic Comput. **44**(8), 943–959 (2009)
6. Giesbrecht, M., Roche, D.S.: Diversification improves interpolation. In: Leykin, A. (ed.) Proceedings of the 2011 International Symposium on Symbolic and Algebraic Computation ISSAC 2011, pp. 123–130. ACM
7. Gilbert, A., Indyk, P., Iwen, M.A., Schmidt, L.: Recent developments in the sparse fourier transform. IEEE Signal Process. Mag. **31**(5), 91–100 (2014)
8. Hassanieh, H., Indyk, P., Katabi, D., Price, E.: Simple and practical algorithm for sparse fourier transform. In: Proc. 23th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12), pp. 1183–1194 (2012)
9. Hassanieh, H., Adib, F., Katabi, D., Indyk, P.: Faster GPS via the sparse fourier transform. In: Proceeding Mobicom 2012, Proceedings of the 18th Annual International Conference on Mobile Computing and networking, pp. 353–364 (2012)
10. Iwen, M.A.: Combinatorial sublinear-time Fourier algorithms. Found. Comput. Math. **10**, 303–338 (2010)
11. Iwen, M.A.: Improved approximation guarantees for sublinear-time Fourier algorithms. Appl. Comput. Harmon. Anal. **34**, 57–82 (2013)
12. Janakiraman, N.T., Vem, A., Narayanan, K.R., Chamberland, J.-F.: Sub-string/pattern matching in sub-linear time using a sparse Fourier transform approach, preprint, arXiv:1704.07852v1
13. Moitra, A.: Super-resolution, extremal functions and the condition number of Vandermonde matrices. In: STOC '15 Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, pp. 821–830 (2015)
14. Montgomery, H.L., Vaughan, R.C.: Hilbert's inequality. J. London Math. Soc. (2) **8**, 73–82 (1974)
15. Lawlor, D., Wang, Y., Christlieb, A.: Adaptive sub-linear time Fourier algorithms. Adv. Adapt. Data Anal. **5**(1), 1350003 (2013)
16. Pawar, S., Ramchandran, K.: Computing a k-sparse n-length discrete Fourier transform using at most $4k$ samples and $\mathcal{O}(kk)$ complexity. IEEE Int. Symp. Inf. Theory, 464–468 (2013)
17. Potts, D., Tasche, M., Volkmer, T.: Efficient spectral estimation by MUSIC and ESPRIT with application to sparse FFT. Frontiers Appl. Math. Stat. (2016)
18. Plonka, G., Wannenwetsch, K.: A deterministic sparse FFT algorithm for vectors with small support. Numer. Algorithms **71**(4), 889–905 (2016)
19. Plonka, G., Wannenwetsch, K.: A sparse fast Fourier algorithm for real non-negative vectors. J. Comput. Appl. Math. **321**, 532–539 (2017)
20. Segal, B., Iwen, M.A.: Improved sparse Fourier approximation results: faster implementations and stronger guarantees. Numer. Algor. **63**(2), 239–263 (2013)