Deterministic sparse FFT algorithms

Gerlind Plonka1,* and Katrin Wannenwetsch1
1 Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, D-37083 Göttingen, Germany

In this paper we consider sparse signals \( x \in \mathbb{C}^N \) which are known to vanish outside a support interval of length bounded by \( m < N \). For the case that \( m \) is known, we propose a deterministic algorithm of complexity \( O(m \log m) \) for reconstruction of \( x \) from its discrete Fourier transform \( \hat{x} \in \mathbb{C}^N \).

© 2015 Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim

1 Introduction

Fast algorithms for the computation of the discrete Fourier transform of a vector of length \( N \) have been known for many years. These FFT algorithms have an algorithmic complexity of \( O(N \log N) \). Recently, there has been a stronger interest in Fourier algorithms for sparse vectors which can even achieve a sublinear complexity. Randomized sparse Fourier algorithms achieving a complexity of \( O(m \log N) \) resp. \( O(m \log m) \) for \( m \)-sparse vectors can e.g. be found in [2] resp. [3], [4]. An overview of the methods of randomized sparse Fourier transforms is given in [1].

In this paper, we present a deterministic FFT algorithm and restrict ourselves to vectors with a short support interval. Such vectors occur in different applications, such as in X-ray microscopy, where compact support is a frequently used a-priori condition in phase retrieval, as well as in computer tomography reconstructions.

Let \( x \in \mathbb{C}^N \). We define the support length \( m = |\text{supp} \, x| \) of \( x \) as the minimal integer \( m \) for which there exists a \( \mu \in \{0, \ldots, N-1\} \) such that the components \( x_k \) of \( x \) vanish for all \( k \notin I \) := \{\( \mu + r \) mod \( N, \ r = 0, \ldots, m-1 \} \). The index set \( I \) is called support interval of \( x \). We always have \( x_\mu \neq 0 \) and \( x_{\mu+m-1} \neq 0 \), but there may be zero components of \( x \) within the support interval. Observe that if \( m \leq N/2 \), the support interval and hence the first support index \( \mu \) of \( x \) is uniquely determined.

We define the discrete Fourier transform of a vector \( x \in \mathbb{C}^N \) by \( \hat{x} = F_N x \), where the Fourier matrix \( F_N \) is given by \( F_N := (\omega_N^{jk})_{j,k=0}^{N-1}, \ \omega_N := e^{-\frac{2\pi i}{N}} \). In the following, we describe a deterministic algorithm for the reconstruction of \( x \) of length \( N = 2^J \) from Fourier data \( \hat{x} \in \mathbb{C}^N \). The algorithm is based on the idea that the (at most) \( m \) nonzero components of \( x \) can already be identified from a periodization of \( x \) of length \( 2^L \geq m \). Hence for the complete reconstruction it remains to determine the support interval (i.e., the first support index) of \( x \).

2 Reconstruction of \( x \) with short support interval

Let \( N := 2^J \) for some \( J > 0 \). We define the periodizations \( x^{(j)} \in \mathbb{C}^{2^j} \) of \( x \) by

\[
x^{(j)} = (x_k^{(j)})_{k=0}^{2^j-1} = \left( \sum_{\ell=0}^{2^j-1} x_{k+2^j \ell} \right)_{k=0}^{2^j-1}
\]

(1)

for \( j = 0, \ldots, J \). Obviously, \( x^{(0)} = \sum_{k=0}^{N-1} x_k \) is the sum of all components of \( x \), \( x^{(1)} = (\sum_{k=0}^{N/2-1} x_{2k} \sum_{k=0}^{N/2-1} x_{2k+1})^T \) and \( x^{(J)} = x \). The discrete Fourier transform of the vectors \( x^{(j)}, j = 0, \ldots, J \), can be described in terms of \( \hat{x} \). According to the following lemma, it can be obtained by just picking suitable components of \( \hat{x} \).

**Lemma 2.1** For the vectors \( x^{(j)} \in \mathbb{C}^{2^j}, j = 0, \ldots, J \), in (1), we have the discrete Fourier transform

\[
\hat{x}^{(j)} := F_{2^j} x^{(j)} = (\hat{x}_{2^j-k})_{k=0}^{2^j-1},
\]

where \( \hat{x} = (\hat{x}_k)_{k=0}^{N-1} = F_N x \) is the Fourier transform of \( x \in \mathbb{C}^N \).

Assume that the Fourier data \( \hat{x} = F_N x \in \mathbb{C}^N \) and \( |\text{supp} \, x| \leq m \) for some given \( m \). Choose \( L \) such that \( 2^{L-1} < m \leq 2^L \).

By Lemma 2.1 we have \( \hat{x}^{(L+1)} = (\hat{x}_{2^L-2(k+1)} \hat{x}_{2^L+k+1})_{k=0}^{2L+1-1} \). Thus, we can compute \( x^{(L+1)} \) using inverse FFT of length \( 2^{L+1} \).

The resulting vector \( x^{(L+1)} \) has already the same support length as \( x \), since \( |\text{supp} \, x^{(L+1)}| \leq m \leq 2^L \), and for each \( k \in \{0, \ldots, 2^{L+1} - 1\} \) the sum in

\[
x_{k}^{(L+1)} = \sum_{\ell=0}^{2^{L-1}-1} x_{k+2^L \ell}
\]

(2)

© 2015 Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim

* Corresponding author: e-mail plonka@math.uni-goettingen.de
contains at most one nonvanishing term. Therefore, the support of \( x^{(L+1)} \) and its first index \( \mu^{(L+1)} \) are uniquely determined. For reconstruction of the complete vector \( x \) it is now sufficient to determine the first support index \( \mu^{(J)} = \mu \) of the support interval of \( x \). Then the components of \( x \) are given by

\[
x_{(\mu^{(J)}+k) \mod N} = \begin{cases} 
  x_{(\mu^{(L+1)}+k) \mod 2L+1}^{(L+1)} & \text{for } k = 0, \ldots, m-1, \\
  0 & \text{for } k = m, \ldots, N-1.
\end{cases}
\]

(3)

By the following theorem (cf. Theorem 3.1 in [5]), it is possible to obtain \( \mu^{(J)} \) and hence to recover \( x \) from the vector \( x^{(L+1)} \) and one additional Fourier component.

**Theorem 2.2** Let \( x \in \mathbb{C}^N, N = 2^J \), have support length \( m \) (or a support length bounded by \( m \)) with \( 2L-1 < m \leq 2L \). For \( L < J - 1 \), let \( x^{(L+1)} \) be the \( 2^{L+1} \)-periodization of \( x \). Then \( x \) can be uniquely recovered from \( x^{(L+1)} \) and one nonzero component of the vector \((\tilde{x}_{2k+1})_{k=0}^{N/2-1}\).

### 3 Sparse FFT Algorithm

We summarize the reconstruction of \( x \) from Fourier data \( \tilde{x} \) in the following algorithm.

**Algorithm 3.1** *(Sparse FFT for vectors with short support)*

**Input:** \( \tilde{x} \in \mathbb{C}^N, N = 2^J, |\text{supp} \ x| \leq m < N. \)

1. **Compute** \( L \) such that \( 2L-1 < m \leq 2L, \text{ i.e., } L : = \lfloor \log_2 m \rfloor \).
2. **If** \( L = J \) or \( L = J - 1 \), **compute** \( x = F_N^{-1} \tilde{x} \) using an FFT of length \( N \).
3. **If** \( L < J - 1 \):
   1. **Choose** \( \tilde{x}^{(L+1)} : = (\tilde{x}_{2k+1})_{k=0}^{2L+1-1} \) and **compute** \( x^{(L+1)} : = F_{2L+1}^{-1} \tilde{x}^{(L+1)} \) using an FFT of length \( 2L+1 \).
   2. **Determine** the first support index \( \mu^{(L+1)} \in \{0, \ldots, 2^{L+1} - 1\} \) of \( x^{(L+1)} \) such that \( x_{(\mu^{(L+1)})}^{(L+1)} \neq 0 \) and \( x_k^{(L+1)} = 0 \) for \( k \not\in \{(\mu^{(L+1)}) + r \mod 2L+1, r = 0, \ldots, m-1\} \).
   3. **Choose a Fourier component** \( \tilde{x}_{2k_0+1} \neq 0 \) of \( \tilde{x} \) and **compute the sum**

\[
a : = \sum_{\ell=0}^{m-1} x_{(\mu^{(L+1)}+\ell) \mod 2L+1}^{(L+1)} \omega_N^{(2k_0+1)(\mu^{(L+1)}+\ell)}.
\]

4. **Compute** \( b : = \tilde{x}_{2k_0+1}/a \) that is by construction of the form \( b = \omega_p^{2L-1} \) for some \( p \in \{0, \ldots, 2J-L-1\} \), and **find** \( \nu \in \{0, \ldots, 2^J-L-1\} \) such that \( (2k_0 + 1) \nu = p \mod 2^J-L-1 \).
5. **Set** \( \mu^{(J)} : = (\mu^{(L+1)}) + 2L+1\nu \), and \( x : = (x_k)_{k=0}^{N-1} \) with entries

\[
x_{(\mu^{(J)}+\ell) \mod N} := \begin{cases} 
  x_{(\mu^{(L+1)}+\ell) \mod 2L+1}^{(L+1)} & \text{for } \ell = 0, \ldots, m-1, \\
  0 & \text{for } \ell = m, \ldots, N-1.
\end{cases}
\]

**Output:** \( x \).

Our algorithm has an arithmetical complexity of \( \mathcal{O}(m \log m) \). This can be seen as follows: In the first step, an FFT algorithm of this complexity is performed. All further steps require at most \( \mathcal{O}(m) \) operations. Moreover, the algorithm needs less than \( 4m \) Fourier values.

The results can be found in a more detailed version in [5] where we also propose an algorithm for noisy input data as well as numerical results.

**Acknowledgements** We gratefully acknowledge the funding of this work by the DFG in the project PL 170/16-1.

**References**