BSFFT - a MATLAB toolbox for the butterfly sparse Fourier transform

Ines Melzer

September 3, 2012

Contents

1 Introduction

This manual gives a short summary, how to use the butterfly sparse Fourier transform. An introduction to the butterfly sparse Fourier transform can be found in [2]. In the following, we use the notation of [1]. For a given space dimension $d \in \mathbb{N}$, a nonharmonic bandwidth $N = 2^L$, $L \in \mathbb{N}$, a set of frequencies $\widetilde{T} = \{\boldsymbol{\xi}_k \in [0, N]^d : k = 1, \ldots, M_2\}$, a set of Fourier coefficients $\hat{u}_k \in \mathbb{C}$, $k = 1, \ldots, M_2$, and a set of evaluation nodes $\widetilde{X} = \{\boldsymbol{x}_j \in [0, N]^d : j = 1, \ldots, M_1\}$, the butterfly sparse Fourier transform computes the sums

$$u_j := u(\boldsymbol{x}_j) = \sum_{k=1}^{M_2} \hat{u}_k \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{\xi}_k \boldsymbol{x}_j / N}, \quad j = 1, \ldots, M_1. \tag{1}$$

The sums (1) with $d \geq 2$ and $M_1 = M_2 = \mathcal{O}(N^{d-1})$, and sampling sets $\widetilde{T}$, $\widetilde{X}$ on smooth $(d-1)$-dimensional manifolds can be computed in $\mathcal{O}(N^{d-1} \log N (|\log \varepsilon| + \log N)^{d+1})$ floating point operations, where $\varepsilon > 0$ denotes the target accuracy.

2 Implementation

The butterfly sparse Fourier transform is implemented for the dimensions$d = 1, 2, 3, 4$ in an object-oriented way. For readers, which are not familiar with the class concept, it is recommended to study the correspond- ing section of the Matlab help, labelled as Object-Oriented Programming. The toolbox consists of the main classes `sparse_FFT1D` for $d = 1$, `sparse_FFT2D` for $d = 2$, `sparse_FFT3D` for $d = 3$, and `sparse_FFT4D` for $d = 4$. We write `sparse_FFT*D` and mean that * can be choosen as 1,2,3,4. There are some classes called `tree1D`, `tree2D`, `tree3D`, and `tree4D`, which computes the trees of the dyadic decompositions of the domains, see [1, section 3.]. These tree classes aren't relevant for the users. Table 1 consists the properties and table 3 the methods of the `sparse_FFT*D` classes. You have to make sure, that your input data fulfil the range conditions, see table 1, because there is no check for wrong inputs implemented. In the toolbox directory are detailed `start*D.m` scripts for each dimension. Here, we will give only a short introduction how to use `mtimes`.

The user have to set the properties `N`, `MX`, `MOmega`, `p`, and `option`. If you want a higher approximation rank $p \geq 6$ you should use for `option` the property `'Ltb'` or `'Ltb*'`, because these variants are more stable, see [1] for more details. Then you create an object of class `sparse_FFT*D`,

$$\texttt{plan} = \texttt{sparse\_FFT} * \texttt{D}(\texttt{MX}, \texttt{MOmega}, \texttt{p}, \texttt{N}, \texttt{option}).$$

Afterwords you have to set the coefficient vector $\mathtt{f} = (\hat{u}_k)_{k=1,\ldots,M_2} \in \mathbb{C}^{M_2 \times 1}$ and can compute the sums $\mathtt{u} := (u(x_j))_{j=1}^{M_1}$, see (1), by applying

$$\texttt{u} = \texttt{mtimes}(\texttt{plan}, \texttt{f}).$$

For detailed examples, see the `start` scripts in the toolbox directory. All available properties and methods of an object can be listed with the Matlab functions properties and methods,

`properties(obj)` or `methods(obj)`,

and documentation is provided by the Matlab `help` and `doc` commands, for example

`help sparse_FFT1D` or `doc sparse_FFT1D`.

You can find all numerical tests of [1] in the directory `BSFFT/paper_KuMe2012`. The numerical files and figures of [1] are listed in table 4. For each `compute` file exists a `show` file, which plots the result of the computations. For more details see the m-Files.

Property	**Range**	**Description**
`N`	`N` $= 2^L, L \in \mathbb{N}$	domain parameter
`MX`	$[0, N]^{M_1 \times d}$	sampling nodes $\widetilde{X}$
`MOmega`	$[0, N]^{M_2 \times d}$	sampling nodes $\widetilde{T}$
`p`	`p` $\in \mathbb{N}$	local expansion degree
`option`	see table 2	
`invG`	$\mathbb{C}^{p \times p}$	matrix $\boldsymbol{G}^{-1}$, see [1, section 2.3.1.]
`H`	$\mathbb{C}^{p \times p}$	matrix $\boldsymbol{H}$, see [1, section 2.3.1.]
Lleft	$\mathbb{C}^{p \times p}$	Lagrange matrix for a left son box, see [1, section 2.3.2.]
Lright	$\mathbb{C}^{p \times p}$	Lagrange matrix for a right son box, see [1, section 2.3.2.]
Laglast	$\mathbb{C}^{d(M_1 \times p)}$	function values of Lagrange polynomials in `MX` just for `option='Ltb*'`, see [1, section 2.3.2.]

Table 1: Properties of the class `sparse_FFT*D` for each dimension $d = 1, 2, 3, 4$ and their default values.

option	**Description**
`'Mtb'`	monomial-type basis, see [1, section 2.3.1.]
`'Ltb'`	Lagrange-type basis, see [1, section 2.3.2.]
`'Ltb*'`	Lagrange-type basis with precomputation of the Lagrange polynomials in `MX`.

Table 2: Possible types for the property `option` of the class `sparse_FFT`.

method	**Description**
`sparse_FFT*D`	constructor
`mtimes`	Computes the sums (1).

Table 3: Methods of the class `sparse_FFT`.

figure	**file**
Figure 4.1. (a), (b)	`box1D/compute1D_relative_error_box`
Figure 4.2. (a)	`box1D/show_condition_number_G`
Figure 4.2. (b)	`box1D/show_condition_number_Lag`
Figure 4.3. (a),(b)	`compute1D_relative_error`
Figure 4.3. (c),(d)	`compute2D_relative_error`
Figure 4.4.	`compute1D_relative_error_dependence_L`
Figure 4.5.	`compute1D_times`
Figure 4.6. (a),(b)	`compute2D_times`
Figure 4.6. (c),(d)	`compute3D_times`
Figure 4.6. (e),(f)	`compute4D_times`

Table 4: Files of the numerical tests in [1].

References

[1] S. Kunis and I. Melzer. A stable and accurate butterfly sparse Fourier transform. *SIAM J. Numer. Anal.*, 50(3):1777–1800, 2012.

[2] L. Ying. Sparse Fourier transform via butterfly algorithm. *SIAM J. Sci. Comput.*, 31(3):1678–1694, 2009.