

# Automated Parameter Tuning based on RMS Errors for nonequispaced FFTs

Franziska Nestler

In this paper we study the error behavior of the well known fast Fourier transform for nonequispaced data (NFFT) with respect to the  $\mathcal{L}_2$ -norm. We compare the arising errors for different window functions and show that the accuracy of the algorithm can be significantly improved by modifying the shape of the window function. Based on the considered error estimates for different window functions we are able to state an easy and efficient method to tune the involved parameters automatically. The numerical examples show that the optimal parameters depend on the given Fourier coefficients, which are assumed not to be of a random structure or roughly of the same magnitude but rather subject to a certain decrease.

*Key words and phrases* : nonequispaced fast Fourier transform, nonuniform fast Fourier transform, NFFT, NUFFT

*2000 AMS Mathematics Subject Classification* : 65T

## 1 Introduction

A broad variety of mathematical algorithms and applications depend on the calculation of the nonequispaced discrete Fourier transform, which is a generalization of the discrete Fourier transform to nonequispaced nodes. Especially, its fast approximate realization called nonequispaced fast Fourier transform (NFFT) or rather nonuniform fast Fourier transform (NUFFT) [5, 1, 23, 25, 21, 9, 15] led to the development of a large number of fast numerical algorithms.

Basically, the NFFT, which is an approximate algorithm, consists of three steps. Using a so called window function, the given coefficients are at first deconvolved in Fourier domain. The result is transformed into spatial domain by an FFT and a discrete convolution with the window function is the final step. Thereby, the window function is chosen such that it is well localized in spatial as well as in Fourier domain. Given this property, the deconvolution step can be realized very efficiently and the resulting aliasing errors can be kept small.

In this paper, we investigate the occurrent errors measured in the  $\mathcal{L}_2$ -norm. In some numerical examples, we evaluate these errors for different window functions. We show that

---

Technische Universität Chemnitz, Faculty of Mathematics, 09107 Chemnitz, Germany

the accuracy of the algorithm can be improved by modifying the shape parameter of the window function and that the optimal value of this shape parameter very much depends on the given set of Fourier coefficients. If the input signals are assumed to be random and uncorrelated, a prediction of the optimal shape parameter is possible for certain window functions. As an example, for the Gaussian window function a convenient choice of the shape parameter has already been derived in [23] as well as in [4]. Also other windows for which the question concerning the optimal choice of the shape parameter is also interesting, as for example Kaiser-Bessel functions [8, 12], have been suggested in the literature.

However, there are many applications, where the given Fourier coefficients are not of a random structure. As an example, the NFFT can be used in order to evaluate sums of the form

$$f(y_k) := \sum_{j=1}^N \alpha_j K(y_k - x_j), \quad k = 1, \dots, M,$$

where the nodes  $y_k, x_j$ , the coefficients  $\alpha_j$  and a certain smooth kernel function  $K$  is given, see [20] for instance. The method, which is widely known as NFFT based fast summation, is based on approximating the kernel function by a trigonometric polynomial, where the corresponding Fourier coefficients are naturally subject to a certain decline. For the fast NFFT based Gauss transform [17] we have an exponential decrease of the Fourier coefficients, for instance. The NFFT based fast summation is also applied for the computation of the Coulomb energies and forces in particle systems, where the kernel function is given by  $K(r) = r^{-1}$ . This problem can also be considered subject to periodic boundary conditions, where the analytical Fourier coefficients are known and also underlie an exponential decrease, see [6, 11, 3, 18].

Thus, in our numerical examples we consider certain sets of decreasing Fourier coefficients and show that an appropriate modification of the window's shape parameter leads to substantially better results. For the Kaiser-Bessel window function the variability of the shape parameter was also considered in [7], but an adaption was not done depending on the given Fourier coefficients. In our tests we additionally compare the errors between two different deconvolution approaches. We also consider the  $\mathcal{L}_2$ -optimized deconvolution, which has already been considered in [4, Appendix A] or [13], and also give numerical evidence that only small improvements are possible by applying this optimized deconvolution scheme. Based on the error estimates, we are able to state an easy and efficient method to tune all parameters involved in the univariate NFFT algorithm. Note that it has already been observed that in some applications the NFFT with very small oversampling factors [26] or even without oversampling [3] leads to very precise approximations. The results presented in this paper confirm that in some cases an oversampling is in fact not needed.

We remark that an overall tuning, which in addition optimizes the set of parameters with respect to runtime, should depend on the used hardware. In addition, the runtime behavior regarding the window evaluation is different for the individual window functions and may also depend on the used hardware as well as on the applied variant of the NFFT (multithreaded NFFT [24], NFFT on GPUs [16], parallel NFFT [18]). However, in order to develop optimal runtime models automated parameter tuning methods, as discussed in this paper, are essential.

The outline of this paper is as follows. In Section 2 we give a short introduction to the NFFT. We start by introducing the necessary notations and then give a formula for the computation of the approximation error in the  $\mathcal{L}_2$ -norm. In Section 3 we introduce different window functions and show how the corresponding aliasing errors can be estimated. We

compare two different deconvolution approaches and point out how the choice of the window's shape parameter can influence the goodness of the approximation. Therefore we consider some univariate examples. A comparison to measured approximation errors is done in Section 4. Based on these error estimates we describe an automatic parameter tuning for the univariate case in Section 5. We continue with some remarks concerning the multivariate case and conclude with a short summary.

## 2 NFFT

In the following we give a short introduction to the NFFT in  $d$  dimensions. At first, we will introduce the necessary notations.

For some  $\mathbf{M} = (M_1, \dots, M_d) \in 2\mathbb{N}^d$  we define the index set  $\mathcal{I}_{\mathbf{M}}$  by

$$\mathcal{I}_{\mathbf{M}} := \bigotimes_{j=1}^d \{-M_j/2, \dots, M_j/2 - 1\}.$$

For two vectors  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$  and  $\mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}^d$  we define the component wise product by  $\mathbf{x} \odot \mathbf{y} := (x_1 y_1, \dots, x_d y_d) \in \mathbb{R}^d$  as well as the inner product via  $\mathbf{x} \cdot \mathbf{y} := x_1 y_1 + \dots + x_d y_d \in \mathbb{R}$ . For a vector  $\mathbf{x} \in \mathbb{R}^d$  with non vanishing components we set  $\mathbf{x}^{-1} := (x_1^{-1}, \dots, x_d^{-1}) \in \mathbb{R}^d$ .

Let some arbitrary nodes  $\mathbf{x}_j \in \mathbb{T}^d$ , where  $\mathbb{T} := \mathbb{R}/\mathbb{Z} \simeq [-1/2, 1/2)$  and  $j = 1, \dots, N$ , be given. We are now interested in a fast evaluation of a given trigonometric polynomial in the unequally spaced points  $\mathbf{x}_j$ , i.e., we want to compute the sums

$$f(\mathbf{x}_j) := \sum_{\mathbf{k} \in \mathcal{I}_{\mathbf{M}}} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \cdot \mathbf{x}_j}, \quad j = 1, \dots, N, \quad (2.1)$$

where the Fourier coefficients  $\hat{f}_{\mathbf{k}} \in \mathbb{C}$  for  $\mathbf{k} \in \mathcal{I}_{\mathbf{M}}$ ,  $\mathbf{M} \in 2\mathbb{N}^d$ , are also given.

The well known NFFT algorithm can be used to evaluate sums of the form (2.1) very efficiently with  $\mathcal{O}(|\mathcal{I}_{\mathbf{M}}| \log |\mathcal{I}_{\mathbf{M}}| + N)$  arithmetic operations. In the following, we will give an overview of the main steps.

The basic idea is to approximate the function  $f$  by a sum of translates of a one-periodic function  $\tilde{\varphi}$ , i.e.,

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) := \sum_{\mathbf{l} \in \mathcal{I}_{\sigma \odot \mathbf{M}}} g_{\mathbf{l}} \tilde{\varphi}(\mathbf{x} - \mathbf{l} \odot (\sigma \odot \mathbf{M})^{-1}), \quad (2.2)$$

where we denote by  $\sigma \geq 1$  (component wise) the oversampling factor and the coefficients  $g_{\mathbf{l}} \in \mathbb{C}$  are by now unknown. In other words, the approximate function values are obtained by computing a discrete convolution of a given window function with some coefficients  $g_{\mathbf{l}}$ , which have to be determined. In the following we denote the oversampled grid size shortly by  $\mathbf{M}_{\circ} := \sigma \odot \mathbf{M}$ . The function  $\tilde{\varphi}$  is the periodization of a window function  $\varphi$ , which is constructed based on a univariate function  $\tilde{\varphi}_{1d}$  via a tensor product scheme, i.e.,

$$\tilde{\varphi}(\mathbf{x}) := \sum_{\mathbf{r} \in \mathbb{Z}^d} \varphi(\mathbf{x} + \mathbf{r}), \quad \text{where } \varphi(\mathbf{y}) = \prod_{j=1}^d \varphi_{1d}(y_j) \text{ for } \mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}^d. \quad (2.3)$$

A transformation of  $\tilde{f}$  into Fourier space gives

$$\tilde{f}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} \hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}) e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} + \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} \hat{g}_{\mathbf{k}} c_{\mathbf{k} + \mathbf{r} \odot M_o}(\tilde{\varphi}) e^{-2\pi i (\mathbf{k} + \mathbf{r} \odot M_o) \cdot \mathbf{x}}, \quad (2.4)$$

where we denote by

$$c_{\mathbf{k}}(\tilde{\varphi}) := \int_{\mathbb{T}^d} \tilde{\varphi}(\mathbf{x}) e^{2\pi i \mathbf{k} \cdot \mathbf{x}} d\mathbf{x} = \int_{\mathbb{R}^d} \varphi(\mathbf{x}) e^{2\pi i \mathbf{k} \cdot \mathbf{x}} d\mathbf{x} = \hat{\varphi}(\mathbf{k})$$

the Fourier coefficients of  $\tilde{\varphi}$  and the discrete Fourier coefficients  $\hat{g}_{\mathbf{k}}$  are given by

$$\hat{g}_{\mathbf{k}} = \sum_{\mathbf{l} \in \mathcal{I}_{M_o}} g_{\mathbf{l}} e^{2\pi i \mathbf{k} \cdot (\mathbf{l} \odot M_o^{-1})}.$$

For the following considerations we assume that we have  $c_{\mathbf{k}}(\tilde{\varphi}) \in \mathbb{R}$ . The idea is now to choose the coefficients  $\hat{g}_{\mathbf{k}}$  appropriately. Then, the coefficients  $g_{\mathbf{l}}$  in (2.2) can be computed by a  $d$ -variate (inverse) FFT

$$g_{\mathbf{l}} = \frac{1}{|\mathcal{I}_{M_o}|} \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} \hat{g}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \cdot (\mathbf{l} \odot M_o^{-1})}$$

and the evaluation of (2.2) gives the approximate function values  $\tilde{f}(\mathbf{x}_j) \approx f(\mathbf{x}_j)$ .

However, the evaluation of the sums (2.2) might be computationally demanding unless  $\varphi$  is compactly supported on a comparable small domain or at least sufficiently small outside of it. In the latter case we replace the window function  $\varphi$  by a truncated version

$$\varphi_t(\mathbf{x}) := \varphi(\mathbf{x}) \cdot \prod_{j=1}^d \chi_{[-\frac{m}{\sigma_j M_j}, \frac{m}{\sigma_j M_j}]}(x_j) = \begin{cases} \varphi(\mathbf{x}) & : \mathbf{x} \in \bigotimes_{j=1}^d [-\frac{m}{\sigma_j M_j}, \frac{m}{\sigma_j M_j}], \\ 0 & : \text{else,} \end{cases}$$

and approximate  $f$  by

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) := \sum_{\mathbf{l} \in \mathcal{I}_{M_o}} g_{\mathbf{l}} \tilde{\varphi}_t(\mathbf{x} - \mathbf{l} \odot M_o^{-1}),$$

where now only  $m^3 \ll |\mathcal{I}_{M_o}|$  summands are not equal to zero. In the following we will refer to  $m$  as the support parameter.

It is an interesting question how to choose the unknown coefficients  $\hat{g}_{\mathbf{k}}$ . A comparison of (2.1) and (2.4), where we have to replace  $\tilde{\varphi}$  by  $\tilde{\varphi}_t$  in the case that  $\varphi$  is not compactly supported, gives  $f(\mathbf{x}) - \tilde{f}(\mathbf{x}) =$

$$\sum_{\mathbf{k} \in \mathcal{I}_M} e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} \hat{f}_{\mathbf{k}} - \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} \left( \hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t) + \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \hat{g}_{\mathbf{k}} c_{\mathbf{k} + \mathbf{r} \odot M_o}(\tilde{\varphi}_t) e^{-2\pi i (\mathbf{r} \odot M_o) \cdot \mathbf{x}} \right)$$

and

$$|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} - \hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t) \right| + \sum_{\mathbf{k} \in \mathcal{I}_{M_o} \setminus \mathcal{I}_M} |\hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t)| + \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} |\hat{g}_{\mathbf{k}} c_{\mathbf{k} + \mathbf{r} \odot M_o}(\tilde{\varphi}_t)|.$$

Thus, at first glance it seems advantageous to set

$$\hat{g}_{\mathbf{k}} := \begin{cases} \hat{d}_{\mathbf{k}} \hat{f}_{\mathbf{k}} & : \mathbf{k} \in \mathcal{I}_M \\ 0 & : \text{else} \end{cases},$$

where we define

$$\hat{d}_{\mathbf{k}} := \frac{1}{c_{\mathbf{k}}(\tilde{\varphi}_t)}, \quad (2.5)$$

cf. [21] for instance. On the other hand, estimating the error in the  $\mathcal{L}_2$ -norm

$$\|f - \tilde{f}\|_2 := \left( \int_{\mathbb{T}^d} |f(\mathbf{x}) - \tilde{f}(\mathbf{x})|^2 d\mathbf{x} \right)^{1/2},$$

which we also refer to as the root mean square (RMS) error, gives

$$\|f - \tilde{f}\|_2^2 = \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} - \hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t) \right|^2 + \sum_{\mathbf{k} \in \mathcal{I}_{M_o} \setminus \mathcal{I}_M} |\hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t)|^2 + \sum_{\mathbf{k} \in \mathcal{I}_{M_o}} \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} |\hat{g}_{\mathbf{k}} c_{\mathbf{k}+\mathbf{r} \circ M_o}(\tilde{\varphi}_t)|^2.$$

In order to minimize this error, we again have to set  $\hat{g}_{\mathbf{k}} = 0$  for  $\mathbf{k} \notin \mathcal{I}_M$ . For  $\mathbf{k} \in \mathcal{I}_M$  we assume again a linear dependence between  $\hat{g}_{\mathbf{k}}$  and  $\hat{f}_{\mathbf{k}}$ , i.e., we write  $\hat{g}_{\mathbf{k}}$  in the form

$$\hat{g}_{\mathbf{k}} = \hat{d}_{\mathbf{k}} \hat{f}_{\mathbf{k}}, \quad \hat{d}_{\mathbf{k}} \in \mathbb{R}.$$

Since we assumed that  $c_{\mathbf{k}}(\tilde{\varphi}), \hat{d}_{\mathbf{k}} \in \mathbb{R}$  we obtain

$$\|f - \tilde{f}\|_2^2 = \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} \right|^2 \left( 1 - \hat{d}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t) \right)^2 + \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} \right|^2 \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \hat{d}_{\mathbf{k}}^2 c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t).$$

Now, it is easy to determine the optimal coefficients  $\hat{d}_{\mathbf{k}}$  by differentiating with respect to  $\hat{d}_{\mathbf{k}}$  and setting the result to zero for each  $\mathbf{k}$ . We obtain

$$\begin{aligned} 0 &= -2c_{\mathbf{k}}(\tilde{\varphi}_t)(1 - \hat{d}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}_t)) + 2\hat{d}_{\mathbf{k}} \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t) \\ &= -2c_{\mathbf{k}}(\tilde{\varphi}_t) + 2\hat{d}_{\mathbf{k}} \sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t). \end{aligned}$$

Thus, the optimal choice of  $\hat{d}_{\mathbf{k}}$  with respect to the  $\mathcal{L}_2$ -norm is

$$\hat{d}_{\mathbf{k}} := \frac{c_{\mathbf{k}}(\tilde{\varphi}_t)}{\sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t)}, \quad (2.6)$$

which can also be found in [4, Appendix A] or [13], for instance. In summary, we obtain the following expressions for the error measured in the  $\mathcal{L}_2$ -norm.

$$\|f - \tilde{f}\|_2^2 = \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} \right|^2 \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \frac{c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t)}{c_{\mathbf{k}}^2(\tilde{\varphi}_t)} \quad \text{for } \hat{d}_{\mathbf{k}} := \frac{1}{c_{\mathbf{k}}(\tilde{\varphi}_t)}, \quad (2.7)$$

$$\|f - \tilde{f}\|_2^2 = \sum_{\mathbf{k} \in \mathcal{I}_M} \left| \hat{f}_{\mathbf{k}} \right|^2 \frac{\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t)}{\sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t)} \quad \text{for } \hat{d}_{\mathbf{k}} := \frac{c_{\mathbf{k}}(\tilde{\varphi}_t)}{\sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \circ M_o}^2(\tilde{\varphi}_t)}. \quad (2.8)$$

Since the coefficients  $c_{\mathbf{k}+\mathbf{r}\odot\mathbf{M}_o}(\tilde{\varphi}_t)$  for  $\mathbf{r} \neq \mathbf{0}$  are supposed to be small, the two approaches seem to be very similar.

The NFFT algorithm as described above requires  $\mathcal{O}(|\mathcal{I}_{\mathbf{M}_o}| \log |\mathcal{I}_{\mathbf{M}_o}| + m^d N)$  arithmetic operations and can be summarized as follows.

**Algorithm 2.1** (NFFT).

Input: nodes  $\mathbf{x}_j \in \mathbb{T}^d$  ( $j = 1, \dots, N$ ), coefficients  $\hat{f}_{\mathbf{k}} \in \mathbb{C}$  ( $\mathbf{k} \in \mathcal{I}_{\mathbf{M}}$ ,  $\mathbf{M} \in 2\mathbb{N}^d$ ), oversampling factor  $\sigma \in \mathbb{R}^d$ ,  $\sigma \geq \mathbf{1}$ .

i) (De-)convolution in Fourier domain:

Define the factors  $\hat{d}_{\mathbf{k}} \in \mathbb{C}$  for all  $\mathbf{k} \in \mathcal{I}_{\mathbf{M}}$ , e.g., as given in (2.5) or (2.6).

Set  $\hat{g}_{\mathbf{k}} := \hat{d}_{\mathbf{k}} \hat{f}_{\mathbf{k}}$  for all  $\mathbf{k} \in \mathcal{I}_{\mathbf{M}}$  and  $\hat{g}_{\mathbf{k}} := 0$  for  $\mathbf{k} \in \mathcal{I}_{\mathbf{M}_o} \setminus \mathcal{I}_{\mathbf{M}}$ .

Complexity:  $\mathcal{O}(|\mathcal{I}_{\mathbf{M}}|)$ .

ii) Use the (inverse) FFT for the computation of the coefficients

$$g_{\mathbf{l}} = \frac{1}{|\mathcal{I}_{\mathbf{M}_o}|} \sum_{\mathbf{k} \in \mathcal{I}_{\mathbf{M}_o}} \hat{g}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \cdot (\mathbf{l} \odot \mathbf{M}_o^{-1})}, \mathbf{l} \in \mathcal{I}_{\mathbf{M}_o}.$$

Complexity:  $\mathcal{O}(|\mathcal{I}_{\mathbf{M}_o}| \log |\mathcal{I}_{\mathbf{M}_o}|)$ .

iii) Convolution in spatial domain: Compute

$$f(\mathbf{x}_j) \approx \tilde{f}(\mathbf{x}_j) := \sum_{\mathbf{l} \in \mathcal{I}_{\mathbf{M}_o}} g_{\mathbf{l}} \tilde{\varphi}_t(\mathbf{x}_j - \mathbf{l} \odot \mathbf{M}_o^{-1})$$

for all  $j = 1, \dots, N$ .

Complexity:  $\mathcal{O}(m^d N)$ .

Output:  $\tilde{f}(\mathbf{x}_j) \approx f(\mathbf{x}_j)$  for  $j = 1, \dots, N$ . □

Using a matrix-vector notation we may write

$$\tilde{\mathbf{f}} = BFD\hat{\mathbf{f}},$$

where we define the vectors  $\tilde{\mathbf{f}} := [\tilde{f}(\mathbf{x}_j)]_{j=1}^N \in \mathbb{C}^N$  and  $\hat{\mathbf{f}} := [\hat{f}_{\mathbf{k}}]_{\mathbf{k} \in \mathcal{I}_{\mathbf{M}}} \in \mathbb{C}^{|\mathcal{I}_{\mathbf{M}}|}$ . The matrix  $D$  is a diagonal matrix with entries  $\hat{d}_{\mathbf{k}}$  and 0,  $F$  is the matrix representing the  $d$ -dimensional (inverse) FFT of size  $|\mathcal{I}_{\mathbf{M}_o}|$  and  $B$  is a block band matrix, assumed that the nodes  $\mathbf{x}_j$  are correspondingly ordered, with entries  $\tilde{\varphi}_t(\mathbf{x}_j - \mathbf{l} \odot \mathbf{M}_o^{-1})$ .

The problem of evaluating sums of the form

$$h(\mathbf{k}) := \sum_{j=1}^N f_j e^{2\pi i \mathbf{k} \cdot \mathbf{x}_j}, \quad \mathbf{k} \in \mathcal{I}_{\mathbf{M}},$$

where for each  $j = 1, \dots, N$  a coefficient  $f_j \in \mathbb{C}$  is given, can be treated very similarly. The corresponding algorithm is known as the adjoint NFFT. Note that the matrix-vector form of the adjoint NFFT is simply obtained by transposing the matrix representing the NFFT algorithm. Thus, the derivation of the algorithm is straightforward, see [21, 15]. However, since the roles of the two sets of indices have been interchanged the corresponding error analysis has to be considered in a different context, which should not be discussed in this paper.

### 3 Window functions and error estimates

There are many possible choices for an NFFT window function. In this section we aim to derive accurate bounds for the above derived error in the  $\mathcal{L}_2$ -norm, which can be evaluated in a fast way. For simplicity we restrict our considerations to the univariate case. In order to get very precise error bounds we only consider window functions for which the Fourier coefficients of the truncated window are known analytically. We will see that the derived error bounds enable a very precise prediction of the occurrent errors. This can be applied in order to develop an automatic parameter tuning, as we describe later on in more detail.

#### 3.1 Cardinal B-Splines

We define the centered cardinal B-splines by

$$B_1(x) := \begin{cases} 1 & : x \in [-1/2, 1/2) \\ 0 & : \text{else} \end{cases}$$

$$B_{n+1}(x) := (B_n * B_1)(x),$$

where we denote by  $*$  the convolution operator on  $\mathbb{R}$ . The cardinal B-spline of order  $n$  is compactly supported with  $\text{supp}B_n = [-n/2, n/2]$  and the Fourier transform is given by

$$\hat{B}_n(\xi) = \text{sinc}^n(\pi\xi),$$

where we define the sinc function

$$\text{sinc}(x) := \begin{cases} \frac{\sin x}{x} & : x \neq 0 \\ 1 & : x = 0 \end{cases}.$$

In the following we denote by

$$\Phi_n(x) := \sum_{r \in \mathbb{Z}} B_n(r)x^r = B_n(0) + \sum_{r=1}^{\infty} B_n(r)(x^r + x^{-r})$$

the well know Euler-Frobenius functions [22]. The Fourier coefficients of the cardinal B-spline  $\hat{B}_n(k)$  fulfill the relation, see [2, page 135] and [19],

$$\sum_{r \in \mathbb{Z}} |\hat{B}_n(k+r)|^2 = \Phi_{2n}(e^{-2\pi i k}) = B_{2n}(0) + 2 \sum_{r=1}^{\infty} B_{2n}(r) \cos(2\pi k r). \quad (3.1)$$

In the univariate case we define the B-spline window function as follows [1, 20].

$$\varphi(x) := B_{2m}(\sigma M x), \quad \text{supp}(\varphi) = \left[-\frac{m}{\sigma M}, \frac{m}{\sigma M}\right].$$

The corresponding Fourier coefficients are given by

$$c_k(\tilde{\varphi}) = \frac{1}{\sigma M} \text{sinc}^{2m}\left(\frac{\pi k}{\sigma M}\right).$$

With the help of (3.1) we end up with

$$\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi}) = \frac{1}{\sigma^2 M^2} \sum_{r \in \mathbb{Z}} \hat{B}_{2m}^2\left(\frac{k}{\sigma M} + r\right) = \frac{1}{\sigma^2 M^2} \Phi_{4m}\left(e^{-2\pi i k / \sigma M}\right).$$

Especially for the introduced B-spline window it is easy to derive an upper bound for sums of the form

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi})}{c_k^2(\tilde{\varphi})}.$$

Utilizing  $\sin^2(x + r\pi) = \sin^2 x$  and estimating the infinite sum by an integral we obtain [23]

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi})}{c_k^2(\tilde{\varphi})} = \sum_{r \in \mathbb{Z} \setminus \{0\}} \left( \frac{\frac{k}{\sigma M}}{\frac{k}{\sigma M} + r} \right)^{4m} < \frac{8m}{4m-1} \left( \frac{\frac{|k|}{\sigma M}}{\frac{|k|}{\sigma M} - 1} \right)^{4m}, \quad (3.2)$$

which can be evaluated in a numerically stable way. In contrast, the evaluation via

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi})}{c_k^2(\tilde{\varphi})} = \frac{\Phi_{4m}(e^{-2\pi ik/\sigma M}) - \text{sinc}^{4m}\left(\frac{\pi k}{\sigma M}\right)}{\text{sinc}^{4m}\left(\frac{\pi k}{\sigma M}\right)}$$

seems numerically unstable. The error terms obtained by using the  $\mathcal{L}_2$ -optimized coefficients  $\hat{d}_k$  can be estimated by

$$\begin{aligned} \frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi})}{\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi})} &= \frac{\text{sinc}^{4m}\left(\frac{\pi k}{\sigma M}\right)}{\Phi_{4m}(e^{-2\pi ik/\sigma M})} \sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi})}{c_k^2(\tilde{\varphi})} \\ &< \frac{\text{sinc}^{4m}\left(\frac{\pi k}{\sigma M}\right)}{\Phi_{4m}(e^{-2\pi ik/\sigma M})} \frac{8m}{4m-1} \left( \frac{\frac{|k|}{\sigma M}}{\frac{|k|}{\sigma M} - 1} \right)^{4m}, \end{aligned} \quad (3.3)$$

i.e., they are obtained by multiplying the error terms (3.2) resulting from setting  $\hat{d}_k := c_k(\tilde{\varphi})^{-1}$  by the factors

$$\frac{\text{sinc}^{4m}\left(\frac{\pi k}{\sigma M}\right)}{\Phi_{4m}(e^{-2\pi ik/\sigma M})} < 1.$$

We plot these factors for different values of  $m$  for all

$$x := |k|/\sigma M \in [0, 1/2\sigma] \subset [0, 1/2]$$

in Figure 3.1. It can be seen that only for relatively large values of  $|k|/\sigma M$  noticeable improvements can be achieved. In other words, if we choose an oversampling factor  $\sigma > 1$  or if the coefficients  $\hat{f}_k$  are comparable small for  $|k|/\sigma M \approx 1/2$ , we do not expect a significant decrease of the error measured in the  $\mathcal{L}_2$ -norm by using the optimized coefficients (2.6) instead of following the standard approach to define  $\hat{d}_k$  via (2.5). We illustrate this by the following example.

**Example 3.1.** We consider the univariate case and compare the results of the above described error estimates for two different choices of  $\hat{f}_k$ . We set  $M := 64$  and choose

$$\hat{f}_k := \frac{1}{1+k^2} \quad \text{for } k \in \mathcal{I}_{64} \quad (3.4)$$

as a first example and

$$\hat{f}_k := e^{-(k/5)^2} \quad \text{for } k \in \mathcal{I}_{64} \quad (3.5)$$



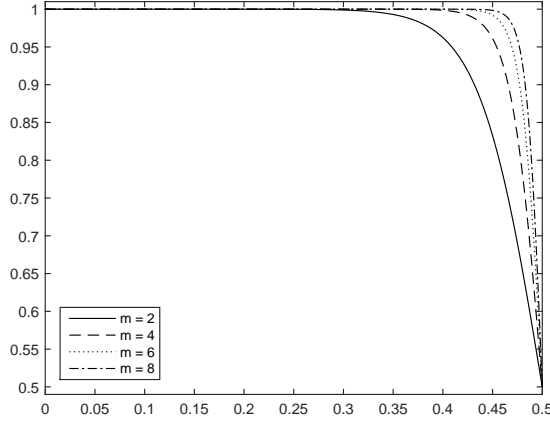


Figure 3.1:  $\text{sinc}^{4m}(\pi x)\Phi_{4m}^{-1}(e^{-2\pi i x})$  for  $x := \frac{|k|}{\sigma M} \in [0, 1/2]$  for different values of  $m$ .

in a second test. Note the difference between the two examples. The coefficients  $\hat{f}_k$  as given in (3.5) tend to zero exponentially fast, i.e., for large values of  $k$  the factors  $\hat{f}_k$  only have an insignificant influence on the overall error. In contrast, the coefficients given in (3.4) tend to zero very slowly.

In Figure 3.2 we plot the estimate of  $\|f - \tilde{f}\|_2^2$  with respect to  $m$  for the two different settings. We have

$$\|f - \tilde{f}\|_2^2 < \begin{cases} \frac{8m}{4m-1} \sum_{k \in \mathcal{I}_M} |\hat{f}_k|^2 \left( \frac{|k|}{\sigma M} - 1 \right)^{4m} & : \hat{d}_k \text{ via (2.5),} \\ \frac{8m}{4m-1} \sum_{k \in \mathcal{I}_M} |\hat{f}_k|^2 \frac{(\text{sinc} \frac{\pi k}{\sigma M})^{4m}}{\Phi_{4m}(e^{-2\pi i k/\sigma M})} \left( \frac{|k|}{\sigma M} - 1 \right)^{4m} & : \hat{d}_k \text{ via (2.6).} \end{cases} \quad (3.6)$$

For the coefficients (3.4) we observe that the error can be somewhat reduced by using the optimized coefficients (2.6) in the case  $\sigma = 1$ . Already for a small oversampling factor  $\sigma = 5/4$  the errors are almost the same. In the case that the very fast decreasing coefficients (3.5) are given, already for  $\sigma = 1$  no difference between the two approaches can be seen.  $\square$

### 3.2 Modified B-spline window

We introduce a shape parameter  $b \in 1/2\mathbb{N} = \{1/2, 1, 3/2, 2, \dots\}$  and define the modified B-spline window function  $\varphi$  by

$$\varphi(x) := B_{2b} \left( \frac{\sigma M b}{m} x \right).$$

As for the standard B-spline window we have  $\text{supp}(\varphi) = [-m/\sigma M, m/\sigma M]$ , but we also allow a different order of the B-spline, which is  $2b$ . The Fourier coefficients of the periodic version  $\tilde{\varphi}$  are given by

$$c_k(\tilde{\varphi}) = \frac{m}{\sigma M b} \text{sinc}^{2b} \left( \frac{m\pi k}{\sigma M b} \right).$$

For  $b = m$  we are in the case of the standard B-spline window and can apply the error estimation described in the previous section. In the case that we have  $m/b \in \{2, 3, 4, \dots\}$  we

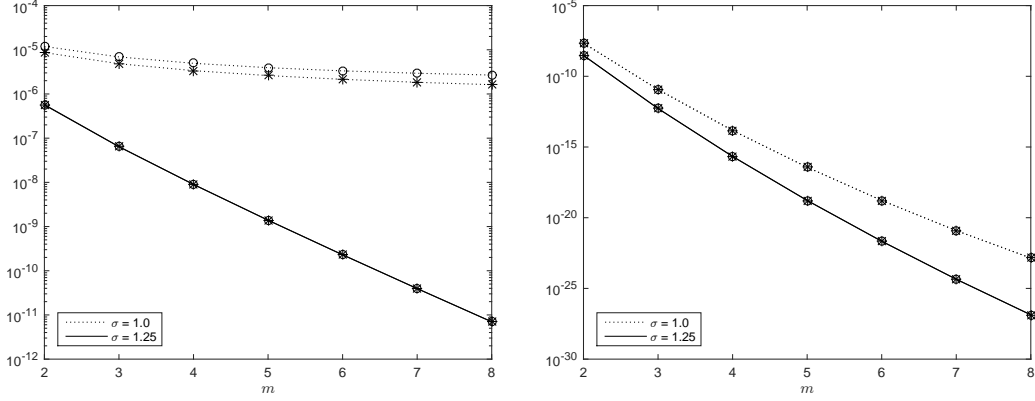


Figure 3.2: Estimated errors (3.6) for different values of  $m$  and  $\sigma$ . We set  $\hat{f}_k := (1 + k^2)^{-1}$ ,  $k \in \mathcal{I}_{64}$ , (left) and  $\hat{f}_k := e^{-(k/5)^2}$ ,  $k \in \mathcal{I}_{64}$ , (right). We compare the error terms for the two different approaches to set  $\hat{d}_k$  (variant 1 (o): define  $\hat{d}_k$  by (2.5), variant 2 (\*): set  $\hat{d}_k$  as given in (2.6)).

may exploit  $\sin^2(x + m/b \cdot r\pi) = \sin^2(x)$  in order to obtain the following estimate.

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi})}{c_k^2(\tilde{\varphi})} = \sum_{r \in \mathbb{Z} \setminus \{0\}} \left( \frac{\frac{k}{\sigma M}}{\frac{k}{\sigma M} + r} \right)^{4b} \geq \sum_{r \in \mathbb{Z} \setminus \{0\}} \left( \frac{\frac{k}{\sigma M}}{\frac{k}{\sigma M} + r} \right)^{4m} \quad (3.7)$$

i.e., we obtain the same error by choosing the window function  $\varphi(x) := B_{2b}(\sigma M x)$  with smaller support  $\text{supp}(\varphi) = [-b/\sigma M, b/\sigma M]$ . This error is obviously larger than for the case  $b = m$ . In other words, it is not necessary to consider the case that  $b \neq m$  is a divisor of  $m$ .

Note that for  $b \leq m/2$  we could also have  $c_k(\tilde{\varphi}) = 0$  for some  $k$ , i.e., the definition of  $\hat{d}_k$  via (2.5) can not be applied. As an example, if we set  $b := m/2$ ,  $\sigma := 1$  then  $c_{M/2}(\tilde{\varphi}) = 0$ . Consequently, we restrict our considerations to the case  $b > m/2$ .

If  $m/b \notin \mathbb{N}$  we proceed as follows. For some  $R_k \in \mathbb{N}$  we have

$$\begin{aligned} \sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi}) &= \frac{m^2}{\sigma^2 M^2 b^2} \sum_{r \in \mathbb{Z} \setminus \{0\}} \text{sinc}^{4b} \left( \frac{m\pi}{b} \left[ \frac{k}{\sigma M} + r \right] \right) \\ &\leq \frac{m^2}{\sigma^2 M^2 b^2} \left( \sum_{0 < |r| \leq R_k} \text{sinc}^{4b} \left( \frac{m\pi}{b} \left[ \frac{k}{\sigma M} + r \right] \right) + \sum_{|r| > R_k} \frac{1}{\left( \frac{m\pi}{b} \right)^{4b} \left( \frac{k}{\sigma M} + r \right)^{4b}} \right), \end{aligned}$$

where

$$\begin{aligned} \sum_{|r| > R_k} \frac{1}{\left( \frac{k}{\sigma M} + r \right)^{4b}} &< \int_{R_k}^{\infty} \frac{dr}{\left( \frac{k}{\sigma M} + r \right)^{4b}} + \int_{R_k}^{\infty} \frac{dr}{\left( \frac{k}{\sigma M} - r \right)^{4b}} \\ &= \frac{1}{4b-1} \left( \frac{1}{\left( \frac{k}{\sigma M} + R_k \right)^{4b-1}} - \frac{1}{\left( \frac{k}{\sigma M} - R_k \right)^{4b-1}} \right). \end{aligned}$$

In summary we have

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi}) < s \left( \frac{k}{\sigma M} \right),$$

where we set

$$s\left(\frac{k}{\sigma M}\right) := \frac{m^2}{\sigma^2 M^2 b^2} \left( \sum_{0 < |r| \leq R_k} \operatorname{sinc}^{4b}\left(\frac{m\pi}{b} \left[\frac{k}{\sigma M} + r\right]\right) + \frac{\left(\frac{k}{\sigma M} + R_k\right)^{1-4b} - \left(\frac{k}{\sigma M} - R_k\right)^{1-4b}}{\left(\frac{m\pi}{b}\right)^{4b} (4b-1)} \right)$$

for some  $R_k \in \mathbb{N}$ . In order to get a precise estimate we suggest to proceed as follows.

- i)  $R_k := 1$ .
- ii) Compute  $s\left(\frac{k}{\sigma M}\right) := \frac{m^2}{\sigma^2 M^2 b^2} \sum_{0 < |r| \leq R_k} \operatorname{sinc}^{4b}\left(\frac{m\pi}{b} \left[\frac{k}{\sigma M} + r\right]\right)$ .
- iii) Set  $r\left(\frac{k}{\sigma M}\right) := \frac{m^2}{\sigma^2 M^2 b^2} \frac{\left(\frac{k}{\sigma M} + R_k\right)^{1-4b} - \left(\frac{k}{\sigma M} - R_k\right)^{1-4b}}{\left(\frac{m\pi}{b}\right)^{4b} (4b-1)}$ .
- iv) While  $r\left(\frac{k}{\sigma M}\right) \geq s\left(\frac{k}{\sigma M}\right)$ :
  - a)  $R_k := R_k + 1$ .
  - b)  $s\left(\frac{k}{\sigma M}\right) := s\left(\frac{k}{\sigma M}\right) + \frac{m^2}{\sigma^2 M^2 b^2} \operatorname{sinc}^{4b}\left(\frac{m\pi}{b} \left[\frac{k}{\sigma M} \pm R_k\right]\right)$ .
  - c)  $r\left(\frac{k}{\sigma M}\right) := \frac{m^2}{\sigma^2 M^2 b^2} \frac{\left(\frac{k}{\sigma M} + R_k\right)^{1-4b} - \left(\frac{k}{\sigma M} - R_k\right)^{1-4b}}{\left(\frac{m\pi}{b}\right)^{4b} (4b-1)}$ .
- v)  $s\left(\frac{k}{\sigma M}\right) := s\left(\frac{k}{\sigma M}\right) + r\left(\frac{k}{\sigma M}\right)$ .

Note that the estimation can be tuned to an even higher accuracy by substituting the condition  $r\left(\frac{k}{\sigma M}\right) \geq s\left(\frac{k}{\sigma M}\right)$  by  $r\left(\frac{k}{\sigma M}\right) \geq \gamma \cdot s\left(\frac{k}{\sigma M}\right)$ , where  $\gamma \in (0, 1)$ .

In order to estimate

$$\frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi})}{\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi})}$$

we use (3.3) in the case  $b = m$ . Otherwise we exploit that a function of the form  $f(y) = \frac{y}{c^2 + y}$  is monotonically increasing and obtain

$$\frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi})}{\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi})} < \frac{s\left(\frac{k}{\sigma M}\right)}{c_k^2(\tilde{\varphi}) + s\left(\frac{k}{\sigma M}\right)}. \quad (3.8)$$

**Example 3.2.** We consider the case  $m = 4$ . For different shape parameters  $b \in 1/2\mathbb{N}$ , we plot the above derived estimates of the terms

$$\frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi})}{\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi})} = \frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} \operatorname{sinc}^{4b}\left(\frac{m\pi}{b} [x + r]\right)}{\sum_{r \in \mathbb{Z}} \operatorname{sinc}^{4b}\left(\frac{m\pi}{b} [x + r]\right)}, \quad (3.9)$$

where we set  $|x| \leq 1/2$ , in Figure 3.3. Since we only expect small differences between the two deconvolution approaches as well as for overview purposes we only plot the error terms for one of the two variants.

It seems not reasonable to use a shape parameter  $b > m$ . Depending on the given coefficients  $\hat{f}_k$  and the chosen oversampling factor, a shape parameter  $b \in \{5/2, 3, 7/2, 4\}$  is supposed to be optimal.  $\square$

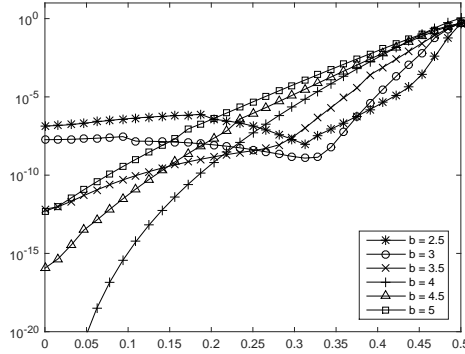


Figure 3.3: Estimated error terms (3.9) for  $m = 4$  and different shape parameters  $b$ . We plot the results with respect to  $x := \frac{|k|}{\sigma M} \in [0, 1/2]$ .

**Example 3.3.** We consider again the two sets of Fourier coefficients as given in (3.4) as well as (3.5). We estimate the quadratic error

$$\|f - \tilde{f}\|_2^2$$

as described above, for  $m \in \{4, 8\}$  with different shape parameters  $b \in \frac{1}{2}\mathbb{N}$  and oversampling factors  $\sigma$ . We plot the results in Figures 3.4 and 3.5, respectively. For the slow decreasing factors (3.4), a shape parameter  $b < m$  is optimal in most cases. In contrast, for the very fast decreasing coefficients (3.5) the minimal error is in most cases obtained by setting  $b = m$ . As expected, we only see small differences between the two deconvolution schemes.  $\square$

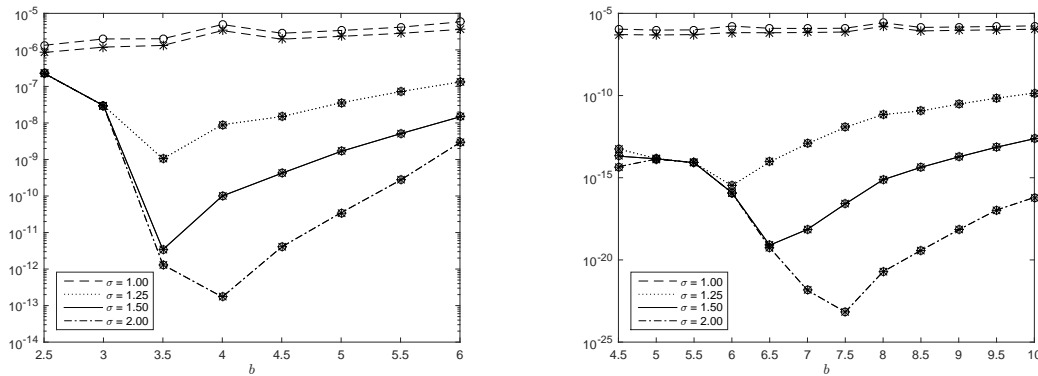


Figure 3.4: Estimate of  $\|f - \tilde{f}\|_2^2$  for the modified B-spline window, where we choose the set of Fourier coefficients  $\hat{f}_k$  given by (3.4). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 4$  (left) and  $m = 8$  (right), for different oversampling factors  $\sigma$ . We compare the error terms for the two different approaches to set  $\hat{d}_k$  (variant 1 (o): define  $\hat{d}_k$  by (2.5), variant 2 (\*): set  $\hat{d}_k$  as given in (2.6)).

### 3.3 Bessel window

In the following we consider a window function which is constructed based on the Kaiser-Bessel NFFT window, which was introduced in [20, Appendix]. In order to get a window function  $\varphi$  with compact support we interchange the roles of time and frequency domain.

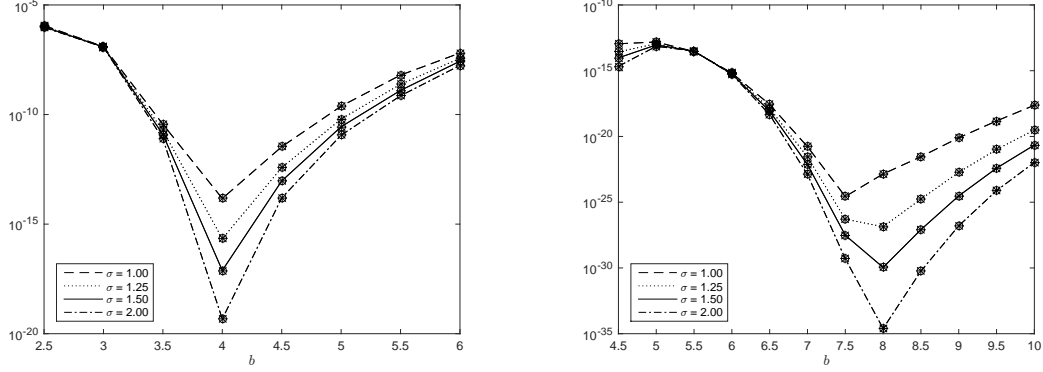


Figure 3.5: Estimate of  $\|f - \tilde{f}\|_2^2$  for the modified B-spline window, where we choose the set of Fourier coefficients  $\hat{f}_k$  given by (3.5). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 4$  (left) and  $m = 8$  (right), for different oversampling factors  $\sigma$ . We compare the error terms for the two different approaches to set  $\hat{d}_k$  (variant 1 (o): define  $\hat{d}_k$  by (2.5), variant 2 (\*): set  $\hat{d}_k$  as given in (2.6)).

We refer to the resulting function as the Bessel ( $I_0$ ) window function, which is also found under the name Kaiser-Bessel function in the literature [14, 8, 12].

For some shape parameter  $b > 0$  we define the Bessel window function by

$$\varphi(x) := \frac{1}{2} \begin{cases} I_0 \left( b\sqrt{m^2 - \sigma^2 M^2 x^2} \right) & : |x| \leq \frac{m}{\sigma M}, \\ 0 & : \text{else,} \end{cases}$$

where  $I_0$  denotes the modified zero-order Bessel function. The corresponding Fourier coefficients are of the form

$$c_k(\tilde{\varphi}) = \frac{1}{\sigma M} \begin{cases} \frac{\sinh \left( m\sqrt{b^2 - 4\pi^2 k^2 / (\sigma^2 M^2)} \right)}{\sqrt{b^2 - 4\pi^2 k^2 / (\sigma^2 M^2)}} & : |k| \leq \frac{\sigma M b}{2\pi}, \\ m \operatorname{sinc} \left( m\sqrt{4\pi^2 k^2 / (\sigma^2 M^2) - b^2} \right) & : \text{else.} \end{cases}$$

In the univariate case, the error sums  $\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi})$  can be estimated as follows. For  $R_k > \frac{|k|}{\sigma M} + \frac{b}{2\pi}$  we have

$$\begin{aligned} \sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi}) &= \sum_{1 < |r| \leq R_k} c_{k+r\sigma M}^2(\tilde{\varphi}) + \frac{m^2}{\sigma^2 M^2} \sum_{r=R_k+1}^{\infty} \operatorname{sinc}^2 \left( m\sqrt{4\pi^2 (|k|/\sigma M \pm r)^2 - b^2} \right) \\ &\leq \sum_{1 < |r| \leq R_k} c_{k+r\sigma M}^2(\tilde{\varphi}) + \frac{1}{\sigma^2 M^2} \sum_{r=R_k+1}^{\infty} \frac{1}{4\pi^2 (|k|/\sigma M \pm r)^2 - b^2} \\ &< \sum_{1 < |r| \leq R_k} c_{k+r\sigma M}^2(\tilde{\varphi}) + \frac{1}{\sigma^2 M^2} \int_{R_k}^{\infty} \frac{dr}{4\pi^2 (|k|/\sigma M \pm r)^2 - b^2}, \end{aligned}$$

where integrals can be computed by

$$\begin{aligned} \int_{R_k}^{\infty} \frac{dr}{4\pi^2(|k|/\sigma M \pm r)^2 - b^2} &= \frac{1}{2\pi} \left( \int_{2\pi(|k|/\sigma M + R_k)}^{\infty} + \int_{-\infty}^{2\pi(|k|/\sigma M - R_k)} \right) \frac{dr}{r^2 - b^2} \\ &= \frac{\ln \left| \frac{2\pi(|k|/\sigma M - R_k) - b}{2\pi(|k|/\sigma M - R_k) + b} \right| + \ln \left| \frac{2\pi(|k|/\sigma M + R_k) + b}{2\pi(|k|/\sigma M + R_k) - b} \right|}{4\pi b}. \end{aligned}$$

In order to get a precise estimate we suggest to proceed as follows, cf. Section 3.2.

- i)  $R_k := \left\lceil \frac{|k|}{\sigma M} + \frac{b}{2\pi} \right\rceil$ .
- ii) Set  $s\left(\frac{k}{\sigma M}\right) := \sum_{1 < |r| \leq R_k} c_{k+r\sigma M}^2(\tilde{\varphi})$ .
- iii) Set  $r\left(\frac{k}{\sigma M}\right) := \frac{1}{4\pi b \sigma^2 M^2} \left( \ln \left| \frac{2\pi(|k|/\sigma M - R_k) - b}{2\pi(|k|/\sigma M - R_k) + b} \right| + \ln \left| \frac{2\pi(|k|/\sigma M + R_k) + b}{2\pi(|k|/\sigma M + R_k) - b} \right| \right)$ .
- iv) While  $r\left(\frac{k}{\sigma M}\right) \geq s\left(\frac{k}{\sigma M}\right)$ :
  - a)  $R_k := R_k + 1$ .
  - b)  $s\left(\frac{k}{\sigma M}\right) := s\left(\frac{k}{\sigma M}\right) + c_{k \pm R_k \sigma M}^2(\tilde{\varphi})$ .
  - c)  $r\left(\frac{k}{\sigma M}\right) := \frac{1}{4\pi b \sigma^2 M^2} \left( \ln \left| \frac{2\pi(|k|/\sigma M - R_k) - b}{2\pi(|k|/\sigma M - R_k) + b} \right| + \ln \left| \frac{2\pi(|k|/\sigma M + R_k) + b}{2\pi(|k|/\sigma M + R_k) - b} \right| \right)$ .
- v)  $s\left(\frac{k}{\sigma M}\right) := s\left(\frac{k}{\sigma M}\right) + r\left(\frac{k}{\sigma M}\right)$ .

If we define the coefficients  $\hat{d}_k$  via (2.5),  $s(k/\sigma M)$  gives an upper bound for the corresponding error terms. In the case that we use the optimized coefficients (2.6), we can again apply (3.8).

**Example 3.4.** We consider the univariate case and investigate the behavior of the error in the  $\mathcal{L}_2$ -norm with respect to  $b$ . By default, the shape parameter  $b$  is set to [20, Appendix]

$$b_0 := 2\pi(1 - 1/2\sigma).$$

If we choose  $\sigma \in \{1, 5/4\}$  we obtain the following standard values for  $b$ .

$$b_0 \approx \begin{cases} 3.14 & : \sigma = 1, \\ 3.77 & : \sigma = 5/4. \end{cases}$$

In this example we consider again the two different sets of Fourier coefficients given in (3.4) and (3.5), respectively. For the slowly decreasing coefficients (3.4) the estimated errors are plotted in the Figure 3.6. For  $\sigma = 1$  we can see a small difference in the errors obtained for the two different approaches to set  $\hat{d}_k$ . The optimal values for  $b$  nearly coincide with the suggested default values  $b_0$ .

The results for the coefficients (3.5) are plotted in Figure 3.7. The predicted errors for the two variants to set  $\hat{d}_k$  are nearly the same. On the other hand, the optimal values for  $b$  considerably differ from  $b_0$ .  $\square$

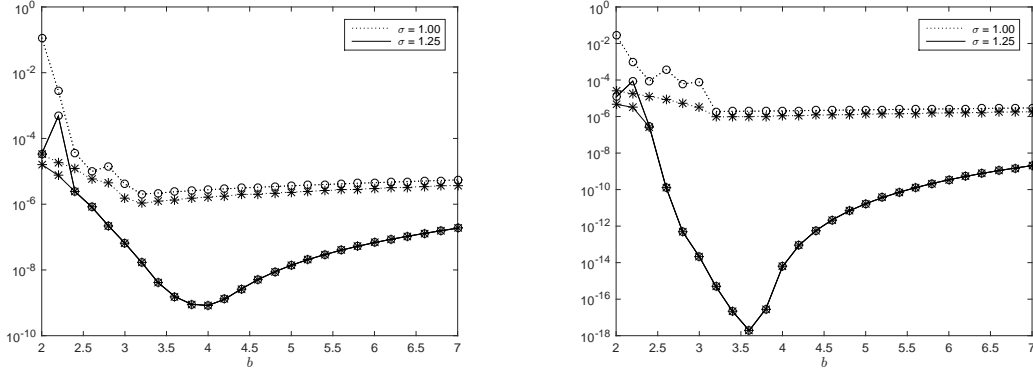


Figure 3.6: Estimate of  $\|f - \tilde{f}\|_2^2$  for the Bessel window, where we choose the set of Fourier coefficients  $\hat{f}_k$  as given by (3.4). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 3$  (left) and  $m = 6$  (right), for  $\sigma \in \{1, 5/4\}$ . We compare the obtained errors for the two variants to set the coefficients  $\hat{d}_k$  (variant 1 (o): define  $\hat{d}_k$  by (2.5), variant 2 (\*): set  $\hat{d}_k$  as given in (2.6)).

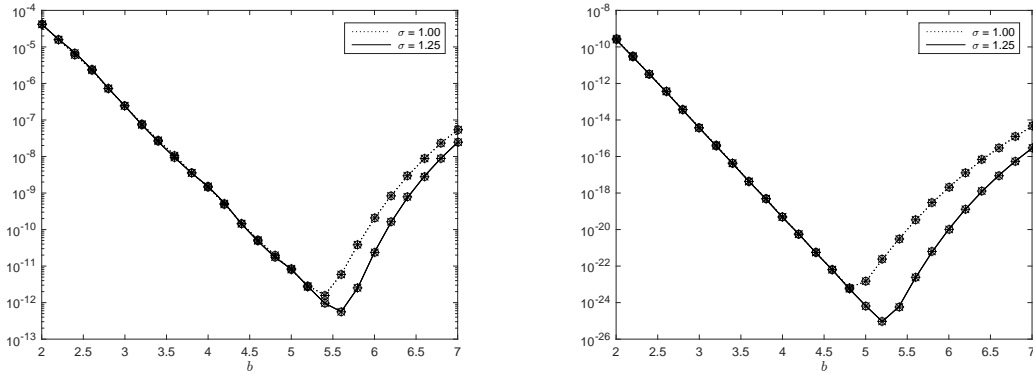


Figure 3.7: Estimate of  $\|f - \tilde{f}\|_2^2$  for the Bessel window, where we choose the set of Fourier coefficients  $\hat{f}_k$  as given by (3.5). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 3$  (left) and  $m = 6$  (right), for  $\sigma \in \{1, 5/4\}$ . We compare the obtained errors for the two variants to set the coefficients  $\hat{d}_k$  (variant 1 (o): define  $\hat{d}_k$  by (2.5), variant 2 (\*): set  $\hat{d}_k$  as given in (2.6)).

### 3.4 Gaussian window function

For some shape parameter  $b > 0$  we define the Gaussian window function in the univariate case by [5, 20, 9]

$$\varphi(x) = \frac{1}{\sqrt{\pi b}} e^{-\sigma^2 M^2 x^2 / b}.$$

The Fourier coefficients of the periodic version  $\tilde{\varphi}$  are given by

$$c_k(\tilde{\varphi}) = \frac{1}{\sigma M} e^{-b\pi^2 k^2 / (\sigma^2 M^2)}.$$

The Fourier coefficients of the truncated version  $\tilde{\varphi}_t := \tilde{\varphi} \cdot \tilde{\chi}_{[-m/\sigma M, m/\sigma M]}$  can be expressed by

$$\begin{aligned} c_k(\tilde{\varphi}_t) = \hat{\varphi}_t(k) &= \frac{1}{\sqrt{\pi b}} \int_{-m/\sigma M}^{m/\sigma M} e^{-\sigma^2 M^2 x^2 / b^2} e^{2\pi i k x} dx \\ &= \frac{e^{-b\pi^2 k^2 / (\sigma^2 M^2)}}{\sqrt{\pi b}} \int_{-m/\sigma M}^{m/\sigma M} e^{-[\sigma M x / \sqrt{b} - i\pi k \sqrt{b}(\sigma M)]^2} dx \\ &= \frac{e^{-b\pi^2 k^2 / (\sigma^2 M^2)}}{2\sigma M} \left[ \operatorname{erf} \left( \frac{m}{\sqrt{b}} + i \frac{\pi k \sqrt{b}}{\sigma M} \right) + \operatorname{erf} \left( \frac{m}{\sqrt{b}} - i \frac{\pi k \sqrt{b}}{\sigma M} \right) \right] \\ &= \frac{e^{-b\pi^2 k^2 / (\sigma^2 M^2)}}{\sigma M} \operatorname{Re} \left[ \operatorname{erf} \left( \frac{m}{\sqrt{b}} + i \frac{\pi k \sqrt{b}}{\sigma M} \right) \right], \end{aligned}$$

where we denote by  $\operatorname{erf}$  the well known error function.

**Example 3.5.** For the univariate case we consider the following three possibilities to set  $\hat{d}_k$  and give some comparisons for the Gaussian window function.

$$\int_{\mathbb{T}} |f(x) - \tilde{f}(x)|^2 dx = \sum_{k \in \mathcal{I}_M} |\hat{f}_k|^2 \cdot \begin{cases} \left(1 - \frac{c_k^2(\tilde{\varphi}_t)}{c_k^2(\tilde{\varphi})}\right)^2 + \sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi}_t)}{c_k^2(\tilde{\varphi})} & : \hat{d}_k = \frac{1}{c_k(\tilde{\varphi})}, \\ \sum_{r \in \mathbb{Z} \setminus \{0\}} \frac{c_{k+r\sigma M}^2(\tilde{\varphi}_t)}{c_k^2(\tilde{\varphi}_t)} & : \hat{d}_k \text{ via (2.5)}, \\ \frac{\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi}_t)}{\sum_{r \in \mathbb{Z}} c_{k+r\sigma M}^2(\tilde{\varphi}_t)} & : \hat{d}_k \text{ via (2.6)}. \end{cases}$$

Commonly, the NFFT deconvolution step is done by using the Fourier coefficients of the non truncated Gaussian. Since the convolution in spatial domain is done with a truncated Gaussian, it seems reasonable to use the Fourier coefficients of the truncated function, see Section 2. In order to compare the two approaches, we here also consider the variant to set  $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ .

The Fourier coefficients of the Gaussian window function tend to zero exponentially fast with growing argument so that the sums

$$\sum_{r \in \mathbb{Z} \setminus \{0\}} c_{k+r\sigma M}^2(\tilde{\varphi}_t)$$

are short and can be evaluated directly. In the following we evaluate the above error terms for different  $\hat{f}_k$  as well as for different shape parameters  $b$ . We will see that the optimal value for the shape parameter depends on both, the given coefficients  $\hat{f}_k$  and the parameters used within the NFFT ( $m, \sigma$ , definition of  $\hat{d}_k$ ).

In many applications, the shape parameter  $b$  is chosen as follows

$$b_0 := \frac{2\sigma}{2\sigma - 1} \frac{m}{\pi},$$

see [23, 4, 9]. For example, if we choose  $m \in \{3, 6\}$  combined with  $\sigma \in \{1, 5/4\}$  we obtain the values listed in Table 3.1.

In the following we consider again the two different sets of Fourier coefficients as defined in (3.4) and (3.5), respectively. We evaluated the quadratic error  $\|f - \tilde{f}\|_2^2$  via the sums stated above parameter sets. For fixed  $m$  and  $\sigma$ , the results are plotted in the following figures.



	$\sigma = 1$	$\sigma = 5/4$
$m = 3$	$b_0 \approx 1.91$	$b_0 \approx 1.59$
$m = 6$	$b_0 \approx 3.82$	$b_0 \approx 3.18$

Table 3.1: Value of  $b_0$  for different combinations of  $m$  and  $\sigma$ .

In most cases, the optimal shape parameter differs from the standard values, which are given in Table 3.1. Which value for  $b$  is optimal, obviously depends on the given Fourier coefficients  $\hat{f}_k$  as well as on the chosen oversampling factor  $\sigma$ .

In general, better results are obtained by using the Fourier coefficients of the truncated Gauss window. For the coefficients (3.4) we obtain slightly better results by using the optimized factors (2.6) if  $\sigma = 1$ . In the case that the very rapidly decreasing coefficients (3.5) are given, the two variants (2.5) and (2.6) again produce almost the same errors, already for  $\sigma = 1$ .

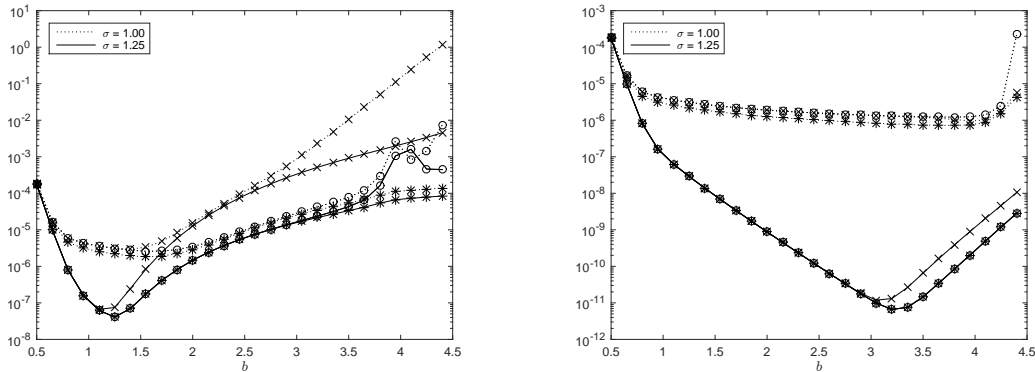


Figure 3.8: Estimate of  $\|f - \tilde{f}\|_2^2$  for the Gaussian window, where we choose the set of Fourier coefficients  $\hat{f}_k$  as given by (3.4). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 3$  (left) and  $m = 6$  (right), for  $\sigma \in \{1, 5/4\}$ . We compare the obtained errors for the three variants to set the coefficients  $\hat{d}_k$  (variant 1 (x):  $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ , variant 2 (o):  $\hat{d}_k$  via (2.5), variant 3 (\*):  $\hat{d}_k$  via (2.6)).

□

### 3.5 Comparison

In the current section we investigated the error of the NFFT in the  $\mathcal{L}_2$ -norm. Some concrete examples for the univariate case showed that, especially in the case that the given Fourier coefficients do not decrease very rapidly and  $\sigma \approx 1$ , the accuracy can be somewhat improved by using the optimized coefficients (2.6) within the NFFT (Algorithm 2.1).

Furthermore, in many cases the optimal values for the shape parameter  $b$  differ from the corresponding default values, which are widely used. Obviously, the optimal shape parameter depends on both, the given Fourier coefficients as well as on the NFFT parameters  $m$  and  $\sigma$ .

We also introduced the modified B-spline window, which allows us to choose the order of the B-spline independently from the support parameter  $m$ .

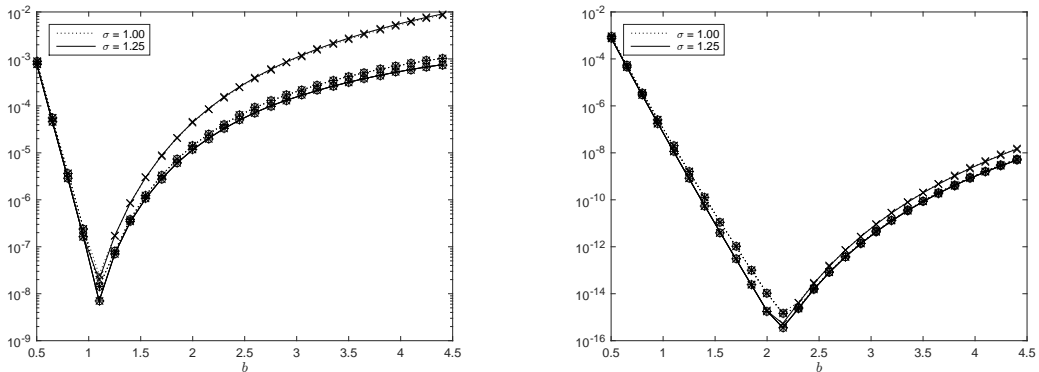


Figure 3.9: Estimate of  $\|f - \tilde{f}\|_2^2$  for the Gaussian window, where we choose the set of Fourier coefficients  $\hat{f}_k$  as given by (3.5). We plot the error with respect to the shape parameter  $b$ , where we set  $m = 3$  (left) and  $m = 6$  (right), for  $\sigma \in \{1, 5/4\}$ . We compare the obtained errors for the three variants to set the coefficients  $\hat{d}_k$  (variant 1 (x):  $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ , variant 2 (o):  $\hat{d}_k$  via (2.5), variant 3 (\*):  $\hat{d}_k$  via (2.6)).

For a comparison of all introduced window functions, we consider again the two sets of Fourier coefficients (3.4) and (3.5). As described in the previous paragraphs, we estimate the quadratic error (2.8), where  $d = 1$ , for  $m \in \{2, \dots, 8\}$  and  $\sigma \in \{1, 5/4\}$ . We plot the results in Figure 3.10 (first row). For each  $m$  we inserted different values for the shape parameter  $b$  and picked out the result for which the obtained error was minimal. The corresponding optimal values for the shape parameter  $b_{\text{opt}}$  are also plotted in Figure 3.10 (second row). For comparison, we also plot the corresponding default values  $b_0$  of the shape parameter. For the (modified) B-spline window the default value is  $b_0 = m$  (standard B-spline window).

In the case of the very slowly decreasing coefficients (3.4), the Bessel window always yields the smallest error. For the coefficients (3.5) the standard B-spline window performs better if  $m$  is very small. For even faster decreasing Fourier coefficients we expect a better performance of the B-spline window.

## 4 Verification of the theoretical estimates

We used a simple implementation of the univariate NFFT with the introduced window functions in MATLAB in order to verify the theoretical estimates.

We revisit the examples presented in Section 3 and compare the theoretical error estimates with experimental measurements. We compute the sums

$$\tilde{f}(x_j) \approx f(x_j) = \sum_{k \in \mathcal{I}_M} \hat{f}_k e^{-2\pi i k x_j}, \quad j = 1, \dots, N,$$

where we consider the two sets of Fourier coefficients (3.4) and (3.5), for  $N = 500$  randomly chosen nodes  $x_j \in \mathbb{T}$ ,  $j = 1, \dots, N$ . Instead of the quadratic error  $\|f - \tilde{f}\|_2^2$  we compute

$$\Delta f^2 := \frac{1}{N} \sum_{j=1}^N [f(x_j) - \tilde{f}(x_j)]^2. \quad (4.1)$$

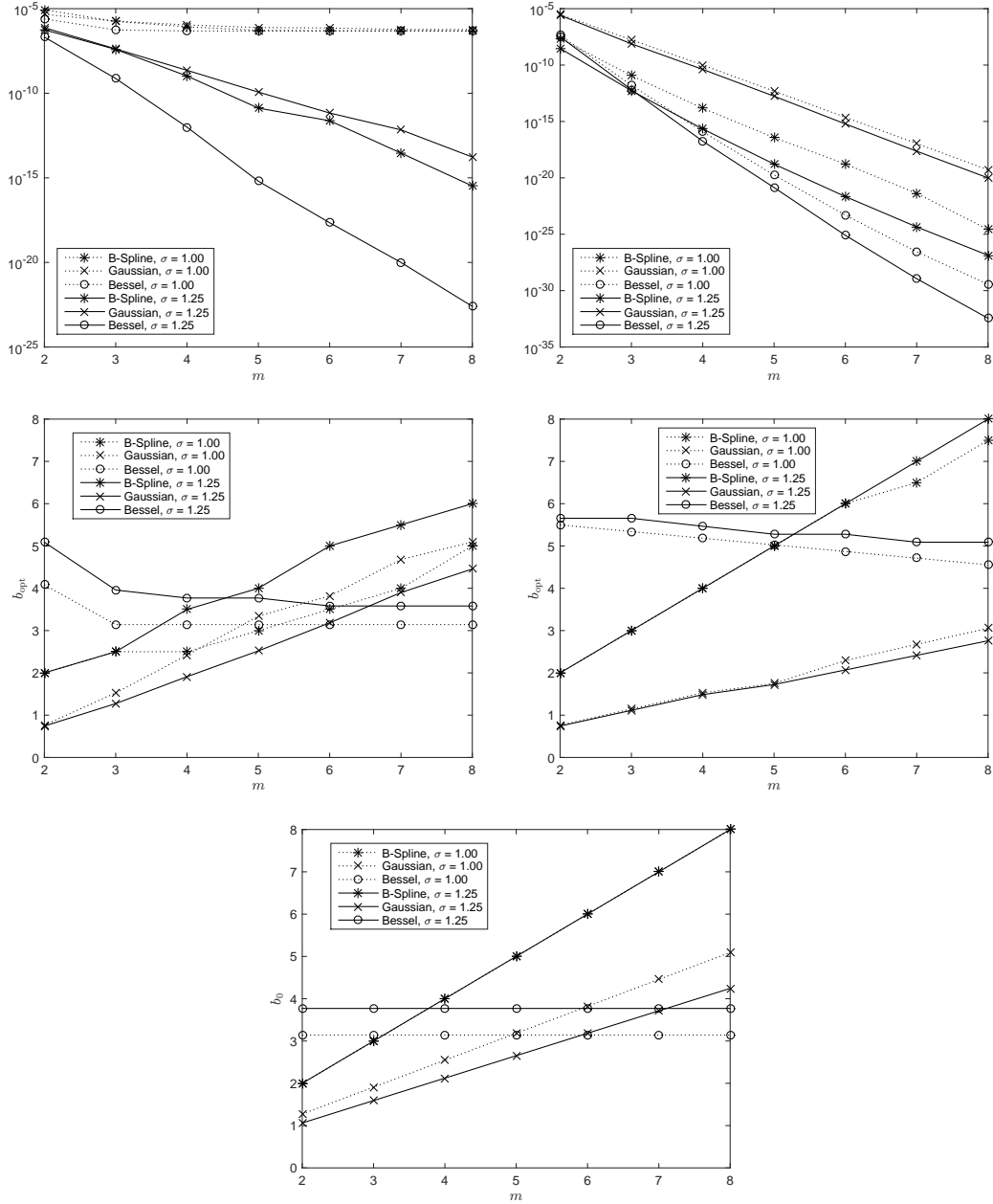


Figure 3.10: First row: Estimated quadratic errors (2.8) for the sets of Fourier coefficients (3.4) and (3.5) (from left to right) with respect to  $m$ . We used different window functions and oversampling factors (see legend).

Second row: Corresponding optimal values for the shape parameter  $b$ .

Bottom: Default values for the shape parameter  $b_0$ .

Since we have seen in the last section that the optimized deconvolution scheme (2.6) does not serve clear benefits compared to the standard approach (2.5), we only run our algorithm applying the standard deconvolution (symbolized by  $\circ$ ). For the Gaussian window function we also plot the results which are obtained by using the Fourier coefficients of the non truncated function, i.e., we set  $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$  (denoted by  $\times$ ) instead of  $\hat{d}_k := c_k^{-1}(\tilde{\varphi}_t)$ . Note that the plots concerning the theoretical estimates still also contain the results for the optimized deconvolution ( $*$ ).

We can see that the theoretical estimates match quite well with the measured errors (4.1). Of course, the measured errors are in some cases somewhat smaller than the predicted errors, which is due to the fact that the computation of the predicted errors is based on computing upper bounds of the involved error terms, see Section 3.

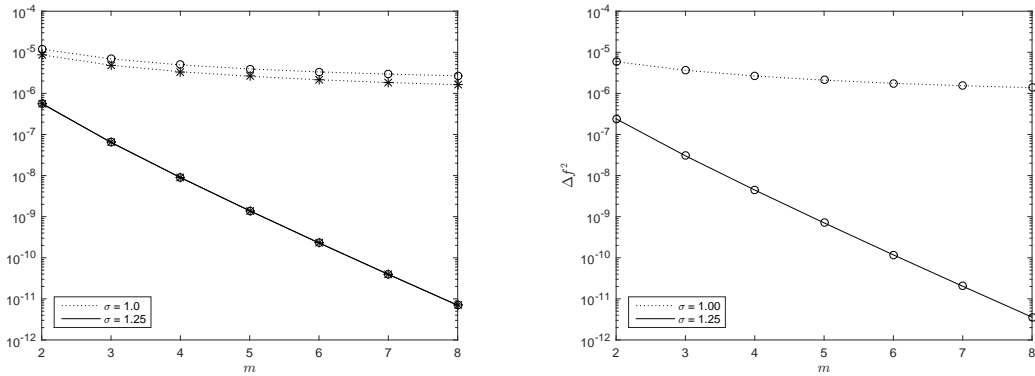


Figure 4.1: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **standard B-spline window**. We used the set of Fourier coefficients (3.4) and plot the results for different oversampling factors  $\sigma$  with respect to  $m$ .

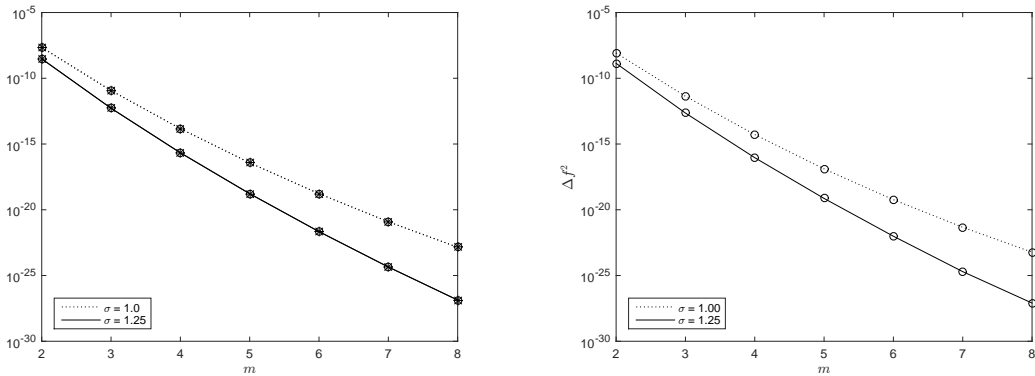


Figure 4.2: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **standard B-spline window**. We used the set of Fourier coefficients (3.5) and plot the results for different oversampling factors  $\sigma$  with respect to  $m$ .

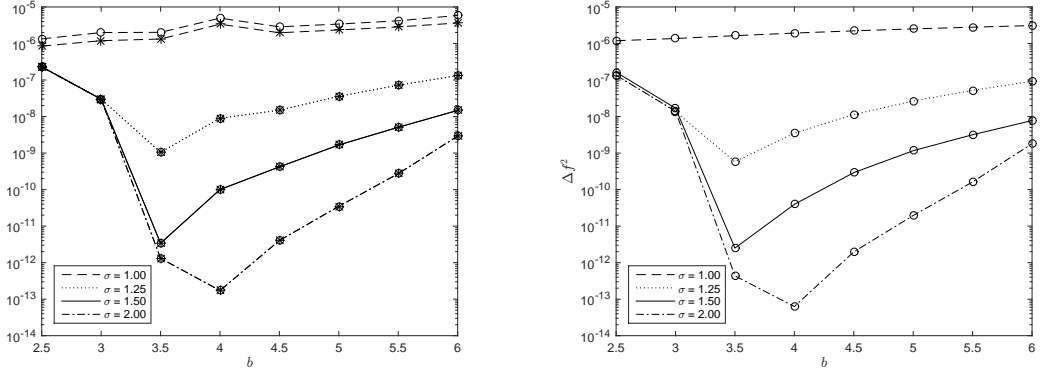


Figure 4.3: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **modified B-spline window**. We used the set of Fourier coefficients (3.4), set  $m := 4$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

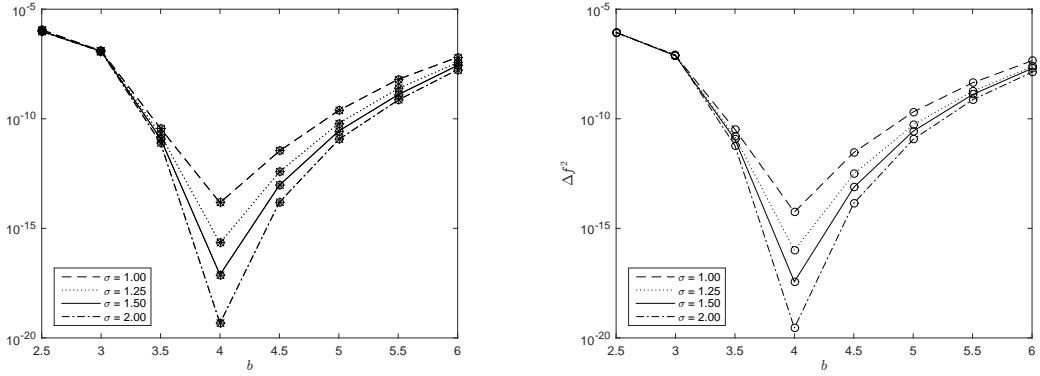


Figure 4.4: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **modified B-spline window**. We used the set of Fourier coefficients (3.5), set  $m := 4$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

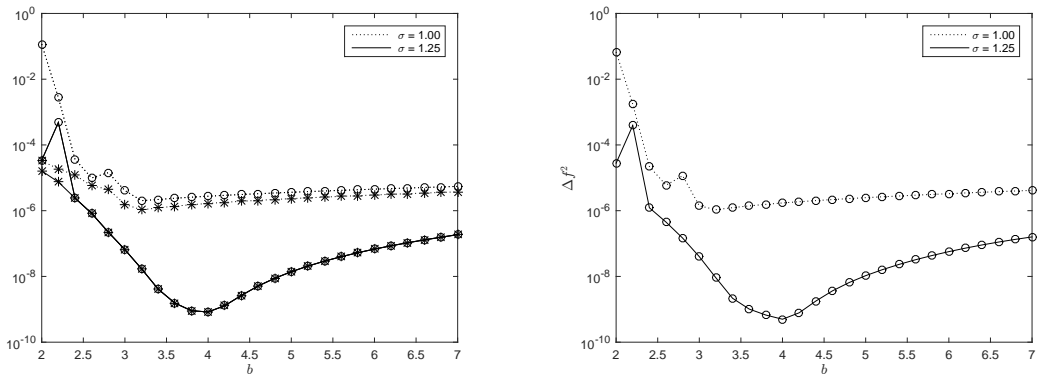


Figure 4.5: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Bessel window**. We used the set of Fourier coefficients (3.4), set  $m := 3$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

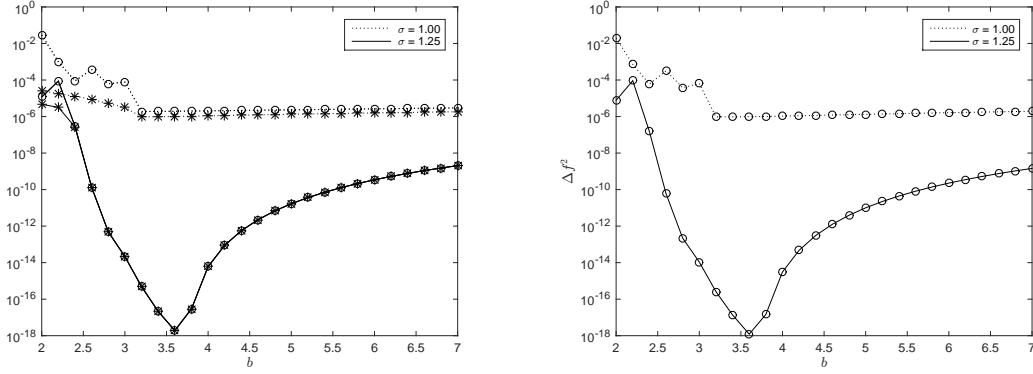


Figure 4.6: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Bessel window**. We used the set of Fourier coefficients (3.4), set  $m := 6$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

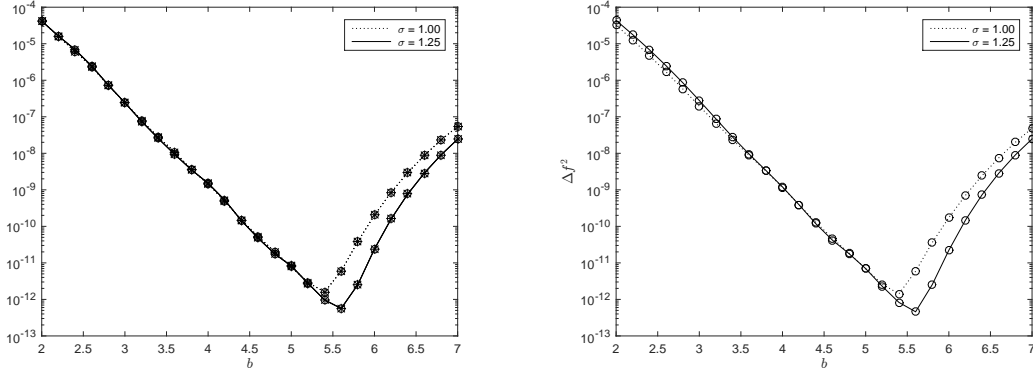


Figure 4.7: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Bessel window**. We used the set of Fourier coefficients (3.5), set  $m := 3$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

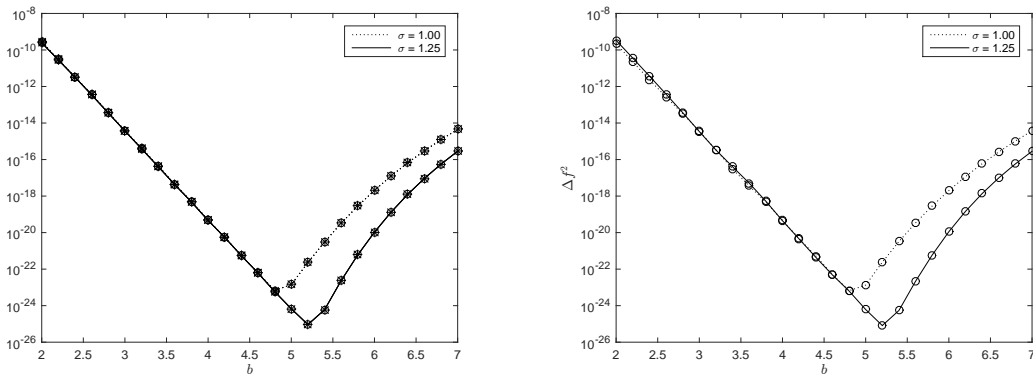


Figure 4.8: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Bessel window**. We used the set of Fourier coefficients (3.5), set  $m := 6$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ .

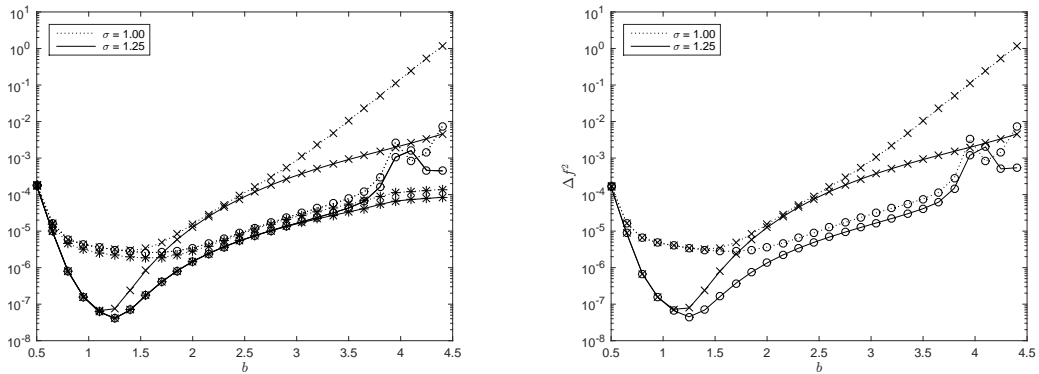


Figure 4.9: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Gaussian window**. We used the set of Fourier coefficients (3.4), set  $m := 3$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ . ( $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ : x,  $\hat{d}_k := c_k^{-1}(\tilde{\varphi}_t)$ : o)

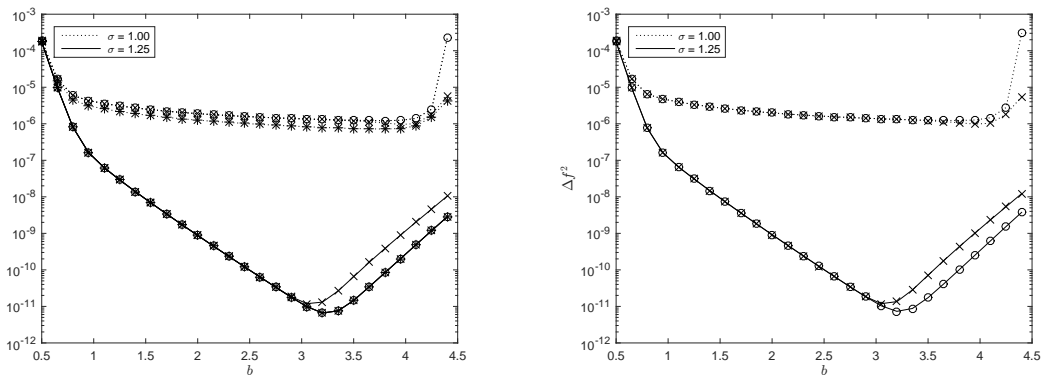


Figure 4.10: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Gaussian window**. We used the set of Fourier coefficients (3.4), set  $m := 6$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ . ( $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ : x,  $\hat{d}_k := c_k^{-1}(\tilde{\varphi}_t)$ : o)

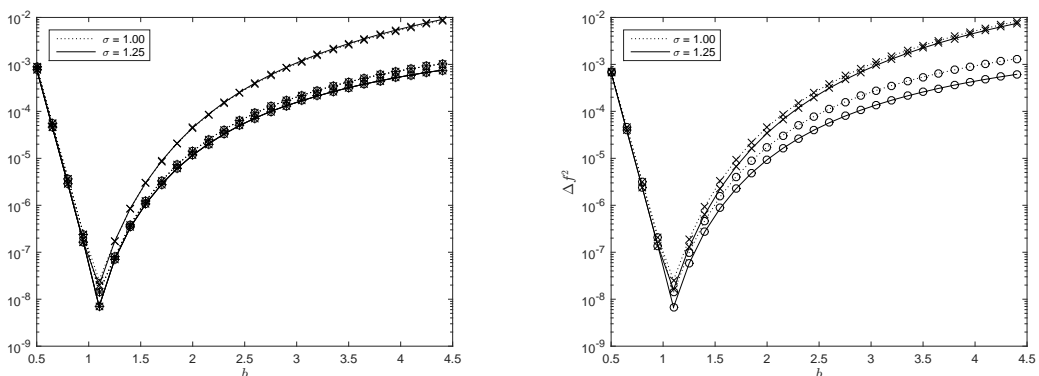


Figure 4.11: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Gaussian window**. We used the set of Fourier coefficients (3.5), set  $m := 3$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ . ( $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ : x,  $\hat{d}_k := c_k^{-1}(\tilde{\varphi}_t)$ : o)

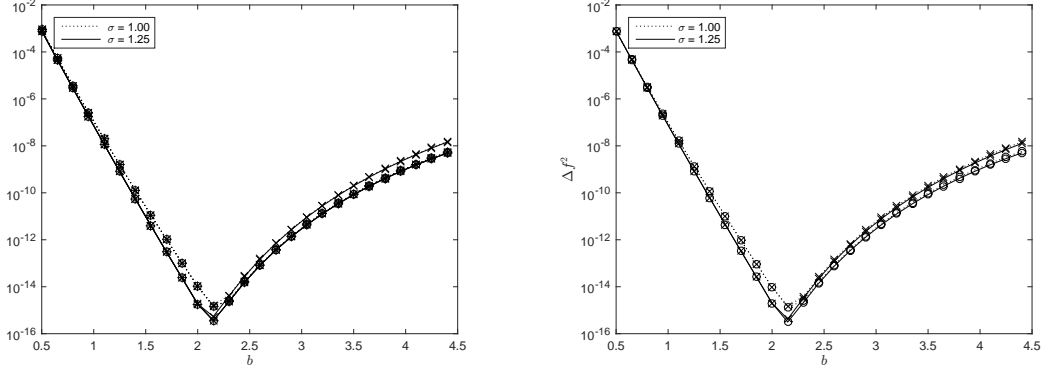


Figure 4.12: Estimated quadratic errors  $\|f - \tilde{f}\|_2^2$  (left) and the measured errors  $\Delta f^2$  (right) for the **Gaussian window**. We used the set of Fourier coefficients (3.5), set  $m := 6$  and plot the results for different oversampling factors  $\sigma$  with respect to the shape parameter  $b$ . ( $\hat{d}_k := c_k^{-1}(\tilde{\varphi})$ :  $\times$ ,  $\hat{d}_k := c_k^{-1}(\tilde{\varphi}_t)$ :  $\circ$ )

## 5 Parameter tuning

Based on the very accurate error estimates we are able to construct a tuning algorithm for the shape parameter  $b$ . In the following we denote by

$$\Delta f_{\approx}^2(\text{window}, b, m, \sigma) \begin{cases} \approx \|f - \tilde{f}\|_2^2 & \text{as described in Section 3} & : b > 0, \\ = +\infty & & : b \leq 0, \end{cases}$$

the predicted quadratic error, which can be computed for the different window functions as described in Section 3.

For the (modified) B-spline window it seems that a shape parameter  $b \in \frac{1}{2}\mathbb{N} \cap (m/2, m]$  is always optimal. Thus, we suggest to tune the shape parameter as follows.

**Algorithm 5.1** (Shape parameter tuning for B-splines).

Input: Fourier coefficients  $\hat{f}_k$ ,  $k \in \mathcal{I}_M$ , support parameter  $m$ , oversampling factor  $\sigma$ .

- i) Set  $b_{\text{opt}} := m$  and  $\Delta f_{\approx, \text{opt}}^2 := \Delta f_{\approx}^2(\text{B-spline}, b_{\text{opt}}, m, \sigma)$ .
- ii) Set  $b_{\text{next}} := m - 1/2$  and  $\Delta f_{\approx, \text{next}}^2 := \Delta f_{\approx}^2(\text{B-spline}, b_{\text{next}}, m, \sigma)$ .
- iii) While  $\Delta f_{\approx, \text{next}}^2 < \Delta f_{\approx, \text{opt}}^2$  and  $b_{\text{next}} > m/2$ :
  - a) Set  $b_{\text{opt}} := b_{\text{next}}$  and  $\Delta f_{\approx, \text{opt}}^2 := \Delta f_{\approx, \text{next}}^2$ .
  - b) Set  $b_{\text{next}} := b_{\text{opt}} - 1/2$  and  $\Delta f_{\approx, \text{next}}^2 := \Delta f_{\approx}^2(\text{B-spline}, b_{\text{next}}, m, \sigma)$ .

Output: Optimal shape parameter  $b_{\text{opt}}$  and predicted quadratic error  $\Delta f_{\approx, \text{opt}}^2$ . □

In order to tune the shape parameter  $b$  for the Bessel or the Gauss window function we suggest to use a simple search algorithm of the following form.

**Algorithm 5.2** (Shape parameter tuning for Bessel and Gauss window).

Input: Fourier coefficients  $\hat{f}_k$ ,  $k \in \mathcal{I}_M$ , support parameter  $m$ , oversampling factor  $\sigma$ , desired window function (Bessel or Gauss), start value  $b_{\text{start}}$ , start step size  $d_{\text{start}}$ .

- i) Set  $d := d_{\text{start}}$ .



- ii) Set  $b_{\text{opt}} := b_{\text{start}}$  and  $\Delta f_{\approx, \text{opt}}^2 := \Delta f_{\approx}^2(\text{window}, b_{\text{opt}}, m, \sigma)$ .
- iii) Set  $b_{\text{left}} := b_{\text{opt}} - d$  and  $\Delta f_{\approx, \text{left}}^2 := \Delta f_{\approx}^2(\text{window}, b_{\text{left}}, m, \sigma)$ .
- iv) Set  $b_{\text{right}} := b_{\text{opt}} + d$  and  $\Delta f_{\approx, \text{right}}^2 := \Delta f_{\approx}^2(\text{window}, b_{\text{right}}, m, \sigma)$ .
- v) Until  $\max\{\Delta f_{\approx, \text{left}}^2, \Delta f_{\approx, \text{opt}}^2, \Delta f_{\approx, \text{right}}^2\} \approx \min\{\Delta f_{\approx, \text{left}}^2, \Delta f_{\approx, \text{opt}}^2, \Delta f_{\approx, \text{right}}^2\}$ 
  - If  $\min\{\Delta f_{\approx, \text{left}}^2, \Delta f_{\approx, \text{opt}}^2, \Delta f_{\approx, \text{right}}^2\} = \Delta f_{\approx, \text{opt}}^2$ :
    - a) Set  $d := \gamma \cdot d$  for some  $\gamma \in (0, 1)$ , i.e., choose a smaller step size.
  - Else:
    - a)  $b_{\text{opt}} := \arg \min_{b_{\text{left}}, b_{\text{opt}}, b_{\text{right}}} \{\Delta f_{\approx, \text{left}}^2, \Delta f_{\approx, \text{opt}}^2, \Delta f_{\approx, \text{right}}^2\}$ .
    - b)  $\Delta f_{\approx, \text{opt}}^2 := \min\{\Delta f_{\approx, \text{left}}^2, \Delta f_{\approx, \text{opt}}^2, \Delta f_{\approx, \text{right}}^2\}$ .
  - Set  $b_{\text{left}} := b_{\text{opt}} - d$  and  $\Delta f_{\approx, \text{left}}^2 := \Delta f_{\approx}^2(\text{window}, b_{\text{left}}, m, \sigma)$ .
  - Set  $b_{\text{right}} := b_{\text{opt}} + d$  and  $\Delta f_{\approx, \text{right}}^2 := \Delta f_{\approx}^2(\text{window}, b_{\text{right}}, m, \sigma)$ .

Output: Optimal shape parameter  $b_{\text{opt}}$  and predicted quadratic error  $\Delta f_{\approx, \text{opt}}^2$ . □

**Example 5.1.** We use Algorithm 5.2 in order to tune the optimal shape parameters for different parameter settings. Thereby, we set  $b_{\text{start}} := b_0$ ,  $d_{\text{start}} := 1/2 \cdot b_0$  and  $\gamma := 1/2$ .

Note, that we have not yet proven if the stated algorithm always converges to the global minimum. However, in our examples the results coincide with the experimentally determined optimal values of  $b$ , which we plotted in Figure 3.10, see Tables 5.1 and 5.2. For the obtained approximation errors compare also to Figure 3.10. □

	$\sigma = 1$	$\sigma = 5/4$		$\sigma = 1$	$\sigma = 5/4$
$m = 2$	4.0743	5.0364	$m = 2$	5.5101	5.5776
$m = 3$	3.1416	4.0350	$m = 3$	5.3751	5.6015
$m = 4$	3.1539	3.8067	$m = 4$	5.2094	5.4597
$m = 5$	3.1907	3.7294	$m = 5$	5.0437	5.3622
$m = 6$	3.2398	3.7340	$m = 6$	4.8781	5.2462
$m = 7$	3.2398	3.6705	$m = 7$	4.7063	5.1358
$m = 8$	3.3379	3.6862	$m = 8$	4.5406	5.0216

Table 5.1: Tuned optimal shape parameters  $b_{\text{opt}}$  for the Fourier coefficients (3.4) on the left hand side and for the Fourier coefficients (3.5) on the right hand side.

Window function: Bessel. Compare to Figure 3.10 for the obtained errors.

Another task concerning the parameter tuning is to find out a parameter set for which the computational time is as minimal as possible, assumed that a certain accuracy has to be achieved. Of course, the optimal set of parameters regarding computation time may very much depend on the processor which is used for the calculations.

Given a required accuracy we could apply a tuning algorithm of the form 5.1 or 5.2 in order to determine for different values of the support parameter  $m$  a sufficiently large oversampling factor  $\sigma$  and an appropriate shape parameter  $b$ , for which the accuracy is achieved, respectively. The optimal parameter set can be determined by a comparison between measured

	$\sigma = 1$	$\sigma = 5/4$		$\sigma = 1$	$\sigma = 5/4$
$m = 2$	0.7759	0.7626	$m = 2$	0.7709	0.7647
$m = 3$	1.6114	1.2434	$m = 3$	1.1116	1.1004
$m = 4$	2.4669	1.8900	$m = 4$	1.4672	1.4382
$m = 5$	3.3323	2.5697	$m = 5$	1.8278	1.7822
$m = 6$	3.7600	3.2453	$m = 6$	2.1934	2.1324
$m = 7$	4.5956	3.8297	$m = 7$	2.5763	2.4878
$m = 8$	4.9338	4.4762	$m = 8$	2.9692	2.8474

Table 5.2: Tuned optimal shape parameters  $b_{\text{opt}}$  for the Fourier coefficients (3.4) on the left hand side and for the Fourier coefficients (3.5) on the right hand side.

Window function: Gauss. Compare to Figure 3.10 for the obtained errors.

computation times on the used computer. A corresponding tuning could be roughly of the following form.

**Algorithm 5.3** (Accuracy tuning).

Input: Fourier coefficients  $\hat{f}_k$ , set of window functions, list of support parameters  $m_1 \leq \dots \leq m_n$ , set of oversampling factors  $1 = \sigma_1 \leq \dots \leq \sigma_{\text{max}}$ , required accuracy  $\epsilon > 0$ .

- i) For each window function and for each support parameter  $m_j$ ,  $j = 1, \dots, n$ , set

$$\sigma_{\min}(m_j) := \min \{ \sigma \in \{ \sigma_1, \dots, \sigma_{\text{max}} \} : \Delta f_{\approx}(\text{window}, b_{\text{opt}}, m_j, \sigma) \leq \epsilon \},$$

if this minimum exists. Thereby, use Algorithm 5.1 or rather Algorithm 5.2 in order to tune the shape parameter  $b$  in each case.

- ii) For all obtained possible sets of parameters find the optimal one regarding runtime by running a simple test scenario.

□

**Example 5.2.** We consider the two sets of Fourier coefficients as given in (3.4) and (3.5). For a given required accuracy  $\epsilon$  we compare the tuned parameters for the Bessel and the B-spline window, see Tables 5.3 and 5.4. Thereby, we computed for each  $m$  the required oversampling factor by

$$\sigma_{\min}(m) := \min \{ \sigma \in \{ 1 + \frac{s}{16}, s = 1, \dots, 16 \} : \Delta f_{\approx}(\text{window}, b_{\text{opt}}, m, \sigma) \leq \epsilon \},$$

i.e., we set  $\sigma_{\text{max}} := 2$ .

The results show that for the Bessel window a smaller oversampling factor is needed compared to the B-spline window in order to obtain the given accuracy. Different combinations of the parameters  $m$  and  $\sigma$  are possible. Which one is the optimal with respect to runtime will depend on the used hardware. □

### The multivariate case

In the multivariate case the prediction of the error is somewhat more complicated. Since the window function is constructed based on a tensor product approach (2.3) the computation of

	Bessel			B-spline		
	$\sigma_{\min}$	$b_{\text{opt}}$	$\Delta f_{\approx}$	$\sigma_{\min}$	$b_{\text{opt}}$	$\Delta f_{\approx}$
$m = 4$	1.5000	4.24	9.62e-08	–	–	–
$m = 5$	1.1875	3.59	4.82e-08	1.6250	4.5	7.88e-08
$m = 6$	1.1250	3.45	1.51e-08	1.4375	5.0	3.77e-08
$m = 7$	1.0625	3.30	2.63e-08	1.3125	5.5	3.39e-08
$m = 8$	1.0625	3.23	1.92e-08	1.1875	5.5	9.21e-08

Table 5.3: Computed parameter sets and predicted errors for the Fourier coefficients (3.4). We set the required accuracy to  $\epsilon := 10^{-7}$ .

	Bessel			B-spline		
	$\sigma_{\min}$	$b_{\text{opt}}$	$\Delta f_{\approx}$	$\sigma_{\min}$	$b_{\text{opt}}$	$\Delta f_{\approx}$
$m = 4$	–	–	–	–	–	–
$m = 5$	1.0625	5.14	8.35e-11	1.4375	5.0	7.99e-11
$m = 6$	1.0000	4.88	2.21e-12	1.1250	6.0	6.90e-11
$m = 7$	1.0000	4.71	5.28e-14	1.0000	6.5	2.05e-11
$m = 8$	1.0000	4.54	1.87e-15	1.0000	7.5	5.19e-13

Table 5.4: Computed parameter sets and predicted errors for the Fourier coefficients (3.5). We set the required accuracy to  $\epsilon := 10^{-10}$ .

the error sums

$$\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \frac{c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}{c_{\mathbf{k}}^2(\tilde{\varphi}_t)} \quad \text{and} \quad \frac{\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}{\sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}$$

is more or less straight forward. We have

$$\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \frac{c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}{c_{\mathbf{k}}^2(\tilde{\varphi}_t)} = \prod_{j=1}^d \left( \sum_{r_j \in \mathbb{Z}} \frac{c_{k_j+r_j \sigma_j M_j}^2(\tilde{\varphi}_{1d,t})}{c_{k_j}^2(\tilde{\varphi}_{1d,t})} \right) - 1$$

and

$$\frac{\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}{\sum_{\mathbf{r} \in \mathbb{Z}^d} c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)} = 1 - \prod_{j=1}^d \left( \sum_{r_j \in \mathbb{Z}} \frac{c_{k_j}^2(\tilde{\varphi}_{1d,t})}{c_{k_j+r_j \sigma_j M_j}^2(\tilde{\varphi}_{1d,t})} \right),$$

which may be further transformed in order to allow a numerically stable evaluation. As an example, for  $d = 2$  we have

$$\sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \frac{c_{\mathbf{k}+\mathbf{r} \odot M_o}^2(\tilde{\varphi}_t)}{c_{\mathbf{k}}^2(\tilde{\varphi}_t)} = \prod_{j=1}^2 \left( \sum_{r_j \in \mathbb{Z} \setminus \{0\}} \frac{c_{k_j+r_j \sigma_j M_j}^2(\tilde{\varphi}_{1d,t})}{c_{k_j}^2(\tilde{\varphi}_{1d,t})} \right) - \sum_{j=1}^2 \sum_{r_j \in \mathbb{Z} \setminus \{0\}} \frac{c_{k_j+r_j \sigma_j M_j}^2(\tilde{\varphi}_{1d,t})}{c_{k_j}^2(\tilde{\varphi}_{1d,t})}.$$

Obviously, the evaluations of the estimates (2.7) and (2.8) are easy and especially possible in an efficient way, if the coefficients  $\hat{f}_{\mathbf{k}}$  are also of a tensor product structure, i.e., we have

$$\hat{f}_{\mathbf{k}} = \prod_{j=1}^d \hat{g}_{k_j}.$$

In this case, all necessary computations can be separated with respect to the  $d$  dimensions and an efficient tuning of the involved parameters can be realized quite similar to the univariate case. As an example, coefficients of the form  $e^{-\alpha\|\mathbf{k}\|^2} = \prod_{j=1}^d e^{-\alpha k_j^2}$  satisfy this condition.

However, in many cases the given Fourier coefficients are not of a tensor product structure or rather not even given analytically. In this case an approximation or estimation of the form

$$\hat{f}_{\mathbf{k}} \approx \prod_{j=1}^d \hat{g}_{k_j} \quad \text{or} \quad \hat{f}_{\mathbf{k}} \leq \prod_{j=1}^d \hat{g}_{k_j}$$

might be necessary in order to enable an efficient prediction of the occurrent errors.

As an example, if the Fourier coefficients are given by  $\hat{f}_{\mathbf{k}} := (1 + \|\mathbf{k}\|^2)^{-1}$  an approximation of the form

$$\frac{1}{x} \approx \sum_{j=1}^n r_j e^{-w_j x} \tag{5.1}$$

over a sufficiently large interval  $[1, \ell)$  gives

$$\frac{1}{1 + \|\mathbf{k}\|^2} \approx \sum_{j=1}^n r_j e^{-w_j} e^{-w_j \|\mathbf{k}\|^2} =: \sum_{j=1}^n \tilde{r}_j e^{-w_j \|\mathbf{k}\|^2},$$

where the single summands are now of a tensor product structure. One possible tool for computing an approximation of the form (5.1) is the Remez algorithm, see [10] for instance.

## 6 Summary

In this paper we revisited the error formulas for the well established NFFT algorithm. We showed how we can achieve a very precise prediction of the approximation errors measured in the  $\mathcal{L}_2$ -norm for different window functions. Thereby, we concentrated on the univariate case, where a straight and efficient evaluation of the correspondent error sums is possible.

In our numerical examples we compared two different deconvolution approaches and modified the shape of the considered window functions. The results show that only minimal improvements can be obtained by applying the  $\mathcal{L}_2$ -optimized deconvolution scheme. Additionally, the examples show that, especially in the case that the Fourier coefficients are subject to a certain decrease, an appropriate modification of the window's shape parameter can lead to significantly smaller approximation errors. For the well established B-spline window function we introduced a modified version, which also contains a shape parameter. In our examples we could achieve considerable improvements compared to the classical B-spline window in a setting where the Fourier coefficients decayed only moderately. However, a comparison between the different window functions showed that the Bessel window is in most cases the best choice.

For the univariate case we suggest an easy parameter tuning. Given a required accuracy, different combinations of the involved parameters are possible. Which set of parameters is optimal with respect to the computation time may depend on the used hardware. A corresponding tuning method could be based on the derived error estimates as well as on the mentioned shape parameter tuning.

The prediction of the approximation errors in the multivariate case holds some more difficulties. If the Fourier coefficients are of a tensor product structure, an efficient computation

of the error terms is more or less straight forward. But in many cases the Fourier coefficients do not have this property or are even not given in an analytical form, which makes it even hard to find an approximation or an estimation by a tensor product like expression. More detailed investigations concerning the multivariate case should be subject to future research.

However, there are some multivariate applications where the underlying Fourier coefficients are known. As one example we refer to the three dimensional periodic coulomb problem, where the electrostatic potentials and energies of a set of charges in the three dimensional space are of interest, see [6]. The well known P<sup>2</sup>NFFT algorithm [18] combines the adjoint NFFT and the NFFT to evaluate the Coulomb potentials and forces very efficiently. The underlying Fourier coefficients are of the form  $\sim \|\mathbf{k}\|^{-2}e^{-\alpha\|\mathbf{k}\|^2}$ . In order to develop an efficient parameter tuning, the Fourier coefficients could be approximated by a tensor product expression. We aim to describe this approach in a subsequent work, where we want to apply the derived error estimates in order to serve a comparison between different window functions as well as to develop a more precise tuning of the involved parameters for this particular application

## Acknowledgments

The author gratefully acknowledges support by the German Research Foundation (DFG), project PO 711/12-1.

## References

- [1] G. Beylkin: *On the fast Fourier transform of functions with singularities*. Appl. Comput. Harmon. Anal., 2:363 – 381, 1995.
- [2] C.K. Chui: *An Introduction to Wavelets*. Academic Press, Boston, 1992.
- [3] M. Deserno and C. Holm: *How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines*. J. Chem. Phys., 109:7678 – 7693, 1998.
- [4] A.J.W. Duijndam and M.A. Schonewille: *Nonuniform fast Fourier transform*. Geophysics, 64:539 – 551, 1999.
- [5] A. Dutt and V. Rokhlin: *Fast Fourier transforms for nonequispaced data*. SIAM J. Sci. Stat. Comput., 14:1368 – 1393, 1993.
- [6] P.P. Ewald: *Die Berechnung optischer und elektrostatischer Gitterpotentiale*. Ann. Phys., 369:253–287, 1921.
- [7] J.A. Fessler and B.P. Sutton: *Nonuniform fast Fourier transforms using min-max interpolation*. IEEE Trans. Signal Process., 51:560 – 574, 2003.
- [8] K. Fourmont: *Non equispaced fast Fourier transforms with applications to tomography*. J. Fourier Anal. Appl., 9:431 – 450, 2003.
- [9] L. Greengard and J.Y. Lee: *Accelerating the nonuniform fast Fourier transform*. SIAM Rev., 46:443 – 454, 2004.

- [10] W. Hackbusch: *Entwicklungen nach Exponentialsummen*. Techn. rep., Max Planck Institute for Mathematics in the Sciences, 2005. <http://www.mis.mpg.de/de/publications/andere-reihen/tr/report-0405.html>.
- [11] R.W. Hockney and J.W. Eastwood: *Computer simulation using particles*. Taylor & Francis, Inc., Bristol, PA, USA, 1988.
- [12] J.I. Jackson, C.H. Meyer, D.G. Nishimura, and A. Macovski: *Selection of a convolution function for Fourier inversion using gridding*. IEEE Trans. Med. Imag., 10:473 – 478, 1991.
- [13] M. Jacob: *Optimized least-square nonuniform Fast Fourier Transform*. IEEE Trans. Signal Process., 57:2165 – 2177, 2009.
- [14] J.F. Kaiser: *Digital filters*. In F.F. Kuo and J.F. Kaiser (eds.): *System analysis by digital computer*. Wiley, New York, 1966.
- [15] J. Keiner, S. Kunis, and D. Potts: *Using NFFT3 - a software library for various nonequispaced fast Fourier transforms*. ACM Trans. Math. Software, 36:Article 19, 1 – 30, 2009.
- [16] S. Kunis and S. Kunis: *The nonequispaced FFT on graphics processing units*. PAMM, Proc. Appl. Math. Mech., 12, 2012.
- [17] S. Kunis, D. Potts, and G. Steidl: *Fast Gauss transform with complex parameters using NFFTs*. J. Numer. Math., 14:295 – 303, 2006.
- [18] M. Pippig and D. Potts: *Parallel three-dimensional nonequispaced fast Fourier transforms and their application to particle simulation*. SIAM J. Sci. Comput., 35:C411 – C437, 2013.
- [19] G. Plonka and M. Tasche: *On the computation of periodic spline wavelets*. Appl. Comput. Harmon. Anal., 2:1 – 14, 1995.
- [20] D. Potts and G. Steidl: *Fast summation at nonequispaced knots by NFFTs*. SIAM J. Sci. Comput., 24:2013 – 2037, 2003.
- [21] D. Potts, G. Steidl, and M. Tasche: *Fast Fourier transforms for nonequispaced data: A tutorial*. In J.J. Benedetto and P.J.S.G. Ferreira (eds.): *Modern Sampling Theory: Mathematics and Applications*, pp. 247 – 270, Boston, MA, USA, 2001. Birkhäuser.
- [22] I.J. Schoenberg: *Cardinal interpolation and spline functions*. J. Approx. Theory, 2(2):167 – 206, 1969.
- [23] G. Steidl: *A note on fast Fourier transforms for nonequispaced grids*. Adv. Comput. Math., 9:337 – 353, 1998.
- [24] T. Volkmer: *OpenMP parallelization in the NFFT software library*. Preprint TU Chemnitz, Preprint 7, 2012. <http://www.tu-chemnitz.de/~potts/paper/openmpNFFT.pdf>.
- [25] A.F. Ware: *Fast approximate Fourier transforms for irregularly spaced data*. SIAM Rev., 40:838 – 856, 1998.
- [26] Z. Yang and M. Jacob: *Mean square optimal NUFFT approximation for efficient non-Cartesian MRI reconstruction*. J. Mag. Reson., (242):126–135, 2014.