Fast Adaptive Expansions in Local Trigonometric Bases

L. Borup [a] and M. Nielsen [b]

[a] *Department of Mathematics, Aalborg University, Fredrik Bajers Vej 7G, 9220 Aalborg East, Denmark*

[b] *Department of Mathematics, University of South Carolina, South Carolina 29208, USA*

Abstract

Smooth local trigonometric bases along with wavelet packet bases have provided a natural setup for the best basis algorithm of Coifman and Wickerhauser. We consider a best basis algorithm for local trigonometric bases with a cost function that only depends on the local behavior of the signal near the points where the local trigonometric functions overlap. The benefit is that the complexity of the algorithm is much lower than using a standard cost function in the best basis algorithm. We compare the performance of the algorithm to the best basis algorithm using an ℓ^1-norm as cost function on several test signals.

Key words: adaptive expansions, local trigonometric basis, best basis algorithm

Introduction

It is often of interest to extract the local time and frequency content of a given signal f. In mathematical terms this is done by testing f against a collection of analyzing functions $\{\phi_n\}$, i.e. one computes the inner products $\langle f, \phi_n \rangle$. The usefulness of such an analysis is enhanced by working with functions ϕ_n that are well localized in both time and frequency. For practical reasons one often choose to consider only collections $\{\phi_n\}$ that form an orthonormal basis of the space $L^2(\mathbb{R})$ (signals of finite energy). One problem with such an approach is that no single orthonormal basis will be good for all signals, i.e. for a given basis there is a signal for which the expansion of the signal has a large number of non-negligible coefficients. Better results can be obtained by an adaptive

Email addresses: `lasse@math.auc.dk`, `nielsen@math.sc.edu` (M. Nielsen).

approach, where one expands the signal into a family of orthonormal bases each with different time-frequency properties and then choose the one that fit the signal the "best" with the notion of best defined by some numerical quantity. One needs a very special structure of the family of orthonormal bases for this to work in the sense that there is a fast associated algorithm that will find this best basis. One setup where this is possible is the wavelet packet construction of Coifman, Meyer and Wickerhauser in [2] which lead to the best basis algorithm that selects a best wavelet packet basis according to an additive information cost function. Another setup where the best basis algorithm works is for the local trigonometric bases introduced by Coifman and Meyer in [3]. The local trigonometric functions are functions perfectly localized in time and have good frequency localization as well. The purpose of the present paper is to use the special structure of the local trigonometric bases to construct an information cost function that will allow a much faster implementation of the best basis algorithm for this type of bases. We will discuss the local trigonometric bases briefly in Section 1. In Section 2 we consider the best basis algorithm and the modifications possible using the special structure of the local trigonometric function. Section 2 also contains some numerical results from our implementation of the algorithms.

1 Local Cosine Bases

In this section we will review the basic steps in the construction of local cosine bases that will be needed later to understand the version of the best basis algorithm we want to consider. We refer to [1,4] for a more detailed discussion of local trigonometric bases.

Let $\mathcal{P} = \{I_k\}_{k\in\mathbb{Z}}$ be a disjoint partition of $\mathbb{R}$ with $I_k = [\alpha_k, \alpha_{k+1})$. To each interval I_k we assign a cutoff radius $\epsilon_k > 0$ at the left endpoint α_k. For technical reasons we require that $\epsilon_k + \epsilon_{k+1} < \alpha_{k+1} - \alpha_k$ for each $k \in \mathbb{Z}$. We take a ramp function r such that

$$r(t) = \begin{cases} 0 & \text{, if } t \leq -1 \\ 1 & \text{, if } t \geq 1, \end{cases}$$

with the property that $|r(t)|^2 + |r(-t)|^2 = 1$ for all $t \in \mathbb{R}$. The associated collection of (smooth) local cosine functions is given by

$$\psi_{n,k}(t) = \sqrt{\frac{2}{\alpha_{k+1} - \alpha_k}}\, r\left(\frac{t - \alpha_k}{\epsilon_k}\right) r\left(\frac{\alpha_{k+1} - t}{\epsilon_{k+1}}\right) \cos\left[\frac{\pi(n + \frac{1}{2}(t - \alpha_k)}{\alpha_{k+1} - \alpha_k}\right],$$

where $k \in \mathbb{Z}$ and $n \geq 0$. Notice that the requirement $\epsilon_k + \epsilon_{k+1} < \alpha_{k+1} - \alpha_k$ implies that any two functions $\psi_{n,k}$ and $\psi_{n',k'}$ only overlap if $|k - k'| \leq 1$. Furthermore, when $k = k' + 1$ the size of the overlap is at most $2\epsilon_k$. The smoothness of $\psi_{n,k}$ is the same as the smoothness of r and we can clearly find

$r \in C^s(\mathbb{R})$ for any value of s including $s = \infty$. The following is the main result on the functions $\psi_{n,k}$.

Theorem 1 ([3]) *The collection* $\{\psi_{n,k}\}_{n\geq 0,\, k\in\mathbb{Z}}$ *form an orthonormal basis for* $L^2(\mathbb{R})$.

To understand why the functions $\psi_{n,k}$ are well adapted for a direct and fast implementation we consider the following unitary operator on $L^2(\mathbb{R})$ given by

$$U_k f(t) = \begin{cases} \overline{r}\left(\frac{\alpha_k - t}{\epsilon_k}\right) f(t) - \overline{r}\left(\frac{t-\alpha_k}{\epsilon_k}\right) f(2\alpha_k - t), & \text{if } \alpha_k - \epsilon_k < t < \alpha_k, \\ r\left(\frac{t-\alpha_k}{\epsilon_k}\right) f(t) + r\left(\frac{\alpha_k - t}{\epsilon_k}\right) f(2\alpha_k - t), & \text{if } \alpha_k < t < \alpha_k + \epsilon_k, \\ f(t), & \text{otherwise.} \end{cases} \tag{1}$$

with its adjoint given by

$$U_k^* f(t) = \begin{cases} r\left(\frac{\alpha_k - t}{\epsilon_k}\right) f(t) + r\left(\frac{t-\alpha_k}{\epsilon_k}\right) f(2\alpha_k - t), & \text{if } \alpha_k - \epsilon_k < t < \alpha_k, \\ \overline{r}\left(\frac{t-\alpha_k}{\epsilon_k}\right) f(t) - \overline{r}\left(\frac{\alpha_k - t}{\epsilon_k}\right) f(2\alpha_k - t), & \text{if } \alpha_n < t < \alpha_k + \epsilon_k, \\ f(t), & \text{otherwise.} \end{cases} \tag{2}$$

For any $I_k = [\alpha_k, \alpha_{k+1}) \in \mathcal{P}$ we put $C_{n,k}(t) = \sqrt{\frac{2}{\alpha_{k+1}-\alpha_k}} C_n(\frac{t-\alpha_k}{\alpha_{k+1}-\alpha_k})$ with $C_n(t) = \cos[\pi(n+1/2)t)]$. Then $\psi_{n,k}(t) = U_k^* U_{k+1}^* \mathbf{1}_{I_k}(t) C_{n,k}(t)$ and we obtain the fact that $\langle f, \psi_{n,k}\rangle = \langle U_{k+1} U_k f, \mathbf{1}_{I_k} C_{n,k}\rangle$. Hence, to expand the signal f we just preprocess f with the folding operators U_k and then expand the result using the discrete cosine transform (DCT-IV using the notation of [4]).

2 Adaptive Expansion in Local Trigonometric Bases

Given a partition $\{I_k\}_{k\in\mathbb{Z}}$ of $\mathbb{R}$ we can expand a function into local trigonometric coefficients as described above. But one partition might give a better representation than another. A best basis algorithm is an adaptive way to find the best representation of a function among a certain finite collection of bases. The construction of local trigonometric bases makes it possible to construct a fast best basis search. If we associate to each interval I_k the projection operator $P_{I_k} : L^2(\mathbb{R}) \to \overline{\text{span}}\{\psi_{n,k} | n \geq 0\}$ given by $P_{I_k} = U_k^* U_{k+1}^* \mathbf{1}_{I_k} U_k U_{k+1}$, we have that $P_{I_k \cup I_{k+1}} = P_{I_k} + P_{I_{k+1}}$ (see [1]). Thus replacing I_k and I_{k+1} in the partition with one interval $I_k \cup I_{k+1}$ only changes the expansion coefficients involving $\{\psi_{n,k}\}_{n\geq 0}$ and $\{\psi_{n,k+1}\}_{n\geq 0}$. Given two adjacent intervals in a partition the best basis algorithm decides whether to keep these intervals or to merge them into one. The decision is made adaptively using a specified cost function, and the algorithm seeks a representation with the lowest "cost" value. We refer to [4, Chap. 8] for a more detailed description of best basis algorithms.

Here we will consider a best basis algorithm with a cost function that measures how fast the local trigonometric coefficients decay. More precisely, given a function $f \in L^2(\mathbb{R})$ and a partition $\{I_k\}_{k\in\mathbb{Z}}$ of $\mathbb{R}$, we want to measure the number $\sum_{n\geq 0} n^{2m}|\langle f, \psi_{n,k}\rangle|^2$, for some integer $m > 0$. In order to compare this quantity for two different subdivisions we need to "rescale" the coefficients, so the cost function used in our best basis is

$$\sum_{n\geq 0}\left(\frac{n}{|I_k|}\right)^{2m}|\langle f, \psi_{n,k}\rangle|^2.$$

In the following we show how to use this measure without actually calculating the coefficients $\{\langle f, \psi_{n,k}\rangle\}_{n\geq 0, k\in\mathbb{Z}}$.

As explained in Section 1 the local trigonometric coefficients of a function is constructed by first applying the folding operators at the edges of the chosen subintervals followed by a discrete cosine transform. Assume for a moment that we use the Fourier transform instead of the cosine transform. Then, since the Fourier transform has the property that $\langle f^{(m)}, e^{-i\pi n\cdot/|I|}\rangle = (-i\pi n/|I|)^m\langle f, e^{-i\pi n\cdot/|I|}\rangle$ for f sufficiently smooth, we can measure how fast the coefficients decay by looking at the derivatives of the folded version of f before applying the Fourier transform. Furthermore, since the folding operators acts locally we only need to calculate the derivatives locally.

When we use a discrete cosine transform we almost have the same situation. Suppose we have a function $f \in C^m \cap L^2(\mathbb{R})$, for some $m \in \mathbb{N}$, and a partition of $\mathbb{R}$ given by a set of points $\{\alpha_k\}_{k\in\mathbb{Z}} \subset \mathbb{R}$. Fix a number k and define $f_1 = \mathbf{1}_{I_{k-1}}U_{k-1}U_k f$ and $f_2 = \mathbf{1}_{I_k}U_k U_{k+1} f$. Then we have from Section 1 that

$$(A) \equiv \sum_{n\geq 0}\left(\frac{\pi(n+1/2)}{|I_{k-1}|}\right)^{2m}|\langle f, \psi_{n,k-1}\rangle|^2 = \|f_1^{(m)}\|^2_{L^2(I_{k-1})},$$
$$(B) \equiv \sum_{n\geq 0}\left(\frac{\pi(n+1/2)}{|I_k|}\right)^{2m}|\langle f, \psi_{n,k}\rangle|^2 = \|f_2^{(m)}\|^2_{L^2(I_k)},$$

with the derivatives given as one sided limits at the endpoints. Now consider a partition where I_{k-1} and I_k are merged into one interval, say $J_{k-1} = [\alpha_{k-1}, \alpha_{k+1})$. Then defining $f_0 = \mathbf{1}_{J_{k-1}}U_{k-1}U_{k+1}f$ we have

$$(C) \equiv \sum_{n\geq 0}\left(\frac{\pi(n+1/2)}{|J_{k-1}|}\right)^{2m}|\langle f, \psi^0_{n,k-1}\rangle|^2 = \|f_0^{(m)}\|^2_{L^2(J_{k-1})},$$

where $\psi^0_{n,k-1}(t) = U^*_{k-1}U^*_{k+1}\mathbf{1}_{J_{k-1}}(t)C^0_{n,k-1}(t)$ and $C^0_{n,k-1}(t) = \sqrt{\frac{2}{|J_{k-1}|}}C_n(\frac{t-\alpha_{k-1}}{|J_{k-1}|})$. According to our choice of cost function, we want to compare the sum $(A)+(B)$ with the value (C). This can now be done just by looking at the m'th derivative of folded versions of f. Furthermore, since U_k only changes f on the interval

$[\alpha_k - \epsilon_k, \alpha_k + \epsilon_k]$, we have

$$
\begin{aligned}
(A) + (B) - (C) \\
&= \|f_1^{(m)}\|^2_{L^2(I_{k-1})} + \|f_2^{(m)}\|^2_{L^2(I_k)} - \|f_0^{(m)}\|^2_{L^2(J_{k-1})} \\
&= \|f_1^{(m)}\|^2_{L^2([\alpha_k-\epsilon_k,\alpha_k))} + \|f_2^{(m)}\|^2_{L^2([\alpha_k,\alpha_k+\epsilon_k])} - \|f_0^{(m)}\|^2_{L^2([\alpha_k-\epsilon_k,\alpha_k+\epsilon_k])},
\end{aligned}
$$

Thus, just by comparing the derivatives of the folded version of f locally around α_k, we are able to decide whether to divide the interval J_{k-1} at α_k or not.

2.1 Implementation

Given a discrete signal on an interval I, we have implemented the above procedure (in Matlab V.5.2.0) in an algorithm consisting of the following steps

(1) choose a finest subdivisions of I given by a set of points $\{\alpha_k\}_{k\in\mathbb{Z}} \subset I$, and choose corresponding ϵ_k satisfying $\epsilon_k + \epsilon_{k+1} < \alpha_{k+1} - \alpha_k$. [1]
(2) for each k
  (a) calculate intermediate coefficients consisting of the original signal with the folding operator applied locally around α_k.
  (b) compare the ℓ^2-norm squared of the m'th discrete derivative of the original signal with the folded version locally around α_k, and choose whether to divide the interval or not at the given point.
(3) calculate the local trigonometric coefficients corresponding to the optimal subdivision.

Given a largest number of subintervals N_{max} then since the α_k's are equally spaced we can denote each by an integer between 1 and N_{max}. Thus a given basis will be represented by a vector of integers corresponding to the chosen partition, i.e. at most N_{max} coefficients. The calculated coefficients are given as a vector of the same length as the original signal, where coefficients corresponding to a given subinterval are located at this subinterval. The algorithms are available at `http://www.math.sc.edu/~nielsen/LTB`.

Since the only difference between calculating the local trigonometric coefficients corresponding to given N_{max} subintervals and this procedure is the local comparison of the derivatives, this best basis algorithm is almost as fast (see section 2.2). The difference in complexity depends on the steepness of the ramp function (the size of the ϵ_k's).

[1] In our implementation we choose to let the subintervals be of the same length and the ϵ_k's of the same value but any divisions can be used in general.

2.2 Examples

In order to test our algorithm we have compared its performance with the best basis algorithm implemented in the Matlab toolbox WaveLab 802 (available at `http://www-stat.stanford.edu/~wavelab`). Since this algorithm only works for signals of length $N \cdot 2^J$, $N, J \in \mathbb{N}$, we have restricted the test to such signals. Also, since this algorithm only works for dyadic subdivisions we let $N_{max} = 2^d$ for some $d \in \mathbb{N}$. But it should be noted, that our algorithm works for signals of arbitrary length and for any partition. Figure 1 shows three test signals.

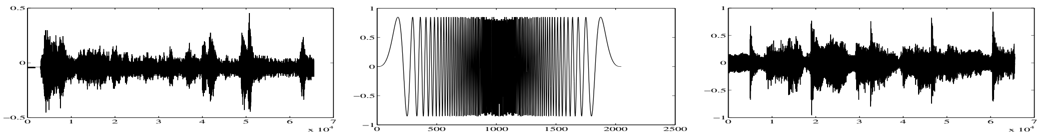

Fig. 1. The three test signals.

In figure 2-4 the results of the tests on the three signals in figure 1 are given. Each figure consists of six plots. The columns represents the result of using: N_{max} uniform distributed subdivisions (ULTB), our best basis algorithm (BEST1) or the best basis from WaveLab (BEST2). Each column consists of the thresholded local trigonometric coefficients corresponding to the chosen basis and the reconstructed signal from the thresholded coefficients. In the plot of the local trigonometric coefficients we have indicated where the signal has been divided. As a measure of the quality of the reconstructed signal we have calculated the signal to noise ratio (SNR) of the reconstructions. Since the signals have different length we have used different N_{max}. In all the tests we have used the second derivative ($m = 2$) in the calculations in BEST1 and the ℓ^1-norm as cost function in BEST2 (for these three examples the Shannon entropy function gave worse results). We use a hard thresholding in our tests i.e. a given percentage of the coefficients are set to zero. In the shown example this value is 90%.

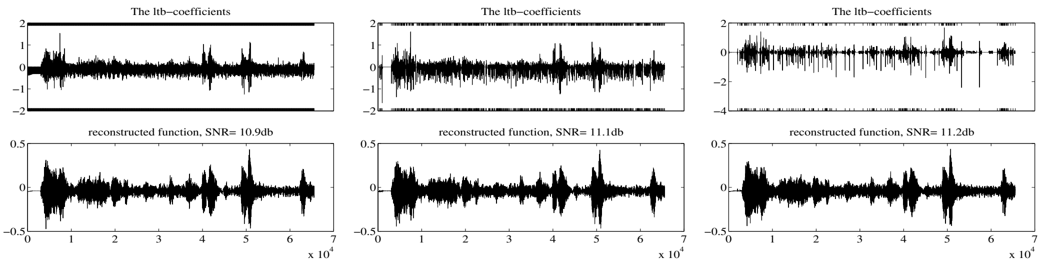

Fig. 2. Example 1.

In table 1 we have collected the data from the tests shown in figure 2-4. The SNR columns shows the signal to noise ratio after reconstruction of the signal from the remaining local trigonometric coefficients when 90% has been set

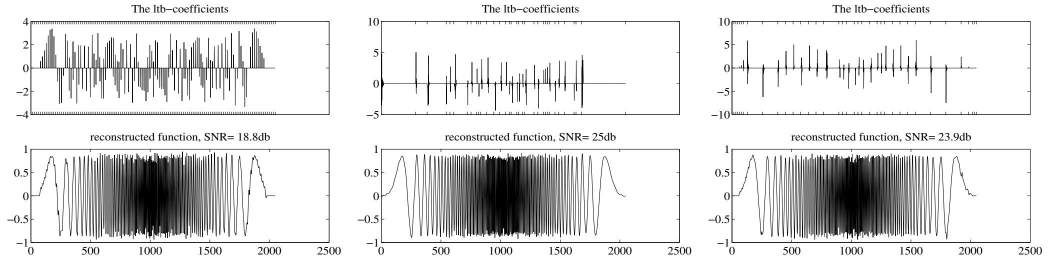

Fig. 3. Example 2.

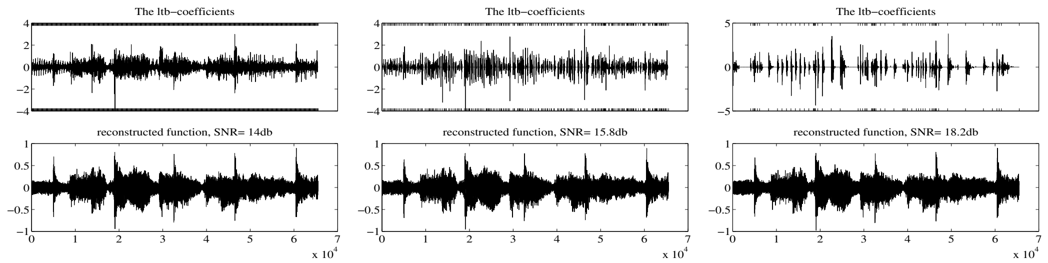

Fig. 4. Example 3.

to zero. The computation time (in seconds) was measured when running the algorithms on a Sun Ultra 5, using non compiled m-files.

Table 1

ex.	length	N_{max}	SNR			computation time		
			ULTB	BEST1	BEST2	ULTB	BEST1	BEST2
1	2^{16}	2^{10}	10.9	11.1	11.2	2.6	6.3	32.0
2	2^{11}	2^{7}	18.8	25.0	23.9	0.3	0.6	0.8
3	2^{16}	2^{9}	14.0	15.8	18.2	1.6	3.7	28.8

The first signal is a recording of some birds singing. Even though BEST2 is around five times slower than BEST1 it only gives a small improvement. The second example is given by $\sin(x^3)$ plus a mirrored version. This example has been included in order to show that our algorithm does what one should expect: divide the interval into finer subintervals as the frequency grows. In this example BEST1 does a better job than BEST2. The third example (part of “wonderful tonight” by Eric Clapton) shows that you can get better representations by using best basis algorithms. In this example BEST2 performs better than BEST1, but again one should compare this improvement with the complexity of the calculations. Here BEST1 is almost ten times faster than BEST2.

3 Conclusion

We have introduced a new type of cost function for the best basis algorithm on local trigonometric bases. The cost function uses the special structure of such bases to reduce the complexity of applying the algorithm considerably. It is not possible to prove mathematically that an adaptive analysis using the best basis algorithm with any given cost function will give better results than a corresponding non-adaptive expansion and in some cases the algorithm will indeed be fooled. Thus, the algorithms performance must be measured by testing it on a number of realistic signals. An implementation of our algorithm in Matlab confirms a considerable gain in complexity and computational time compared to the best basis algorithm with a standard cost function. Moreover, the tests in Section 2 show that the algorithm in most cases delivers results comparable to that of the best basis algorithm with standard cost functions such as the ℓ^1-norm or Shannon entropy. It should be noted that we observed the best performance of the algorithm on the non-harmonic parts of signals such as $\sin(x^3)$ signals. On the parts of the signal with pure frequencies or very fast oscillation the algorithm did not show any gain over a non-adaptive expansion, but this is not a big problem since the adaptive expansion is essentially as fast as the non-adaptive one.

The local nature of the local trigonometric functions and the low complexity of the new algorithm also makes it suited to "real time implementation" where one just has to set an upper limit on the interval length allowed in the local trigonometric basis. This limit should correspond to the buffer size of the device feeding the data to the algorithm. Finally, we note that our implementation of the algorithm is independent of both signal length and the length of the subintervals chosen at the finest resolution.

References

[1] Pascal Auscher, Guido Weiss, and M. Victor Wickerhauser. Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets. In *Wavelets*, pages 237–256. Academic Press, Boston, MA, 1992.

[2] R. R. Coifman, Y. Meyer, and V. Wickerhauser. Wavelet analysis and signal processing. In *Wavelets and their applications*, pages 153–178. Jones and Bartlett, Boston, MA, 1992.

[3] Ronald R. Coifman and Yves Meyer. Remarques sur l'analyse de Fourier à fenêtre. *C. R. Acad. Sci. Paris Sér. I Math.*, 312(3):259–261, 1991.

[4] Mladen Victor Wickerhauser. *Adapted wavelet analysis from theory to software.* A K Peters Ltd., Wellesley, MA, 1994.