Numerical stability of fast trigonometric and orthogonal wavelet transforms

Gerlind Plonka and Manfred Tasche

Abstract. Fast trigonometric transforms and periodic orthogonal wavelet transforms are essential tools for numerous practical applications. It is very important that fast algorithms work stable in a floating point arithmetic. This survey paper presents recent results on the worst case analysis of roundoff errors occurring in floating point computation of fast Fourier transforms, fast cosine transforms, and periodic orthogonal wavelet transforms. All these algorithms realize matrix-vector products with unitary matrices. The results are mainly based on a factorization of a unitary matrix into a product of sparse, almost unitary matrices. It is shown that under certain conditions fast trigonometric and periodic orthogonal wavelet transforms can be remarkably stable.

§1. Introduction

An algorithm for the discrete Fourier transform with low arithmetical complexity is called a fast Fourier transform (FFT). Fast algorithms of other discrete trigonometric transforms, such as a discrete cosine transform (DCT) or discrete sine transform, can be realized by FFT. These discrete trigonometric transforms are linear mappings generated by unitary or orthogonal matrices. Periodic orthogonal wavelet transforms are also linear mappings with orthogonal transform matrices. Nowadays, fast trigonometric transforms and periodic orthogonal wavelet transforms are essential tools in numerous practical computations. Therefore it is very important that the fast algorithms work stable in a floating point arithmetic with unit roundoff u. In this survey paper, it is shown that under certain conditions fast trigonometric transforms and periodic orthogonal wavelet transforms can be remarkably stable.

Advances in Constructive Approximation
XXX (eds.), pp. 1–3.

ISBN 0-9728482-x-x

Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a unitary matrix. For every input vector $\mathbf{x} \in \mathbb{C}^n$ let $\mathbf{y} := \mathbf{A}\mathbf{x}$ be the exact output vector. By $\hat{\mathbf{y}} \in \mathbb{C}^n$ we denote the computed vector of $\mathbf{A}\mathbf{x}$. Fast algorithms for the computation of $\mathbf{A}\mathbf{x}$ are based on a factorization of $\mathbf{A} = \mathbf{A}^{(t)} \ldots \mathbf{A}^{(1)}$ into a product of sparse, almost unitary matrices $\mathbf{A}^{(s)}$ and its stepwise realization by

$$\hat{\mathbf{y}} := \mathrm{fl}(\hat{\mathbf{A}}^{(t)}\mathrm{fl}(\hat{\mathbf{A}}^{(t-1)} \ldots \mathrm{fl}(\hat{\mathbf{A}}^{(2)}\mathrm{fl}(\hat{\mathbf{A}}^{(1)}\mathbf{x})) \ldots)),$$

where $\hat{\mathbf{A}}^{(s)}$ consists of precomputed entries of $\mathbf{A}^{(s)}$ and where $\mathrm{fl}(\hat{\mathbf{A}}^{(s)}\mathbf{z})$ denotes the vector $\hat{\mathbf{A}}^{(s)}\mathbf{z}$ computed in floating point arithmetic.

Let $\Delta\mathbf{y} := \hat{\mathbf{y}} - \mathbf{y}$. An algorithm for computing $\mathbf{A}\mathbf{x}$ is called *normwise forward stable* (see [8], p. 142), if there exist a constant $k_n > 0$ such that for all $\mathbf{x} \in \mathbb{C}^n$

$$\|\Delta\mathbf{y}\|_2 \le (k_n\, u + \mathcal{O}(u^2))\, \|\mathbf{x}\|_2$$

and $k_n u \ll 1$. Here $\|\mathbf{x}\|_2$ denotes the Euclidean norm of $\mathbf{x} \in \mathbb{C}^n$. We introduce $\Delta\mathbf{x} := \mathbf{A}^{-1}(\hat{\mathbf{y}} - \mathbf{y})$. Since $\mathbf{A}$ is unitary and since the Euclidean norm is unitary invariant, we also have *normwise backward stability* by

$$\|\Delta\mathbf{x}\|_2 \le (k_n\, u + \mathcal{O}(u^2))\, \|\mathbf{x}\|_2.$$

With other words, we measure the numerical stability of an algorithm for computing $\mathbf{A}\mathbf{x}$ by a *worst case stability constant* k_n. In this worst case analysis one can only obtain upper bounds of the roundoff errors. However, the worst case bounds already reflect the structure of roundoff errors and their dependence on different ingredients of an algorithm as e.g. the precomputation of matrix entries, the kind of matrix factorization, the choice of recursive or cascade summation etc.. This has also been confirmed by a series of numerical experiments (see e.g. [2, 3, 14, 15, 23, 24]).

More realistic estimates of the roundoff errors can be obtained in an average case study. Here it is assumed that all components of an input vector $\mathbf{x}$ and the resulting roundoff errors are random variables. One is interested in the distribution of the error vector $\Delta\mathbf{x}$. Then one can measure the average case backward stability in terms of the expected values

$$\mathrm{E}(\|\Delta\mathbf{x}\|_2^2) = (\overline{k}_n^2\, u^2 + \mathcal{O}(u^3))\, \mathrm{E}(\|\mathbf{x}\|_2^2)$$

with an average case stability constant $\overline{k}_n > 0$ such that $\overline{k}_n\, u \ll 1$. For details see [4, 22, 24, 27].

This survey paper is organized as follows: In Section 2, we introduce Wilkinson's model for the worst case study of roundoff errors. Further we estimate the roundoff errors of matrix-vector products. The key point of this paper is Section 3 which is devoted to fast matrix-vector multiplications. We assume that a unitary matrix $\mathbf{A}$ can be factorized into a product

of sparse, almost unitary matrices. Using this factorization, we can compute step by step the product $\mathbf{Ax}$ with arbitrary $\mathbf{x} \in \mathbb{C}^n$. In Theorems 5 and 6, we present a unified approach to worst case stability constants k_n. This roundoff error analysis is then applied to the FFT in Section 4, to fast cosine transforms in Section 5, and to periodic orthogonal wavelet transforms in Section 6. Here Theorems 7 and 8 on the factorization of the wavelet matrix and the polyphase matrix are new.

In this paper it is shown that the numerical stability of orthogonal transforms can be very different. In particular, *errors in precomputed entries* of a matrix (or a matrix factor) have a strong influence on the numerical stability of the algorithm. Further, the factorization of the transform matrix *should preserve the orthogonality*. Sparsity of the factor matrices means often that each row and column contains at most 2 nonzero entries. (This can always be obtained for unitary transform matrices.) Finally, we note that numerical stability and arithmetical complexity are rather independent properties of an algorithm, i.e., an algorithm with low arithmetical complexity can possess a bad numerical stability, and an algorithm with large arithmetical complexity can possess a good numerical stability.

§2. Matrix-vector products in floating point arithmetic

We consider a floating point number system $F \subset \mathbb{R}$ which is characterized by the following integer parameters: the base β, the precision t, and the exponent range $e_{\min} \le e \le e_{\max}$. The elements of F can be expressed in the form

$$y = \pm\beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_t}{\beta^t} \right),$$

where each digit d_i satisfies $0 \le d_i \le \beta - 1$. One can ensure a unique representation of each element $y \in F \setminus \{0\}$ by assuming that the most significant digit d_1 is not equal to zero. The system is then called *normalized*. The *range* of nonzero floating point numbers in a normalized system F is given by

$$\beta^{e_{\min}-1} \le |y| \le \beta^{e_{\max}}(1 - \beta^{-t}).$$

Then each real number $x \neq 0$ in the range of F can be approximated by a number $\mathrm{fl}(x) \in F$ with a relative error smaller than $u := \frac{1}{2}\beta^{1-t}$, i.e.,

$$\left| \frac{x - \mathrm{fl}(x)}{x} \right| < u.$$

Here, the constant u is called *unit roundoff* of the floating point system F. Note that the elements of F are not equally spaced.

The IEEE arithmetic in double precision uses a floating point number system with the parameters

$$\beta = 2, \quad t = 53, \quad e_{\min} = -1021, \quad e_{\max} = 1024.$$

The corresponding unit roundoff is $u = 2^{-53} \approx 1.11 \times 10^{-16}$. For more information on floating point number systems and standards we refer to [8] and references therein.

In order to carry out a rounding error analysis of an algorithm, we assume that the following *standard model of floating point arithmetic* by Wilkinson [25] is true: For arbitrary real numbers x, y and any basic arithmetical operation $\circ \in \{+, -, \times, /\}$, the exact value $x \circ y$ and the computed value $\mathrm{fl}(x \circ y)$ are related by

$$\mathrm{fl}(x \circ y) = (x \circ y)(1 + \epsilon^{\circ}) \qquad (|\epsilon^{\circ}| \le u). \tag{1}$$

This model is valid for most computers, in particular it holds for IEEE arithmetic.

We are especially interested in a roundoff error analysis for matrix-vector products, where the matrix is unitary (or orthogonal). At first, we consider inner products. With the unit roundoff u let now

$$\gamma_n := \frac{nu}{1 - nu} \qquad (n \in \mathbb{N},\ nu < 1).$$

Further, for vectors $\mathbf{a} \in \mathbb{R}^n$ and matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ let $|\mathbf{a}| := (|a_j|)_{j=0}^{n-1}$ and $|\mathbf{A}| := (|a_{jk}|)_{j,k=0}^{n-1}$ be the corresponding vectors and matrices of absolute values. Then we have

Lemma 1. *Let $n \in \mathbb{N}$ be given with $nu < 1$. Then for a recursive computation of the inner product for arbitrary vectors $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^n$ it follows*

$$|\mathbf{a}^T\mathbf{b} - \mathrm{fl}(\mathbf{a}^T\mathbf{b})| \le \gamma_n |\mathbf{a}|^T|\mathbf{b}| = (nu + \mathcal{O}(u^2))|\mathbf{a}|^T|\mathbf{b}|.$$

For cascade summation of the inner product it follows

$$|\mathbf{a}^T\mathbf{b} - \mathrm{fl}(\mathbf{a}^T\mathbf{b})| \le \gamma_{\lceil \log_2 n \rceil + 1} |\mathbf{a}|^T|\mathbf{b}| = ((\lceil \log_2 n \rceil + 1)\, u + \mathcal{O}(u^2))|\mathbf{a}|^T|\mathbf{b}|,$$

where for $a \in \mathbb{R}$, $\lceil a \rceil := \min\{m \in \mathbb{Z} : m \ge a\}$ is the smallest integer greater than or equal to a.

The proof follows immediately by induction over n (see e.g. [8], p. 69).

If the vector $\mathbf{a} \in \mathbb{R}^n$ possesses at most $l \le n$ nonzero entries, then we obtain as a trivial consequence of Lemma 1 that for arbitrary $\mathbf{b} \in \mathbb{R}^n$

$$|\mathbf{a}^T\mathbf{b} - \mathrm{fl}(\mathbf{a}^T\mathbf{b})| \le \begin{cases} \gamma_l\, |\mathbf{a}|^T|\mathbf{b}| & \text{for recursive summation,} \\ \gamma_{\lceil \log_2 l \rceil + 1}\, |\mathbf{a}|^T|\mathbf{b}| & \text{for cascade summation.} \end{cases}$$

Now we want to consider matrix-vector products. For a matrix $\mathbf{A} = (a_{jk})_{j,k=0}^{n-1} \in \mathbb{R}^{n\times n}$ let $\operatorname{sign}\mathbf{A} := (\operatorname{sign} a_{jk})_{j,k=0}^{n-1}$ be the corresponding sign-matrix, where for $a \in \mathbb{R}$,

$$\operatorname{sign} a := \begin{cases} 1 & \text{for } a > 0, \\ -1 & \text{for } a < 0, \\ 0 & \text{for } a = 0. \end{cases}$$

Further, for two vectors $\mathbf{a} = (a_j)_{j=0}^{n-1}$, $\mathbf{b} = (b_j)_{j=0}^{n-1} \in \mathbb{R}^n$ let $\mathbf{a} \le \mathbf{b}$ be defined by $a_j \le b_j$ for all $j = 0, \ldots, n-1$. Analogously, we write $\mathbf{A} \le \mathbf{B}$ for two matrices $\mathbf{A}$, $\mathbf{B}$ of same size if this inequality is true elementwise. Then we obtain

Theorem 2. *Let $n \in \mathbb{N}$, $n \ge 2$ and $2 \le l \le n$ with $lu < 1$ be given. Let $\mathbf{A} = (a_{jk})_{j,k=0}^{n-1} \in \mathbb{R}^{n\times n}$ be a matrix containing at most l nonzero entries in each row. Further, assume that the nonzero entries a_{jk} are precomputed by $\hat{a}_{jk}$, where*

$$|\hat{a}_{jk} - a_{jk}| \le \eta u \tag{2}$$

with some constant $\eta > 0$, and set $\hat{a}_{jk} = 0$ for $a_{jk} = 0$. Let $\hat{\mathbf{A}} := (\hat{a}_{jk})_{j,k=0}^{n-1}$. Then for arbitrary $\mathbf{x} \in \mathbb{R}^n$ the error $\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}$ satisfies the estimate

$$\begin{aligned} |\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}| &\le \gamma_{\tilde{l}}\,|\mathbf{A}||\mathbf{x}| + (\eta\,u + \gamma_{\tilde{l}}\,\eta\,u)|\operatorname{sign}\mathbf{A}||\mathbf{x}| \\ &= (\tilde{l}\,u + \mathcal{O}(u^2))|\mathbf{A}||\mathbf{x}| + (\eta u + \mathcal{O}(u^2))|\operatorname{sign}\mathbf{A}||\mathbf{x}|, \end{aligned}$$

where $\tilde{l} := l$ for recursive summation and $\tilde{l} := \lceil \log_2 l \rceil + 1$ for cascade summation.

Proof: The assumption (2) implies that

$$|\hat{\mathbf{A}} - \mathbf{A}| \le \eta u\,|\operatorname{sign}\mathbf{A}|.$$

Hence the error vector $\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}$ can be estimated as follows

$$\begin{aligned} |\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}| &\le |\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \hat{\mathbf{A}}\mathbf{x}| + |(\hat{\mathbf{A}} - \mathbf{A})\mathbf{x}| \\ &\le |\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \hat{\mathbf{A}}\mathbf{x}| + \eta\,u|\operatorname{sign}\mathbf{A}|\,|\mathbf{x}|. \end{aligned}$$

For the first term we obtain by Lemma 1

$$\begin{aligned} |\operatorname{fl}(\hat{\mathbf{A}}\mathbf{x}) - \hat{\mathbf{A}}\mathbf{x}| &\le \gamma_{\tilde{l}}\,|\hat{\mathbf{A}}|\,|\mathbf{x}| \\ &\le \gamma_{\tilde{l}}\,|\mathbf{A}|\,|\mathbf{x}| + \gamma_{\tilde{l}}\,|\hat{\mathbf{A}} - \mathbf{A}|\,|\mathbf{x}| \\ &\le \gamma_{\tilde{l}}\,|\mathbf{A}|\,|\mathbf{x}| + \gamma_{\tilde{l}}\,\eta\,u\,|\operatorname{sign}\mathbf{A}|\,|\mathbf{x}|, \end{aligned}$$

where we have used that each entry contains at most l nonzero entries. □

Using the spectral norm of the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, given by

$$||\mathbf{A}||_2 = \sqrt{\rho(\mathbf{A}^T\mathbf{A})},$$

where $\rho(\mathbf{A}^T\mathbf{A})$ denotes the spectral radius of $\mathbf{A}^T\mathbf{A}$, we finally obtain an error estimate in the Euclidean norm

$$\|\mathrm{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}\|_2 \le \gamma_{\tilde{l}}\,\|(|\mathbf{A}|)\|_2\|\mathbf{x}\|_2 + (1+\gamma_{\tilde{l}})\,\eta\,u\,\|(|\mathrm{sign}\,\mathbf{A}|)\|_2\,\|\mathbf{x}\|_2, \quad (3)$$

and for the relative forward error

$$\begin{aligned} E^f_{rel}(\mathbf{x}) &:= \frac{\|\mathrm{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}\|_2}{\|\mathbf{A}\mathbf{x}\|_2} \\ &\le \|\mathbf{A}^{-1}\|_2 \left(\gamma_{\tilde{l}}\,\|(|\mathbf{A}|)\|_2 + ((1+\gamma_{\tilde{l}})\,\eta u\,\|(|\mathrm{sign}\mathbf{A}|)\|_2\right). \end{aligned}$$

This error estimate can be further simplified if $\mathbf{A}$ is an orthogonal, sparse $(n\times n)$-matrix. Let us assume that the orthogonal matrix $\mathbf{A}$ possesses at most l entries in each row and in each column. Then we have

$$\|(|\mathrm{sign}\,\mathbf{A}|)\|_2 \le (\|\mathrm{sign}\,\mathbf{A}\|_\infty\,\|\mathrm{sign}\,\mathbf{A}\|_1)^{1/2} = l,$$

where $\|\cdot\|_\infty$ and $\|\cdot\|_1$ denote the row sum matrix norm and the column sum matrix norm of $\mathbf{A}$, respectively. Since the entries of an orthogonal matrix have an absolute value smaller than or equal to 1, and in each row (and each column) the sum of the squared entries is equal to 1 we obtain by

$$(\sum_{j=1}^{l} |a_j|)^2 \le l\,(\sum_{j=1}^{l} a_j^2)$$

that

$$\|(|\mathbf{A}|)\|_2 \le (\|\mathbf{A}\|_\infty\,\|\mathbf{A}\|_1)^{1/2} = \sqrt{l},$$

and by $\|\mathbf{A}^{-1}\|_2 = 1$ we finally obtain in this case

$$E^f_{rel}(\mathbf{x}) \le \gamma_{\tilde{l}}\,\sqrt{l} + (1+\gamma_{\tilde{l}})\,\eta\,u\,l = (\tilde{l}\,\sqrt{l} + \eta\,l)(u + \mathcal{O}(u^2)).$$

In the special case $l = 2$ one can even obtain

$$E^f_{rel}(\mathbf{x}) \le \left(\frac{4}{\sqrt{3}} + \sqrt{2}\,\eta\right) u + \mathcal{O}(u^2)$$

(see e.g. [12]). Observe that the estimates for the relative forward error always consist of two relevant terms, namely the first term depending on $|\mathbf{A}|$ and on the kind of summation and the second term depending on $|\mathrm{sign}\,\mathbf{A}|$ and on the precomputation of matrix entries.

This theory can be extended to complex matrix-vector products. Since complex arithmetic is implemented using real arithmetic, the following bounds can be derived for basic complex floating point operations.

Lemma 3. *Let z, $w \in \mathbb{C}$. Then we have*

$$\begin{aligned} \mathrm{fl}(z+w) &= (z+w)(1+\epsilon^+) \qquad (|\epsilon^+| \le u), \\ \mathrm{fl}(z \times w) &= (z \times w)(1+\epsilon^\times) \qquad (|\epsilon^\times| \le \mu_{\mathbf{C}} u + \mathcal{O}(u^2)), \end{aligned}$$

where $\mu_{\mathbf{C}} = \frac{4\sqrt{3}}{3} \approx 2.31$ is the best possible constant under Wilkinson's model (1). For $z \in \mathbb{R} \cup i\mathbb{R}$ and $w \in \mathbb{C}$ we even have

$$\mathrm{fl}(z \times w) = (z \times w)(1+\epsilon^\times) \qquad (|\epsilon^\times| \le u).$$

For a proof we refer to [22], Lemma 8.1 or [23]. Other proofs can be found in [8], p. 79 with a constant $\mu_{\mathbf{C}} = 2\sqrt{2}$ and in [5] with $\mu_{\mathbf{C}} = 1+\sqrt{2}$.

By a suitable modification of Lemma 1 and Theorem 2 one obtains in complex arithmetic the following

Corollary 4. *Let $\mathbf{A} = (a_{jk})_{j,k=0}^{n-1} \in \mathbb{C}^{n\times n}$ and $\mathbf{x} \in \mathbb{C}^n$. Assume that each row of $\mathbf{A}$ contains at most l nonzero entries and that the precomputed values $\hat{a}_{jk}$ satisfy, for $a_{jk} \notin \{0, \pm 1, \pm i\}$,*

$$|\hat{a}_{jk} - a_{jk}| \le \eta\, u$$

with $\eta > 0$, and $\hat{a}_{jk} = a_{jk}$ otherwise. Here i denotes the imaginary unity. Let $\hat{\mathbf{A}} := (\hat{a}_{jk})_{j,k=0}^{n-1}$. Further, let $\tilde{\mathbf{A}} := (\tilde{a}_{jk})_{j,k=0}^{n-1}$ with

$$\tilde{a}_{jk} := \begin{cases} 1 & \text{if } a_{jk} \notin \{0, \pm 1, \pm i\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then we have

$$|\mathrm{fl}(\hat{\mathbf{A}}\mathbf{x}) - \mathbf{A}\mathbf{x}| \le ((\tilde{l} + \mu_{\mathbf{C}} - 1)u + \mathcal{O}(u^2))|\mathbf{A}|\,|\mathbf{x}| + \eta\tilde{\mathbf{A}}\,|\mathbf{x}|(u + \mathcal{O}(u^2)),$$

where $\tilde{l} := l$ for recursive summation and $\tilde{l} := \lceil \log_2 l \rceil + 1$ for cascade summation.

A proof of this assertion can be found in [22], Lemma 8.4.

§3. Fast matrix-vector multiplications

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an orthogonal matrix. Assume that $\mathbf{A}$ possesses a factorization into a product of sparse, almost orthogonal matrices

$$\mathbf{A} = \mathbf{A}^{(t)}\,\mathbf{A}^{(t-1)} \ldots \mathbf{A}^{(2)}\,\mathbf{A}^{(1)}. \tag{4}$$

A matrix $\mathbf{A}^{(s)} \in \mathbb{R}^{n\times n}$ is called *almost orthogonal*, if $(\mathbf{A}^{(s)})^T\mathbf{A}^{(s)} = \alpha_s^2\,\mathbf{I}_n$ with the identity matrix $\mathbf{I}_n$ and some constant $\alpha_s = \alpha(\mathbf{A}^{(s)}) > 0$. Now, the matrix-vector product $\mathbf{A}\mathbf{x}$ can be computed by starting with $\mathbf{y}^{(0)} := \mathbf{x}$ and by recursive evaluation of $\mathbf{y}^{(s)} := \mathbf{A}^{(s)}\mathbf{y}^{(s-1)}$ for $s = 1, \ldots, t$. Most fast algorithms for a matrix-vector product $\mathbf{A}\mathbf{x}$ are based on a factorization of the matrix $\mathbf{A}$. Considering now the numerical stability of such algorithms, we obtain

Theorem 5. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an orthogonal matrix with a factorization (4), where $\mathbf{A}^{(s)} = (a_{jk}^{(s)})_{j,k=0}^{n-1} \in \mathbb{R}^{n\times n}$, $s = 1, \ldots, t$ are almost orthogonal. Assume that in each row and in each column of $\mathbf{A}^{(s)}$ are at most $l_s = l(\mathbf{A}^{(s)})$ nonzero entries which are precomputed with*

$$|\hat{a}_{jk}^{(s)} - a_{jk}^{(s)}| \le \eta_s\, u \qquad (\eta_s = \eta(\mathbf{A}^{(s)}) > 0),$$

and let $\hat{a}_{jk}^{(s)} = 0$ for $a_{jk}^{(s)} = 0$. Further let $\hat{\mathbf{A}}^{(s)} = (\hat{a}_{jk}^{(s)})_{j,k=0}^{n-1} \in \mathbb{R}^{n\times n}$ for $s = 1, \ldots, t$. Then the forward error vector for sequential summation

$$\Delta\mathbf{y} := \mathrm{fl}(\hat{\mathbf{A}}^{(t)}\mathrm{fl}(\hat{\mathbf{A}}^{(t-1)} \ldots \mathrm{fl}(\hat{\mathbf{A}}^{(2)}\mathrm{fl}(\hat{\mathbf{A}}^{(1)}\mathbf{x})) \ldots)) - \mathbf{A}\mathbf{x}$$

satisfies

$$\|\Delta\mathbf{y}\|_2 \le (k_n\, u + \mathcal{O}(u^2))\, \|\mathbf{x}\|_2$$

with

$$k_n := \sum_{s=1}^{t} (l_s^{3/2} + \frac{\eta_s}{\alpha_s} l_s).$$

Proof: Let $\hat{\mathbf{y}}^{(0)} := \mathbf{x}$ and $\hat{\mathbf{y}}^{(s)} := \mathrm{fl}(\hat{\mathbf{A}}^{(s)}\, \hat{\mathbf{y}}^{(s-1)})$ for $s = 1, \ldots, t$, the intermediate vectors obtained by the algorithm in floating point arithmetic. Further, for $s = 1, \ldots, t$, let $\mathbf{e}^{(s)} := \mathrm{fl}(\hat{\mathbf{A}}^{(s)}\hat{\mathbf{y}}^{(s-1)}) - \mathbf{A}^{(s)}\hat{\mathbf{y}}^{(s-1)}$ denote the error vector in step s. Then from (3) it follows that

$$\|\mathbf{e}^{(s)}\|_2 \le \Big(\gamma_{l_s} \|(|\mathbf{A}^{(s)}|)\|_2 + (1+\gamma_{l_s})\, \eta_s\, u \|(|\mathrm{sign}\, \mathbf{A}^{(s)}|)\|_2\Big)\, \|\hat{\mathbf{y}}^{(s-1)}\|_2.$$

In view of $\|(|\mathrm{sign}\, \mathbf{A}|)\|_2 \le l_s$ and

$$\|(|\mathbf{A}^{(s)}|)\|_2 = \alpha_s \|(|\frac{1}{\alpha_s}\mathbf{A}^{(s)}|)\|_2 \le \alpha_s \sqrt{l_s}$$

we obtain by $\gamma_{l_s} = l_s u + \mathcal{O}(u^2)$ that

$$\begin{aligned}\|\mathbf{e}^{(s)}\|_2 &\le \Big(\alpha_s\, l_s^{3/2}\, u + (1 + l_s u)\, \eta_s\, l_s\, u + \mathcal{O}(u^2)\Big)\, \|\hat{\mathbf{y}}^{(s-1)}\|_2 \\ &= \Big((\alpha_s l_s^{3/2} + \eta_s\, l_s)u + \mathcal{O}(u^2)\Big)\, \|\hat{\mathbf{y}}^{(s-1)}\|_2.\end{aligned}$$

From $\|\mathbf{y}^{(s-1)}\|_2 = \alpha_1 \ldots \alpha_{s-1}\|\mathbf{x}\|_2$ (where $\mathbf{y}^{(s-1)}$ are the exact intermediate vectors of the algorithm above) and

$$\|\hat{\mathbf{y}}^{(s-1)}\|_2 \le \|\mathbf{y}^{(s-1)}\|_2 + \|\hat{\mathbf{y}}^{(s-1)} - \mathbf{y}^{(s-1)}\|_2$$

it follows for $s = 2, \ldots, t$ that

$$\begin{aligned}\|\hat{\mathbf{y}}^{(s-1)}\|_2 &\leq (\alpha_1 \ldots \alpha_{s-1})\|\mathbf{x}\|_2 + \|\mathrm{fl}(\hat{\mathbf{A}}^{(s-1)}\hat{\mathbf{y}}^{(s-2)}) - \mathbf{A}^{(s-1)}\hat{\mathbf{y}}^{(s-2)}\|_2 \\ &\quad + \|\mathbf{A}^{(s-1)}\hat{\mathbf{y}}^{(s-2)} - \mathbf{A}^{(s-1)}\mathbf{y}^{(s-2)}\|_2 \\ &\leq (\alpha_1 \ldots \alpha_{s-1})\|\mathbf{x}\|_2 + \|\mathbf{e}^{(s-1)}\|_2 + \alpha_{s-1}\|\hat{\mathbf{y}}^{(s-2)} - \mathbf{y}^{(s-2)}\|_2 \\ &\leq (\alpha_1 \ldots \alpha_{s-1})\|\mathbf{x}\|_2 + \mathcal{O}(u)\|\mathbf{x}\|_2,\end{aligned}$$

where the last inequality follows by an induction argument. Using telescope summation, we find

$$\begin{aligned}\Delta\mathbf{y} &= \hat{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)} \\ &= (\hat{\mathbf{y}}^{(t)} - \mathbf{A}^{(t)}\hat{\mathbf{y}}^{(t-1)}) + \mathbf{A}^{(t)}(\hat{\mathbf{y}}^{(t-1)} - \mathbf{y}^{(t-1)}) \\ &= \mathbf{e}^{(t)} + \mathbf{A}^{(t)}(\hat{\mathbf{y}}^{(t-1)} - \mathbf{y}^{(t-1)}) \\ &= \mathbf{e}^{(t)} + \mathbf{A}^{(t)}\mathbf{e}^{(t-1)} + \mathbf{A}^{(t)}\mathbf{A}^{(t-1)}(\hat{\mathbf{y}}^{(t-2)} - \mathbf{y}^{(t-2)}) \\ &= \mathbf{e}^{(t)} + \mathbf{A}^{(t)}\mathbf{e}^{(t-1)} + \ldots + \mathbf{A}^{(t)} \ldots \mathbf{A}^{(2)}\mathbf{e}^{(1)},\end{aligned}$$

and hence, by $\|\mathbf{A}^{(s)}\|_2 = \alpha_s$ and $\prod_{s=1}^{t} \alpha_s = 1$, we obtain

$$\begin{aligned}\|\Delta\mathbf{y}\|_2 &\leq \sum_{s=1}^{t-1} \|\mathbf{A}^{(t)} \ldots \mathbf{A}^{(s+1)}\|_2\|\mathbf{e}^{(s)}\|_2 + \|\mathbf{e}^{(t)}\|_2 \\ &\leq \sum_{s=1}^{t-1}(\alpha_t \ldots \alpha_{s+1}) \left((\alpha_s l_s^{3/2} + \eta_s\, l_s)u + \mathcal{O}(u^2)\right)\, \|\hat{\mathbf{y}}^{(s-1)}\|_2 \\ &\quad + \left((\alpha_t l_t^{3/2} + \eta_t\, l_t)\, u + \mathcal{O}(u^2)\right)\, \|\hat{\mathbf{y}}^{(t-1)}\|_2 \\ &\leq \sum_{s=1}^{t} \frac{1}{\alpha_s} \left((\alpha_s l_s^{3/2} + \eta_s\, l_s)\, u + \mathcal{O}(u^2)\right)\, \|\mathbf{x}\|_2 \\ &= \sum_{s=1}^{t} \left(\left(l_s^{3/2} + \frac{\eta_s}{\alpha_s} l_s\right)\, u + \mathcal{O}(u^2)\right)\, \|\mathbf{x}\|_2.\end{aligned}$$

This completes the proof. □

This theorem shows that worst case roundoff error heavily depends on the precomputation of the entries in the matrix factors, i.e. on η_s. A best possible stability can only be achieved, if the value $\sum_{s=1}^{t} \frac{\eta_s}{\alpha_s} l_s$ has at most the same magnitude as $\sum_{s=1}^{t} l_s^{3/2}$.

For the complex case we state the following theorem by Tasche and Zeuner [22] without a proof.

Theorem 6. *Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a unitary matrix with a factorization (4), where $\mathbf{A}^{(s)} = (a_{jk}^{(s)})_{j,k=0}^{n-1} \in \mathbb{C}^{n\times n}$ for $s = 1, \ldots, t$, are almost unitary, i.e., $\mathbf{A}^{(s)}(\overline{\mathbf{A}}^{(s)})^T = \alpha_s^2 \mathbf{I}_n$ with $\alpha_s = \alpha(\mathbf{A}^{(s)}) > 0$, and block diagonal (up to some permutation) with blocks of size $\le \lambda_s = \lambda(\mathbf{A}^{(s)})$. Assume that in each row and each column at least $\kappa_s = \kappa(\mathbf{A}^{(s)})$ of the nonzero entries are in $\{\pm 1, \pm i\}$ and that all entries $a_{jk}^{(s)} \notin \{0, \pm 1, \pm i\}$ are precomputed with $|\hat{a}_{jk}^{(s)} - a_{jk}^{(s)}| \le \eta_s u$ with $\eta_s = \eta(\mathbf{A}^{(s)}) > 0$, and $\hat{a}_{jk}^{(s)} = a_{jk}^{(s)}$ for $a_{jk}^{(s)} \in \{0, \pm 1, \pm i\}$. Further let $\hat{\mathbf{A}}^{(s)} = (\hat{a}_{jk}^{(s)})_{j,k=0}^{n-1}$ for $s = 1, \ldots, t$. Then for arbitrary $\mathbf{x} \in \mathbb{C}^n$, the forward error vector*

$$\Delta\mathbf{y} := \mathrm{fl}(\hat{\mathbf{A}}^{(t)}\mathrm{fl}(\ldots \hat{\mathbf{A}}^{(2)}\mathrm{fl}(\hat{\mathbf{A}}^{(1)}\mathbf{x}))) - \mathbf{A}\mathbf{x}$$

satisfies

$$\|\Delta\mathbf{y}\|_2 \le (k_n u + \mathcal{O}(u^2))\, \|\mathbf{x}\|_2$$

with

$$k_n = \sum_{s=1}^{t} (\theta_s + \sqrt{\lambda_s}\, \mu_{\mathbf{C}} + \frac{\eta_s}{\alpha_s}(\lambda_s - \kappa_s)),$$

$$\theta_s := \begin{cases} (\lambda_s - 1)\sqrt{\lambda_s} & \text{for sequential summation,} \\ \lceil \log_2 \lambda_s \rceil \sqrt{\lambda_s} & \text{for cascade summation,} \\ 1 & \text{for } \lambda_s = 2. \end{cases}$$

§4. Fast Fourier transforms

We consider the unitary Fourier matrix

$$\mathbf{F}_n := \frac{1}{\sqrt{n}} (\omega_n^{jk})_{j,k=0}^{n-1}$$

with $\omega_n := \exp(-2\pi i/n)$ and $n \in \mathbb{N}$. The discrete Fourier transform is the linear mapping from $\mathbb{C}^n$ into $\mathbb{C}^n$ induced by the matrix $\mathbf{F}_n$.

Let $\mathbf{x} \in \mathbb{C}^n$. For a direct computation of the matrix-vector product $\mathbf{F}_n\mathbf{x}$ with precomputed entries ω_n^k, we obtain from Corollary 4 the worst case estimate

$$\|\mathrm{fl}(\mathbf{F}_n\mathbf{x}) - \mathbf{F}_n\mathbf{x}\|_2 \le \left((n - 1 + \mu_{\mathbf{C}})\sqrt{n}\, u + \eta\, n\, u + \mathcal{O}(u^2)\right)\, \|\mathbf{x}\|_2,$$

since for $|\mathbf{F}_n|$ and $\tilde{\mathbf{F}}_n = (1)_{j,k=0}^{n-1}$ we have

$$\|(|\mathbf{F}_n|)\|_2 = \sqrt{n}, \qquad \|\tilde{\mathbf{F}}_n\|_2 = n.$$

For the stability constant we hence obtain $k_n = \mathcal{O}(n^{3/2})$, i.e., a direct computation of $\mathbf{F}_n\mathbf{x}$ possesses not only a large arithmetical complexity but also a bad numerical stability.

In the literature, there is a wide variety of fast algorithms for computing $\mathbf{F}_n\mathbf{x}$. An FFT is based on a factorization of the Fourier matrix into a product of sparse, almost unitary matrices. Let us describe the error analysis only for the example of Cooley-Tukey algorithm and different precomputation of twiddle factors.

We need the following notations. The tensor product of two matrices $\mathbf{A} := (a_{jk})_{j,k=0}^{m-1} \in \mathbb{C}^{m\times m}$ and $\mathbf{B} \in \mathbb{C}^{n\times n}$ is by definition the block matrix

$$\mathbf{A} \otimes \mathbf{B} := (a_{jk}\,\mathbf{B})_{j,k=0}^{m-1} \in \mathbb{C}^{mn\times mn}.$$

The direct sum of $\mathbf{A} \in \mathbb{C}^{m\times m}$ and $\mathbf{B} \in \mathbb{C}^{n\times n}$ is defined by

$$\mathbf{A} \oplus \mathbf{B} := \operatorname{diag}(\mathbf{A},\, \mathbf{B}) \in \mathbb{C}^{(m+n)\times(m+n)}.$$

Let now $n = p^t$, where $p, t \in \mathbb{N}$ with $p, t \geq 2$, and let $n_j := p^{t-j}$ for $j = 1, \ldots, t$. With $\mathbf{B}_n^{(j)}$ we denote the radix-p butterfly matrices

$$\mathbf{B}_n^{(j)} := \mathbf{I}_{p^{t-j}} \otimes (\sqrt{p}\,\mathbf{F}_p) \otimes \mathbf{I}_{p^{j-1}}, \qquad (j = 1, \ldots, t),$$

where $\mathbf{I}_r$ is the identity matrix of order r. Further, we consider the radix-p twiddle matrices

$$\begin{aligned}
\mathbf{T}_n^{(1)} &:= \mathbf{I}_n, \\
\mathbf{T}_n^{(j)} &:= \mathbf{I}_{p^{t-j}} \otimes \left(\mathbf{I}_{p^{j-1}} \oplus (\mathbf{W}_{p^{j-1}}(p^j))^1 \oplus \ldots \oplus (\mathbf{W}_{p^{j-1}}(p^j))^{p-1}\right)
\end{aligned}$$

for $j = 2, \ldots, t$ with

$$\mathbf{W}_{p^{j-1}}(p^j) = \operatorname{diag}(\omega_{p^j}^k)_{k=0}^{p^{j-1}-1}.$$

Observe that $\mathbf{B}_n^j$ are sparse, almost unitary matrices with

$$\mathbf{B}^{(j)}(\overline{\mathbf{B}^{(j)}})^T = p\,\mathbf{I}_n$$

and $\mathbf{T}_n^{(j)}$ are unitary diagonal matrices. Then $\mathbf{F}_n$ can be factorized into

$$\mathbf{F}_n = n^{-1/2}\,\mathbf{B}_n^{(t)}\,\mathbf{T}_n^{(t)} \ldots \mathbf{B}_n^{(2)}\,\mathbf{T}_n^{(2)}\,\mathbf{B}_n^{(1)}\,\mathbf{R}_n(p),$$

where $\mathbf{R}_n(p)$ is the radix-p digit-reversal permutation matrix, i.e. with δ denoting the Kronecker symbol,

$$\mathbf{R}_n(p) := (\delta(\operatorname{rev}_n(k) - l))_{k,l=0}^{n-1}.$$

Here, for a p-adic representation $k = \sum_{s=0}^{t-1} k_s p^s$ of $k \in \{0, \ldots, n-1\}$ the reversion is given by

$$\operatorname{rev}_n(k) = \sum_{s=0}^{t-1} k_s p^{t-s-1}$$

(see e.g. [19, 21]).

We consider the precomputation of twiddle factors

$$\omega_{p^j}^k = \cos\frac{2\pi k}{p^j} - i\sin\frac{2\pi k}{p^j}, \quad j = 2,\dots,t,\ k = 0,\dots p^{j-1}-1.$$

The most expensive, but also most accurate computation of $\omega_{p^j}^k$ is the *direct call*. If the library routines for sine and cosine are of high quality, one can obtain an error estimate

$$|\hat{\omega}_{p^j}^k - \omega_{p^j}^k| \le \frac{\sqrt{2}}{2}u$$

for the precomputed value $\hat{\omega}_{p^j}^k$.

A faster method, based on only two calls of trigonometric functions, is the *repeated multiplication*

$$\begin{aligned}\hat{\omega}_n &:= \mathrm{fl}(\cos(\frac{2\pi}{n})) - i\,\mathrm{fl}(\sin(\frac{2\pi}{n}))\\ \hat{\omega}_n^k &:= \mathrm{fl}(\hat{\omega}_n \times \hat{\omega}_n^{k-1}), \qquad (k = 2,\dots,n-1).\end{aligned}$$

For the roundoff error, we have then the upper bound

$$|\hat{\omega}_{p^j}^k - \omega_{p^j}^k| \le (\mu_{\mathbf{C}} + \frac{\sqrt{2}}{2})\,\frac{kn}{p^j}\,u, \qquad j = 2,\dots,t,\ k = 1,\dots,p^{j-1}-1.$$

Hence for high powers of ω_n the error is of size $\mathcal{O}(nu)$.

Finally we consider the *repeated subvector scaling* which combines the above methods. Compute by direct call for $j = 1,\dots,t$ and $r = 1,\dots,p-1$

$$\hat{\omega}_{p^j}^r = \mathrm{fl}(\cos(\frac{2\pi r}{p^j})) - i\,\mathrm{fl}(\sin(\frac{2\pi r}{p^j}))$$

and then for $j = 1,\dots,t-1$ and $r = 1,\dots,p-1$

$$(\hat{\omega}_n^k)_{k=rn_j+1}^{(r+1)n_j-1} := \mathrm{fl}(\hat{\omega}_{p^j}^r \times (\hat{\omega}_n^k)_{k=1}^{n_j-1}).$$

Hence, for the computation of ω_n^k with $k = \sum_{l=0}^{t-1} k_l\,p^l$ we need at most $A(k) := \#\{l \in \{0,\dots,t-1\} : k_l > 0\}$ direct calls and $A(k)$ complex multiplications such that we arrive at

$$\begin{aligned}|\hat{\omega}_n^k - \omega_n^k| &\le (A(k)-1)\mu_{\mathbf{C}}\,u + \frac{\sqrt{2}}{2}A(k)\,u + \mathcal{O}(u^2)\\ &\le \log_p n\,(\mu_{\mathbf{C}} + \frac{\sqrt{2}}{2})u - \mu_{\mathbf{C}}u + \mathcal{O}(u^2).\end{aligned}$$

By $c_{n,j}$ we denote an upper bound of

$$\max\left\{\frac{|\hat{\omega}_{p^j}^k - \omega_{p^j}^k|}{u} : \; k = 1, \ldots, p^j - 1\right\}.$$

Then we obtain

$$c_{n,j} = \begin{cases} \frac{\sqrt{2}}{2} & \text{for direct call,} \\ (\mu_{\mathbf{C}} + \frac{\sqrt{2}}{2})\, n & \text{for repeated multiplication,} \\ (\mu_{\mathbf{C}} + \frac{\sqrt{2}}{2})j - \mu_{\mathbf{C}} & \text{for repeated subvector scaling.} \end{cases}$$

We want to apply Theorem 6 to the above Cooley-Tukey factorization of $\mathbf{F}_n$, where

$$\begin{aligned} \lambda(\mathbf{B}_n^{(s)}) &= \lambda(\sqrt{p}\,\mathbf{F}_p) = p, \qquad \kappa(\mathbf{B}_n^{(s)}) = \kappa(\sqrt{p}\,\mathbf{F}_p) = 1, \\ \alpha(\mathbf{B}_n^{(s)}) &= \sqrt{p}, \qquad \eta(\mathbf{B}_n^{(s)}) = c_{n,1}, \quad (s = 1, \ldots, t), \end{aligned}$$

since the entries of $\sqrt{p}\,\mathbf{F}_p$ are the roots of unity ω_p^k, $k = 0, \ldots, p-1$. For the diagonal matrices $\mathbf{T}_n^{(s)}$ we find

$$\begin{aligned} \lambda(\mathbf{T}_n^{(s)}) &= 1, \qquad \kappa(\mathbf{T}_n^{(s)}) = 0, \\ \alpha(\mathbf{T}_n^{(s)}) &= 1, \qquad \eta(\mathbf{T}_n^{(s)}) = c_{n,s}, \quad (s = 1, \ldots, t). \end{aligned}$$

The permutation matrix $\mathbf{R}_n(p)$ does not contribute to the roundoff error. Hence we find the stability constant for sequential summation

$$k_n = t\sqrt{p}\left(p - 1 + \mu_{\mathbf{C}} + c_{n,1}\frac{(p-1)}{p}\right) + t\mu_{\mathbf{C}} + \sum_{s=1}^{t} c_{n,s}.$$

Especially we have

$$k_n = \begin{cases} \mathcal{O}(\log_p n) & \text{for direct call,} \\ \mathcal{O}(n \log_p n) & \text{for repeated multiplication,} \\ \mathcal{O}((\log_p n)^2) & \text{for repeated subvector scaling.} \end{cases}$$

Remarks. *The worst case roundoff errors of the Cooley-Tukey FFT with accurately precomputed twiddle factors have been already studied by Ramos [13] and Yalamov [26]. The PhD thesis of Chu [5] contains a comprehensive worst case study of the Cooley-Tukey and the Gentleman-Sande FFT, where numerical errors caused by precomputation of the twiddle factors is also especially considered.*

Calvetti [4] presented at first an average case analysis of the roundoff errors for direct DFT and Cooley-Tukey-FFT for accurate twiddle factors

and non-random input data. In Arioli et al. [1], worst case and average case analysis of roundoff errors for Gentleman-Sande FFT has been studied. In an interesting report of Schatzman [14] it has been shown by a series of numerical tests that all twiddle factors should be precomputed by the most accurate method, the direct call. These results have been founded theoretically in [2, 23, 24]. Especially, we want to refer to the survey paper by Tasche and Zeuner [22] for a comprehensive error analysis of different FFT's, covering the worst case as well as the average case and showing the strong influence of precomputation errors.

§5. Fast cosine transforms

Now we consider discrete cosine transforms (DCT) which are generated by cosine matrices of type II - IV given by

$$\begin{aligned}
\mathbf{C}_n^{II} &:= \sqrt{\frac{2}{n}} \left(\epsilon_j^{(n)} \cos \frac{j(2k+1)\pi}{2n} \right)_{j,k=0}^{n-1}, \\
\mathbf{C}_n^{III} &:= (\mathbf{C}_n^{II})^T, \\
\mathbf{C}_n^{IV} &:= \sqrt{\frac{2}{n}} \left(\cos \frac{(2j+1)(2k+1)\pi}{2n} \right)_{j,k=0}^{n-1}
\end{aligned}$$

with $\epsilon_0^{(n)} = 2^{-1/2}$ and $\epsilon_j^{(n)} = 1$ for $j = 1, \ldots, n-1$. Note that these cosine matrices are orthogonal.

We first consider a direct computation of the matrix-vector product $\mathbf{C}_n^{II}\mathbf{x}$ with $\mathbf{x} \in \mathbb{R}^n$ and want to apply Theorem 2 and formula (3). We obtain

$$\begin{aligned}
\|(|\text{sign } \mathbf{C}_n^{II}|)\|_2 &= n, \\
\|(|\mathbf{C}_n^{II}|)\|_2^2 &\le \|\mathbf{C}_n^{II}\|_1 \, \|\mathbf{C}_n^{II}\|_\infty \\
&= \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{2n}} (\cot(\frac{\pi}{4n}) - 1) \right) \sqrt{n} \\
&= 1 + \frac{1}{\sqrt{2}} \left(\cot(\frac{\pi}{4n}) - 1 \right).
\end{aligned}$$

For the equality $\|\mathbf{C}_n^{II}\|_1 = \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{2n}}(\cot(\frac{\pi}{4n}) - 1)$ we refer to [11]. Observe that by Taylor expansion $\cot \frac{\pi}{4n} \le \frac{4n}{\pi}$ such that

$$\|(|\mathbf{C}_n^{II}|)\|_2 \le \left(\frac{2\sqrt{2}n}{\pi} + 1 - \frac{1}{\sqrt{2}} \right)^{1/2}.$$

Applying (3) it follows with precomputation of matrix entries with bound $\eta\, u$

$$\|\text{fl}(\hat{\mathbf{C}}_n^{II}\mathbf{x}) - \mathbf{C}_n^{II}\mathbf{x}\|_2 \le \left(\left(\frac{2\sqrt{2}n}{\pi} + 1 - \frac{\sqrt{2}}{2} \right)^{1/2} nu + \eta\, nu + \mathcal{O}(u^2) \right) \|\mathbf{x}\|_2$$

such that we have $k_n = \mathcal{O}(n^{3/2})$ as for direct computation of the discrete Fourier transform. This bad numerical stability is also found in numerical examples, see e.g. [3]. Similar estimates follow for the direct computation of $\mathbf{C}_n^{IV}\mathbf{x}$. However, there are a lot of fast algorithms for computing the DCT with $\mathcal{O}(n \log n)$ arithmetical operations.

One idea to compute the DCT efficiently is to use FFT. Let $n := 2^t$, $t \in \mathbb{N}$ with $t > 1$. Observing that

$$\mathbf{C}_n^{III} = \sqrt{\frac{2}{n}}\, \tilde{\mathbf{R}}_n^T\, \tilde{\mathbf{C}}_n\, \mathbf{D}_n$$

with the modified even-odd permutation matrix $\tilde{\mathbf{R}}_n$ given by

$$\tilde{\mathbf{R}}_n\mathbf{x} := (x_0, x_2, \ldots, x_{n-2}, x_{n-1}, x_{n-3}, \ldots, x_3, x_1)^T, \quad \mathbf{x} = (x_j)_{j=0}^{n-1},$$

and with

$$\mathbf{D}_n := \operatorname{diag}(\epsilon_j^{(n)})_{j=0}^{n-1}, \qquad \tilde{\mathbf{C}}_n := \left(\cos\frac{(4j+1)k\pi}{4n}\right)_{j,k=0}^{n-1},$$

one obtains

$$\tilde{\mathbf{C}}_n = \operatorname{Re}\left(\omega_{4n}^{(4j+1)k}\right)_{j,k=0}^{n-1} = \sqrt{n}\operatorname{Re}(\mathbf{F}_n \operatorname{diag}(\omega_{4n}^k)_{k=0}^{n-1}).$$

Application of the Cooley-Tukey algorithm provides a stability constant $k_n = \mathcal{O}(\log_2 n)$, i.e., such an algorithm is perfectly stable (see e.g. [3]).

But, since the cosine matrices are real, one likes to have fast algorithms working in real arithmetic only.

In the remaining part of this section we want to give an orthogonal, real factorization of $\mathbf{C}_n$ into sparse, almost orthogonal matrices and show that the corresponding fast split-radix algorithm is again excellently stable with $k_n = \mathcal{O}(\log_2 n)$ (see [12]).

In addition to the notations used in Section 4 we need here the following special matrices. Let $\mathbf{J}_n := (\delta(j+k-n+1))_{j,k=0}^{n-1}$ be the counteridentity matrix, where δ is again the Kronecker symbol. Further, $\Sigma_n := \operatorname{diag}((-1)^k)_{k=0}^{n-1}$ is the diagonal sign matrix. For even n, $\mathbf{R}_n$ denotes the even-odd permutation matrix defined by

$$\mathbf{R}_n\mathbf{x} := (x_0, x_2, \ldots, x_{n-2}, x_1, x_3, \ldots, x_{n-1})^T, \qquad \mathbf{x} = (x_j)_{j=0}^{n-1}. \tag{5}$$

First we observe that the matrices $\mathbf{C}_n^{II}$ and $\mathbf{C}_n^{IV}$ satisfy the factorizations

$$\begin{aligned}\mathbf{C}_n^{II} &= \mathbf{R}_n^T\,(\mathbf{C}_{n_1}^{II} \oplus \mathbf{C}_{n_1}^{IV})\,\mathbf{T}_n(0),\\ \mathbf{C}_n^{IV} &= \mathbf{R}_n^T\,\mathbf{A}_n(1)\,(\mathbf{C}_{n_1}^{II} \oplus \mathbf{C}_{n_1}^{II})\,\mathbf{T}_n(1),\end{aligned}$$

where $n_1 := n/2$, and with the orthogonal sparse matrices,

$$\mathbf{T}_n(0) := \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{I}_{n_1} & \mathbf{J}_{n_1} \\ \mathbf{I}_{n_1} & -\mathbf{J}_{n_1} \end{pmatrix},$$

$$\mathbf{A}_n(1) := \frac{1}{\sqrt{2}} \left(\sqrt{2} \oplus \begin{pmatrix} \mathbf{I}_{n_1-1} & \Sigma_{n_1-1} \\ \mathbf{I}_{n_1-1} & \Sigma_{n_1-1} \end{pmatrix} \oplus \sqrt{2} \right) (\mathbf{I}_{n_1} \oplus \mathbf{J}_{n_1}),$$

$$\mathbf{T}_n(1) := (\mathbf{I}_{n_1} \oplus \Sigma_{n_1}) \times \begin{pmatrix} \operatorname{diag}(\cos \frac{(2k+1)\pi}{4n})_{k=0}^{n_1-1} & \operatorname{diag}(\sin \frac{(2k+1)\pi}{4n})_{k=0}^{n-1} \mathbf{J}_{n_1} \\ -\mathbf{J}_{n_1} \operatorname{diag}(\sin \frac{(2k+1)\pi}{4n})_{k=0}^{n-1} & \operatorname{diag}\left(\mathbf{J}_{n_1} (\cos \frac{(2k+1)\pi}{4n})_{k=0}^{n_1-1} \right) \end{pmatrix}.$$

We consider now the split-radix algorithm for $\mathbf{C}_n^{II}$. Let $n = 2^t$ with $t > 1$ and $n_s := n/2^s$ for $s = 0, \ldots, t-1$. In a first factorization step, $\mathbf{C}_n^{II}$ is split into $\mathbf{C}_{n_1}^{II} \oplus \mathbf{C}_{n_1}^{IV}$. In the second step we split $\mathbf{C}_{n_1}^{II} \oplus \mathbf{C}_{n_1}^{IV}$ into $\mathbf{C}_{n_2}^{II} \oplus \mathbf{C}_{n_2}^{IV} \oplus \mathbf{C}_{n_2}^{II} \oplus \mathbf{C}_{n_2}^{II}$ and so on.

For a complete factorization of $\mathbf{C}_n^{II}$ we introduce binary vectors $\beta_s := (\beta_s(1), \ldots, \beta_s(2^s))$ for $s = 0, \ldots, t-1$, where we put $\beta_s(k) := 0$, if after the s-th factorization step $C_{n_s}^{II}$ stands at position k, and $\beta_s(k) = 1$, if $\mathbf{C}_{n_s}^{IV}$ stands at position k. The vectors β_s satisfy the recursion relation

$$\beta_{s+1} = (\beta_s, \tilde{\beta}_s), \qquad (s = 0, \ldots, t-2),$$

where $\tilde{\beta}_s$ equals to β_s with the exception that the last bit position is reversed. Now, using the matrices

$$\begin{aligned} \mathbf{R}_n(s) &= \mathbf{R}_{n_s}^T \oplus \ldots \oplus \mathbf{R}_{n_s}^T, \qquad (s = 0, \ldots, t-2), \\ \mathbf{A}_n(\beta_s) &= \mathbf{A}_{n_s}(\beta_s(1)) \oplus \ldots \oplus \mathbf{A}_{n_s}(\beta_s(2^s)), \qquad (s = 0, \ldots, t-2), \\ \mathbf{S}_n(\beta_s) &= \sqrt{2}\, (\mathbf{T}_{n_s}(\beta_s(1)) \oplus \ldots \oplus \mathbf{T}_{n_s}(\beta_s(2^s))), \qquad (s = 0, \ldots, t-1) \end{aligned}$$

with $\mathbf{A}_{n_s}(0) = \mathbf{I}_{n_s}$ and $\mathbf{A}_{n_s}(1)$, $\mathbf{T}_{n_s}(0)$, $\mathbf{T}_{n_s}(1)$ as before, we obtain a factorization of $\mathbf{C}_n^{II}$ of the form

$$\mathbf{C}_n^{II} = \frac{1}{\sqrt{n}} \left(\mathbf{R}_n(0) \mathbf{A}_n(\beta_0) \right) \ldots \left(\mathbf{R}_n(t-2) \mathbf{A}_n(\beta_{t-2}) \right) \mathbf{S}_n(\beta_{t-1}) \ldots \mathbf{S}_n(\beta_0)$$

which leads to a fast split-radix DCT-II algorithm (see [12]). Note that all matrix factors are sparse and almost orthogonal. A similar factorization can be derived for $\mathbf{C}_n^{IV}$.

Considering the numerical stability, we apply Theorem 5. Here we have

$$\begin{aligned} l(\mathbf{S}(\beta_s)) &= 2, \qquad \alpha(\mathbf{S}(\beta_s)) = \sqrt{2}, \qquad (s = 0, \ldots, t-1) \\ l(\mathbf{A}(\beta_s)) &= 2, \qquad \alpha(\mathbf{A}(\beta_s)) = 1, \qquad (s = 0, \ldots, t-2). \end{aligned}$$

Hence for direct call of all precomputed entries in $\mathbf{A}(\beta_s)$ and $\mathbf{S}(\beta_s)$ (i.e., $\eta_s = \eta(\mathbf{A}(\beta_s)) = \eta(\mathbf{S}(\beta_s)) = \frac{\sqrt{2}}{2}$), we obtain the worst case stability constant

$$\begin{aligned} k_n &= \sum_{s=0}^{t-1}(2\sqrt{2} + \frac{2\eta_s}{\sqrt{2}}) + \sum_{s=0}^{t-2}(2\sqrt{2} + 2\eta_s) \\ &= (5\sqrt{2}+1)t - 3\sqrt{2}. \end{aligned}$$

This estimate does not take into consideration that a lot of entries in the matrix factors are just ± 1, but we find already $k_n = \mathcal{O}(\log_2 n)$. A more detailed estimate of the roundoff error in [12] leads to

$$k_n = \left(\frac{4}{\sqrt{3}} + 3 + \frac{\sqrt{2}}{2}\right)(\log_2 n - 1) - 1.$$

Remarks. *There are some fast algorithms in the literature which are based on polynomial arithmetic. These algorithms use the idea that all components of $\mathbf{C}_n\mathbf{x}$ can be interpreted as values of one polynomial at n nodes. Reducing the degree of this polynomial by divide-and-conquer technique, one obtains real and fast DCT-algorithms with low arithmetical complexity (see e.g. [7, 17, 18]). A polynomial DCT-algorithm generates a factorization of a cosine matrix with sparse, but non-orthogonal matrix factors, i.e., the factorization does not preserve the orthogonality of $\mathbf{C}_n$. For that reason these fast algorithms have a relatively bad numerical stability with a constant $k_n = \mathcal{O}(n)$, and this is also attested in numerical examples (see e.g. [3, 15, 22]).*

§6. Periodic orthogonal wavelet transforms

Let $h = (h_k)_{k=-\infty}^{\infty}$ be a real, orthogonal low-pass filter of finite length l and let $g = (g_k)_{k=-\infty}^{\infty}$ with $g_k := (-1)^k h_{1-k}$ be the corresponding high-pass filter. For $j \in \mathbb{N}$ and a fixed $n_0 \in \mathbb{N}$ the $2^j n_0$-periodic filter coefficients $h_{j,k}$ and $g_{j,k}$ are then given by

$$h_{j,k} = \sum_{m=-\infty}^{\infty} h_{k+2^j n_0 m}, \qquad g_{j,k} = \sum_{m=-\infty}^{\infty} g_{k+2^j n_0 m}.$$

Observe that for $n_j := 2^j n_0 > l$ these two series contain only one nonzero term. Now, putting

$$\mathbf{H}_j := (h_{j,r-2k})_{r,k=0}^{n_j-1,n_{j-1}-1}, \qquad \mathbf{G}_j := (g_{j,r-2k})_{r,k=0}^{n_j-1,n_{j-1}-1},$$

the discrete periodic wavelet transform (wavelet decomposition) of a vector $\mathbf{s}^0 = (s_k^0)_{k=0}^{n_{j_0}-1}$ of length $n_{j_0} = 2^{j_0} n_0$ can be presented in the form

$$\mathbf{s}^1 = (s_r^1)_{r=0}^{n_{j_0-1}} = \mathbf{H}_{j_0}^T \mathbf{s}^0, \quad \mathbf{d}^1 = (d_r^1)_{r=0}^{n_{j_0-1}} = \mathbf{G}_{j_0}^T \mathbf{s}^0,$$

or equivalently, by

$$s_r^1 = \sum_{k=0}^{n_{j_0}-1} h_{j_0,k-2r}\, s_k^0, \quad d_r^1 = \sum_{k=0}^{n_{j_0}-1} g_{j_0,k-2r}\, s_k^0, \quad (r = 0, \ldots, n_{j_0-1} - 1).$$

The inverse discrete periodic wavelet transform (wavelet reconstruction) is based on

$$\mathbf{s}^0 = \mathbf{H}_{j_0}\mathbf{s}^1 + \mathbf{G}_{j_0}\mathbf{d}^1,$$

or equivalently on

$$s_r^0 = \sum_{k=0}^{n_{j_0-1}-1} (h_{j_0,r-2k}\, s_k^1 + g_{j_0,r-2k}\, d_k^1).$$

The discrete wavelet transform of $\mathbf{s}^0$ through L levels ($1 \le L \le j_0$) is the vector $\mathbf{y}^L = (\mathbf{s}^L, \mathbf{d}^L, \mathbf{d}^{L-1}, \ldots, \mathbf{d}^1)^T$, where $\mathbf{d}^j$ and $\mathbf{s}^j$ are of length n_{j_0-j} and where for

$$\mathbf{M}_j = (\mathbf{H}_j, \mathbf{G}_j) \tag{6}$$

we have

$$\mathbf{y}^{j+1} = (\mathbf{M}_{j_0-j}^T \oplus \mathbf{I}_{n_{j_0}-n_{j_0-j}})\, \mathbf{y}^j, \qquad (j = 0, \ldots, L-1).$$

Hence,

$$\mathbf{y}^L = (\mathbf{M}_{j_0-L+1}^T \oplus \mathbf{I}_{n_{j_0}-n_{j_0-L+1}}) \ldots (\mathbf{M}_{j_0-1}^T \oplus \mathbf{I}_{n_{j_0}-n_{j_0-1}})\mathbf{M}_{n_{j_0}}^T \mathbf{s}^0.$$

In particular, we have $\mathbf{y}^1 = (\mathbf{s}^1, \mathbf{d}^1)$ with $\mathbf{y}^1 = \mathbf{M}_{j_0}^T\mathbf{s}^0$ and $\mathbf{s}^0 = \mathbf{M}_{j_0}\mathbf{y}^1$, since the transform matrices $\mathbf{M}_j$ are orthogonal. Moreover, since the filters h and g have at most l nonzero coefficients, the matrices $\mathbf{M}_j$ are sparse and contain at most l nonzero entries per row and per column.

Using Theorem 2 and formula (3), and assuming that the nonzero filter coefficients h_k are precomputed with

$$|\hat{h}_k - h_k| \le \eta\, u,$$

we obtain an error bound for the forward error of the matrix-vector product $\mathbf{M}_{j_0}^T\mathbf{s}^0$ (for sequential summation) of the form

$$\|\mathrm{fl}(\hat{\mathbf{M}}_{j_0}^T\mathbf{s}^0) - \mathbf{M}_{j_0}^T\mathbf{s}^0\|_2 \le (l(\sqrt{l} + \eta)u + \mathcal{O}(u^2))\|\mathbf{s}^0\|_2. \tag{7}$$

Analogously, for the wavelet decomposition through L levels it follows by Theorem 5 with the transform matrix

$$\mathbf{W}_L := (\mathbf{M}_{j_0-L+1}^T \oplus \mathbf{I}_{n_{j_0}-n_{j_0-L+1}}) \ldots (\mathbf{M}_{j_0-1}^T \oplus \mathbf{I}_{n_{j_0}-n_{j_0-1}})\mathbf{M}_{n_{j_0}}^T$$

the estimate

$$\|\mathrm{fl}(\hat{\mathbf{W}}_L \mathbf{s}^0) - \mathbf{W}_L \mathbf{s}^0\|_2 \le (L\, l\, (\sqrt{l} + \eta)\, u + \mathcal{O}(u^2))\|\mathbf{s}^0\|_2.$$

Hence, for accurate precomputation of the filter coefficients (by direct call) and for small filter lengths, the periodic orthogonal wavelet transform is perfectly stable.

But for longer filter length l one can ask whether by orthogonal factorization of $\mathbf{M}_j$ the arithmetical complexity as well as the numerical stability can even be improved. Indeed such factorizations can be found. In the remaining part of this section, we want to present a new orthogonal matrix factorization of $\mathbf{M}_j$ and show its connection with a factorization of the corresponding polyphase matrix.

First, recall that the orthogonality of the filter h implies that for all $k \in \mathbb{Z}$

$$\sum_{r=-\infty}^{\infty} h_r\, h_{r-2k} = \delta(k). \tag{8}$$

In particular, it follows that the length l of the filter h is even. Let now $\mathbf{V}_j \in \mathbb{R}^{n_j \times n_j}$ be the circulant backward shift matrix of order n_j given by

$$\mathbf{V}_j \mathbf{x} := (x_1, x_2, \ldots, x_{n_j-1}, x_0)^T, \qquad \mathbf{x} = (x_k)_{k=0}^{n_j-1}.$$

Then $\mathbf{V}_j^{-1} = \mathbf{V}_j^T$ is the forward shift matrix with

$$\mathbf{V}_j^T \mathbf{x} = (x_{n_j-1}, x_0, x_1, \ldots, x_{n_j-2})^T.$$

Recall that with $\mathbf{G}_2 \in \mathbb{R}^{2\times 2}$ the matrix

$$\mathbf{I}_{n_{j-1}} \otimes \mathbf{G}_2 = \mathrm{diag}\,(\mathbf{G}_2, \ldots, \mathbf{G}_2) \in \mathbb{R}^{n_j \times n_j}$$

is a block diagonal matrix. Then we find

Theorem 7. *Let $\mathbf{M}_j \in \mathbb{R}^{n_j \times n_j}$ be the matrix of the form (6) determined by an orthogonal filter $h = (h_k)_{k=-\infty}^{\infty}$ of length $l < n_j$, where $h_0 \neq 0$, $h_{l-1} \neq 0$, and $h_k = 0$ for all $k \in \mathbb{Z} \setminus \{0, \ldots, l-1\}$. Then $\mathbf{M}_j$ can be factorized in the form*

$$\mathbf{M}_j = \mathbf{A}_j^1\, \mathbf{V}_j^T\, \mathbf{M}_j^1\, (\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}), \tag{9}$$

where $\mathbf{A}_j^1 := \mathbf{I}_{n_{j-1}} \otimes \mathbf{G}_2^1$ with

$$\mathbf{G}_2^1 := \frac{1}{\sqrt{h_0^2 + h_1^2}} \begin{pmatrix} -h_1 & h_0 \\ h_0 & h_1 \end{pmatrix},$$

and where $\mathbf{M}_j^1$ is an orthogonal matrix of the form (6) determined by the orthogonal filter $h^1 = (h_k^1)_{k=-\infty}^{\infty}$ of length $l-2$ given by

$$h_{2k}^1 = \frac{1}{\sqrt{h_0^2 + h_1^2}}(h_0 h_{2k} + h_1 h_{2k+1}) \qquad (k = 0, \ldots, l/2-2),$$

$$h_{2k-1}^1 = \frac{1}{\sqrt{h_0^2 + h_1^2}}(h_0 h_{2k+1} - h_1 h_{2k}) \qquad (k = 1, \ldots, l/2-1).$$

The proof of this theorem will be given later in this section.

Example. *For the Daubechies D2-filter with the nonzero coefficients*

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}},$$

we obtain the transform matrix $\mathbf{M}_j$*, which reads for instance for* $n_0 = 1$ *and* $j = 3$*,*

$$\mathbf{M}_3 = \begin{pmatrix} h_0 & 0 & 0 & h_2 & h_1 & h_3 & 0 & 0 \\ h_1 & 0 & 0 & h_3 & -h_0 & -h_2 & 0 & 0 \\ h_2 & h_0 & 0 & 0 & 0 & h_1 & h_3 & 0 \\ h_3 & h_1 & 0 & 0 & 0 & -h_0 & -h_2 & 0 \\ 0 & h_2 & h_0 & 0 & 0 & 0 & h_1 & h_3 \\ 0 & h_3 & h_1 & 0 & 0 & 0 & -h_0 & -h_2 \\ 0 & 0 & h_2 & h_0 & h_3 & 0 & 0 & h_1 \\ 0 & 0 & h_3 & h_1 & -h_2 & 0 & 0 & -h_0 \end{pmatrix}.$$

With

$$\frac{h_0}{\sqrt{h_0^2+h_1^2}} = \frac{1}{2}, \qquad \frac{h_1}{\sqrt{h_0^2+h_1^2}} = \frac{\sqrt{3}}{2}$$

we find

$$\mathbf{M}_j = \left(\mathbf{I}_{n_{j-1}} \otimes \frac{1}{2}\begin{pmatrix} -\sqrt{3} & 1 \\ 1 & \sqrt{3} \end{pmatrix}\right) \mathbf{V}_j^T \, \mathbf{M}_j^1 \, (\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}),$$

where $\mathbf{M}_j^1$ *is induced by* h^1 *with*

$$h_0^1 = \frac{1+\sqrt{3}}{2\sqrt{2}}, \quad h_1^1 = \frac{1-\sqrt{3}}{2\sqrt{2}}, \quad h_k^1 = 0 \quad \textit{for} \quad k \in \mathbb{Z} \setminus \{0, 1\}.$$

As we shall see in the following, the factorization of the transform matrix $\mathbf{M}_j$ for periodic orthogonal wavelet transform given in Theorem 7 can also be interpreted as a factorization of the corresponding polyphase matrix. Let $h(z) := \sum_{k=0}^{l-1} h_k z^k$ be the z-transform of the orthogonal filter h of length l and consider the polynomials

$$h_e(z) := \sum_{k=0}^{l/2-1} h_{2k} z^k, \qquad h_o(z) := \sum_{k=0}^{l/2-1} h_{2k+1} z^k,$$

such that $h(z) = h_e(z^2) + z h_o(z^2)$. Then the Laurent polynomial matrix

$$P(z) = \begin{pmatrix} h_e(z) & -h_o(1/z) \\ h_o(z) & h_e(1/z) \end{pmatrix} \qquad (z \in \mathbb{C} \setminus \{0\}) \tag{10}$$

is called *polyphase matrix* corresponding to h. The orthogonality (8) of h is equivalent with

$$P(z) P(1/z)^T = \mathbf{I}_2.$$

Now we obtain

Theorem 8. *For a polyphase matrix $P(z)$ determined by an orthogonal filter $h = (h_k)_{k=-\infty}^{\infty}$ with $h_0 \neq 0$, $h_{l-1} \neq 0$ and $h_k = 0$ for all $k \in \mathbb{Z} \setminus \{0, \ldots, l-1\}$, there exists a factorization of the form*

$$P(z) = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{z} \end{pmatrix} \left[\prod_{k=0}^{l/2-1} \begin{pmatrix} c_0^k & -c_1^k \\ c_1^k z & c_0^k z \end{pmatrix} \right] \begin{pmatrix} 1 & 0 \\ 0 & z^{-l/2+1} \end{pmatrix},$$

where $(c_0^k)^2 + (c_1^k)^2 = 1$ for $k = 0, \ldots, l/2-1$. In particular, for $|z| = 1$ the matrix factors of $P(z)$ are unitary.

Proof: We shall give a constructive proof for this factorization. Let $P(z)$ be of the form (10) and $h_e(z)$, $h_o(z)$ defined as above. We choose

$$c_0^0 = \frac{h_0}{\sqrt{h_0^2 + h_1^2}}, \qquad c_1^0 = \frac{h_1}{\sqrt{h_0^2 + h_1^2}}$$

and find

$$\begin{aligned} \tilde{P}^1(z) &:= \begin{pmatrix} c_0^0 & -c_1^0 \\ c_1^0 z & c_0^0 z \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 0 & z \end{pmatrix} P(z) \\ &= \begin{pmatrix} c_0^0 & c_1^0 \\ -c_1^0 & c_0^0 \end{pmatrix} \begin{pmatrix} h_e(z) & -h_o(1/z) \\ h_o(z) & h_e(1/z) \end{pmatrix} \\ &= \frac{1}{\sqrt{h_0^2+h_1^2}} \begin{pmatrix} \sum_{k=0}^{l/2-1} (h_0 h_{2k} + h_1 h_{2k+1}) z^k & \sum_{k=0}^{l/2-1} (-h_0 h_{2k+1} + h_1 h_{2k}) z^{-k} \\ \sum_{k=0}^{l/2-1} (-h_1 h_{2k} + h_0 h_{2k+1}) z^k & \sum_{k=0}^{l/2-1} (h_1 h_{2k+1} + h_0 h_{2k}) z^{-k} \end{pmatrix}. \end{aligned}$$

Putting

$$h_{2k}^1 := \frac{h_0 h_{2k} + h_1 h_{2k+1}}{\sqrt{h_0^2 + h_1^2}}, \qquad h_{2k-1}^1 := \frac{-h_1 h_{2k} + h_0 h_{2k+1}}{\sqrt{h_0^2 + h_1^2}},$$

we see that $h_{-1}^1 = 0$ by definition and $h_{l-2}^1 = 0$ by (8). Further by (8), $h^1 = (h_k^1)_{k=-\infty}^{\infty}$ is again an orthogonal filter. With

$$h_e^1(z) = \sum_{k=0}^{l/2-1} h_{2k}^1 z^k, \qquad h_o^1(z) = \sum_{k=0}^{l/2-1} h_{2k+1}^1 z^k$$

we obtain

$$\tilde{P}^1(z) = \begin{pmatrix} h_e^1(z) & -\frac{1}{z} h_o^1(1/z) \\ z h_o^1(z) & h_e^1(1/z) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & z \end{pmatrix} P^1(z) \begin{pmatrix} 1 & 0 \\ 0 & 1/z \end{pmatrix},$$

where $P^1(z)$ is the polyphase matrix corresponding to h^1. By construction is now $h^1_k = 0$ for $k \in \mathbb{Z} \setminus \{0, \ldots, l-3\}$. Hence we have

$$P(z) = \begin{pmatrix} 1 & 0 \\ 0 & 1/z \end{pmatrix} \begin{pmatrix} c_0^0 & -c_1^0 \\ c_1^0 z & c_0^0 z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z \end{pmatrix} P^1(z) \begin{pmatrix} 1 & 0 \\ 0 & 1/z \end{pmatrix}.$$

The procedure can now be applied to $P^1(z)$ (instead of $P(z)$) and so on. Finally we arrive at the wanted factorization of $P(z)$. □

Now we show the connection between the above factorization of the polyphase matrix $P(z)$ and the factorization of the transform matrix $\mathbf{M}_j$ giving the

Proof of Theorem 7: Let $\mathbf{R}_{n_j}$ be the even-odd permutation matrix in (5). Then we obtain

$$\begin{aligned} \mathbf{R}_{n_j}\, \mathbf{M}_j &= \begin{pmatrix} (h_{j,2k-2r})_{k,r=0}^{n_{j-1}-1} & (h_{j,2r-2k+1})_{k,r=0}^{n_{j-1}-1} \\ (h_{j,2k-2r+1})_{k,r=0}^{n_{j-1}-1} & (-h_{j,2r-2k})_{k,r=0}^{n_{j-1}-1} \end{pmatrix} \\ &= \begin{pmatrix} \sum\limits_k h_{j,2k} \mathbf{V}_{j-1}^{-k} & \sum\limits_k h_{j,2k+1} \mathbf{V}_{j-1}^{k} \\ \sum\limits_k h_{j,2k+1} \mathbf{V}_{j-1}^{-k} & -\sum\limits_k h_{j,2k} \mathbf{V}_{j-1}^{k} \end{pmatrix} \\ &= \begin{pmatrix} h_e(\mathbf{V}_{j-1}^T) & h_o(\mathbf{V}_{j-1}) \\ h_o(\mathbf{V}_{j-1}^T) & -h_e(\mathbf{V}_{j-1}) \end{pmatrix} \\ &= P(\mathbf{V}_{j-1}^T)\, (\mathbf{I}_{n_{j-1}} \oplus (-\mathbf{I}_{n_{j-1}})), \end{aligned}$$

where $P(\mathbf{V}_{j-1}^T)$ is the polyphase matrix in (10) with the argument $\mathbf{V}_{j-1}^T$, i.e., a block matrix with four circulant blocks. From the proof of Theorem 8 it follows that

$$\begin{aligned} P(\mathbf{V}_{j-1}^T) = (\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}) \begin{pmatrix} c_0^0\, \mathbf{I}_{n_{j-1}} & -c_1^0\, \mathbf{I}_{n_{j-1}} \\ c_1^0\, \mathbf{V}_{j-1}^T & c_0^0\, \mathbf{V}_{j-1}^T \end{pmatrix} (\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}^T) \times \\ P^1(\mathbf{V}_{j-1}^T)\, (\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}). \end{aligned}$$

Using this relation we find

$$\begin{aligned} \mathbf{M}_j &= \mathbf{R}_{n_j}^T\, P(\mathbf{V}_{j-1}^T)\, (\mathbf{I}_{n_{j-1}} \oplus (-\mathbf{I}_{n_{j-1}})) \\ &= \mathbf{R}_{n_j}^T \begin{pmatrix} c_0^0\, \mathbf{I}_{n_{j-1}} & -c_1^0\, \mathbf{V}_{j-1}^T \\ c_1^0\, \mathbf{I}_{n_{j-1}} & c_0^0\, \mathbf{V}_{j-1}^T \end{pmatrix} P^1(\mathbf{V}_{j-1}^T)\, (\mathbf{I}_{n_j - 1} \oplus (-\mathbf{V}_{j-1})). \end{aligned}$$

With

$$\mathbf{M}_j^1 := \mathbf{R}_{n_j}^T\, P^1(\mathbf{V}_{j-1}^T)\, (\mathbf{I}_{n_{j-1}} \oplus (-\mathbf{I}_{n_{j-1}}))$$

we obtain

$$\mathbf{M}_j = \mathbf{R}_{n_j}^T \begin{pmatrix} c_0^0\,\mathbf{I}_{n_{j-1}} & -c_1^0\,\mathbf{V}_{j-1}^T \\ c_1^0\,\mathbf{I}_{n_{j-1}} & c_0^0\,\mathbf{V}_{j-1}^T \end{pmatrix} \mathbf{R}_{n_j}\,\mathbf{M}_j^1\,(\mathbf{I}_{n_{j-1}} \oplus \mathbf{V}_{j-1}).$$

Finally, by

$$\mathbf{R}_{n_j}^T \begin{pmatrix} c_0^0\mathbf{I}_{n_{j-1}} & -c_1^0\mathbf{V}_{j-1}^T \\ c_1^0\mathbf{I}_{n_{j-1}} & c_0^0\mathbf{V}_{j-1}^T \end{pmatrix} \mathbf{R}_{n_j} = \mathbf{A}_j^1\,\mathbf{V}_j^T$$

the factorization (9) follows. □

We are now ready to apply Theorem 5 to the periodic orthogonal wavelet transform using the factorized polyphase matrix (or equivalently) the factorization of $\mathbf{M}_j$ into $l/2$ orthogonal matrix factors with only two nonzero entries per row and per column. Assuming that the entries c_0^k, c_1^k for $k = 0, \ldots, l/2-1$ in the matrix factors are precomputed with

$$|\hat{c}_0^k - c_0^k| \le \eta u, \qquad |\hat{c}_1^k - c_1^k| \le \eta u,$$

we obtain with this procedure,

$$\|\mathrm{fl}(\hat{\mathbf{M}}_{j_0}^T \mathbf{s}^0) - \mathbf{M}_{j_0}^T \mathbf{s}^0\|_2 \le (l\,(\sqrt{2}+\eta)\,u + \mathcal{O}(u^2))\|\mathbf{s}^0\|_2.$$

Comparing this estimate with (7), we observe that an improvement of the numerical stability by factorization can only be achieved, if the coefficients c_0^k and c_1^k in the matrix factors are computed very accurately.

Remarks. *1. There are other factorizations of the polyphase matrix of an orthogonal filter bank known in the literature. Ladder structures for the efficient realization of perfect reconstruction filter banks have been widely used, see e.g. [6, 20] and references therein. Most such factorizations are non-orthogonal and can by applied also to the biorthogonal wavelet transform. The factorizations of paraunitary filter banks by Vaidyanathan [20], Chapter 14, are non-orthogonal and the number of matrix factors is not directly related to the filter length. The lifting factorization by Daubechies and Sweldens [6] greatly reduces the arithmetical complexity of the wavelet transform. However, this factorization also does not preserve the orthogonality of* $\mathbf{M}_j$. *An exact investigation of the rounding error analysis for the lifting factorization has not been done up to now. An orthogonal matrix factorization of* $\mathbf{M}_j\,\mathbf{R}_{n_j}$ *using Pollen products can be found in [9]. Note that the factorization in [9] strongly differs from ours.*

2. One idea to find good factorizations of biorthogonal polyphase matrices which lead to numerically stable algorithms may be to look for matrix factorizations, where the product of the spectral norms of the matrix factors is minimal.

3. The forward error occurring for periodic biorthogonal wavelet transforms has been estimated by Keinert [10] using spectral norms of corresponding transform matrices. For a comprehensive roundoff error analysis of the worst case and the average case for periodic biorthogonal wavelet transforms (without matrix factorization) we refer to Schumacher [16].

References

1. Arioli, M., H. Munthe-Kaas, and L. Valdettaro, Componentwise error analysis for FFTs with applications to fast Helmholtz solvers, Numer. Algorithms **12** (1996), 65–88.
2. Baszenski, G., R. Xuefang, and M. Tasche, Numerical stability of fast Fourier transforms, in *Advances in Multivariate Approximation*, W. Haußmann, K. Jetter, and M. Reimer (eds.), Wiley, Berlin, 1999, 57–72.
3. Baszenski, G., U. Schreiber, and M. Tasche, Numerical stability of fast cosine transforms, Numer. Funct. Anal. Optim. **21** (2000), 25–46.
4. Calvetti, D., A stochastic roundoff error analysis for the fast Fourier transform, Math. Comp. **56** (1991), 755–774.
5. Chu, C.Y., The fast Fourier transform on hypercube parallel computers, PhD thesis, Cornwell Univ., Ithaca, NY, 1987.
6. Daubechies, I., and W. Sweldens, Factoring wavelet transforms into lifting steps. J. Fourier Anal. Appl. **4** (1998), 247–269.
7. Feig, E., and S. Winograd, Fast algorithms for the discrete cosine transform, IEEE Trans. Signal Process. **40** (1992), 2174–2193.
8. Higham, N.J., *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
9. Kautsky, J., and R. Turcajová, Pollen product factorization and construction of higher multiplicity wavelets, Linear Algebra Appl. **222** (1995), 241–260.
10. Keinert, F., Numerical stability of biorthogonal wavelet transforms, Adv. Comput. Math. **4** (1997), 293–316.
11. Plonka, G., A global method for integer DCT and integer wavelet transforms, preprint, 2003.
12. Plonka, G., and M. Tasche, Split-radix algorithms for discrete trigonometric transforms, preprint, 2002.
13. Ramos, G.U., Roundoff error analysis of the fast Fourier transform, Math. Comp. **25** (1971), 757–768.
14. Schatzman, J.C., Accuracy of the discrete Fourier transform and the fast Fourier transform, SIAM J. Sci. Comput. **17** (1996), 1150–1166.

15. Schreiber, U., Fast and numerically stable trigonometric transforms (in German), thesis, Univ. Rostock, 1999.
16. Schumacher, H., Numerical stability of wavelet algorithms (in German), thesis, Univ. Rostock, 2003.
17. Steidl, G., Fast radix-p discrete cosine transform, Appl. Algebra Engrg. Comm. Comput. **3** (1992), 39–46.
18. Steidl, G., and M. Tasche, A polynomial approach to fast algorithms for discrete Fourier-cosine and Fourier-sine transforms, Math. Comput. **56** (1991), 281–296.
19. Tolimieri, R., M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*, Springer, New York, 1997.
20. Vaidyanathan, P.P., *Multirate Systems and Filter Banks*, Signal Processing Series, Prentice Hall, Englewood Cliffs, N.J., 1992.
21. Van Loan, C., *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
22. Tasche, M., and H. Zeuner, Roundoff error analysis for fast trigonometric transforms, in *Handbook of Analytic-Computational Methods in Applied Mathematics*, G. Anastassiou (ed.), Chapman & Hall/ CRC, Boca Rota, 2000, 357–406.
23. Tasche, M., and H. Zeuner, Roundoff error analysis for the fast Fourier transform with precomputed twiddle factors, J. Comput. Anal. Appl. **4** (2002), 1–18.
24. Tasche, M., and H. Zeuner, Worst and average case roundoff error analysis for FFT, BIT **41** (2001), 563–581.
25. Wilkinson, J.H., Error analysis of floating point computation, Numer. Math. **2** (1960), 319–343.
26. Yalamov, P.Y., Improvements of some bounds on stability of fast Helmholtz solvers, Numer. Algorithms **26** (2001), 11–20.
27. Zeuner, H., Stochastic roundoff error analysis with applications to DFT and DCT, J. Comput. Anal. Appl., to appear.

Gerlind Plonka
Institute of Mathematics
University of Duisburg–Essen
47048 Duisburg, Germany
plonka@math.uni-duisburg.de

Manfred Tasche
Department of Mathematics
University of Rostock
18051 Rostock, Germany
manfred.tasche@mathematik.uni-rostock.de