AN UPDATE ALGORITHM FOR FOURIER COEFFICIENTS

Erich Fuchs, Tobias Hanning and Oliver Schwarz

FORWISS (Bavarian Research Centre for Knowledge-Based Systems), University of Passau
Innstr. 33, D-94032 Passau, Germany (Europe)
phone: +49 851 509 3147, fax: +49 851 509 3142, email: fuchs@forwiss.uni-passau.de
web: www.forwiss.uni-passau.de/

ABSTRACT

In this article we present a new technique to obtain the Discrete Fourier coefficients for a moving data window of an arbitrary length. Unlike the classic approaches we derive an update algorithm by exploiting results of update formulas for orthogonal polynomials. For the vector space of polynomials we define a discrete inner product by evaluating the functions on the complex unit circle at equidistant points. With certain weights for the inner product the coefficients of the best approximating polynomial with respect to this inner product are the wanted Fourier coefficients. Therefore we can apply updating strategies for orthogonal polynomials to obtain Fourier coefficients. By this approach we obtain a constant number of arithmetic operations for every single Fourier coefficient. Moreover the algorithm is numerically stable, fast, and flexible since it can be applied to obtain the Discrete Fourier Transformation for data windows with a length not equal to powers of two.

1. INTRODUCTION

Starting point of our considerations is the view to the computation of coefficients of a Discrete Fourier Transform (DFT) as the solution of a least-squares approximation with trigonometric polynomials by analogy to the computation of a polynomial approximating real discrete measuring values.

Let $Y_N \stackrel{\text{def}}{=} \left\{(x_j, y_j, w_j) \,\middle|\, j \in \{0, \ldots, N-1\}\right\}$ be a set of triples consisting of pairwise distinct nodes $x_j \in \mathbb{R}$, values $y_j \in \mathbb{R}$ measured at these nodes, and positive weights w_j. By computing polynomials $(p_k)_{0 \le k \le N-1}$ orthogonal with respect to the discrete inner product

$$(f\,|\,g)_w \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} f(x_j)\,g(x_j)\,w_j,$$

it is well known that it is possible to write the least-squares approximating polynomial q of degree $n \le N-1$ as $q = \sum_{i=0}^{n} d_i p_i$ with some orthogonal expansion coefficients d_i [2, 7].

Downdating (deleting a triple $(x, y, w) \in Y$) and updating (adding a triple (x_N, y_N, w_N) to Y) techniques for such polynomial least-squares fits were analyzed by Elhay, Golub and Kautsky in [3] making use of special properties of orthogonal polynomials; several algorithms for modifying a polynomial fit of degree n by computing a new set of orthogonal polynomials and the respective least-squares-fit coefficients were derived and compared. These general and efficient algorithms can be used for creating “moving” least-squares fits by updating (include a new measurement) and downdating (discard an old value) an existing solution. Visually, this can be imagined as a window (with variable size depending on the abscissas) moving over the received signal data.

In the case of equidistant data, we can change the point of view in the following way: instead of using a window moving over the data (and changing the orthogonal polynomial basis with every step) one can consider a fixed window and a signal passing by that window. Translating this line of thinking into mathematics leads to the following advantage: one can keep the same orthogonal polynomial basis for every least-squares approximation, and therefore it is only required to modify the orthogonal expansion coefficients representing the least-squares solution in that basis.

Such update techniques for polynomial least-squares approximations in moving time windows restricted to equidistant data were analyzed in [4, 5]. For specific classes of weights (equal weights, exponential weights $w_j = \rho^j$ with $\rho \in \mathbb{R}^+$, polynomial weights) different modification algorithms were developed. Compared to the more general techniques analyzed in [3], these specific methods have the following advantages:

- the algorithms are much simpler and use less arithmetic operations,
- since only multiplications and additions (no square roots etc.) are necessary, these algorithms are especially suitable for digital signal processors (DSP) and
- the numerical behavior is much better.

Key point for the derivation of these algorithms was the simple fact that (orthogonal) polynomials shifted along the real axis can be represented in the basis of the original orthogonal basis.

By analogy to the real case, it is convenient to compute polynomials that are orthogonal with respect to a discrete inner product on the unit circle (Szegö–Polynomials) when approximating a complex-valued function on the unit circle in the least-squares sense. These methods are closely related to least-squares approximations of real-valued functions by trigonometric polynomials. Updating and downdating techniques on the unit circle using Szegö–Polynomials and the representation of the approximating polynomial in the corresponding orthogonal expansion were developed by Ammar, Gragg and Reichel in [9] and [1]. For adding, deleting respectively, a general triple (x, y, w) consisting of a node, a value, and a weight, an inverse QR and a QR algorithm for unitary upper Hessenberg matrices are used. This leads to less arithmetic operations needed compared to a computation of the new approximation polynomial from scratch. With these methods it is again possible to generate least-squares approximations in moving time windows with trigonometric polynomials.

It is the purpose of this paper to derive new algorithms for this problem by the same methods as applied in [5] using the restriction of equidistant nodes and equal weights. This leads to efficient and reliable update algorithms for the coefficients of a DFT computed from continuously arriving data. Compared to methods developed in the area of Fourier transforms [6, 8, 10, 13], this approach allows faster and more flexible algorithms.

In section 2 we describe the theoretical background and we derive the new update algorithm for Fourier coefficients. A comparison with related methods follows in section 3. A few applications are described in section 4 and some remarks concerning future development possibilities in section 5 conclude the paper.

2. THE ALGORITHM

2.1 Theoretical Background

Let $N \in \mathbb{N}$, $\{\theta_0, \ldots, \theta_{N-1}\}$ be a set of pairwise distinct nodes in the interval $[0, 2\pi[$ and $\{w_0^2, \ldots, w_{N-1}^2\}$ a set of positive real weights. For complex-valued functions f and g, defined at the nodes $z_j \stackrel{\text{def}}{=} \exp(i\theta_j)$, $j \in \{0, \ldots, N-1\}$, a discrete inner product on the unit circle is defined by

$$(f\,|\,g) \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} f(z_j)\overline{g(z_j)}\,w_j^2,$$

where $\bar{z}$ is the complex conjugated value of z. This inner product $(\cdot|\cdot)$ uniquely determines a (finite) sequence $(p_k)_{0\le k\le N-1}$ of orthogonal polynomials on the unit circle.

In the special case of equidistant nodes and constant weights it is easy to find the corresponding orthogonal polynomials. Let $w_j^2 = 1$, for all $j \in \{0, \ldots, N-1\}$, and $\theta_j \stackrel{\text{def}}{=} \frac{2\pi j}{N}$ be equidistant nodes in the interval $[0, 2\pi[$. Orthogonal polynomials $\{\varphi_0, \ldots, \varphi_{N-1}\}$ with respect to the inner product

$$(f\,|\,g)_{DFT} \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} f(z_j)\overline{g(z_j)},$$

given the nodes $z_j \stackrel{\text{def}}{=} \exp(i\theta_j) = \exp(\frac{2\pi i j}{N})$ on the unit circle, are

$$\varphi_k(x) \stackrel{\text{def}}{=} x^k$$

for $k \in \{0, \ldots, N-1\}$. This can be proven easily using the summation formula for finite geometric series using the fact $e^{2\pi i m} = 1$ for $m \in \mathbb{Z}$.

Given measured values $y_j \stackrel{\text{def}}{=} f(z_j)$ of an unknown function $f : \mathbb{C} \to \mathbb{C}$ at the nodes z_j, the polynomial $p \in \mathscr{P}_n(\mathbb{C}, \mathbb{C})$ with $n \le N-1$ minimizing

$$\begin{aligned} \|f-p\|_{DFT}^2 &= (f-p\,|\,f-p)_{DFT} \\ &= \sum_{j=0}^{N-1} \left|f(z_j)-p(z_j)\right|^2 = \sum_{j=0}^{N-1} \left|y_j - p(z_j)\right|^2 \end{aligned}$$

is given by the orthogonal expansion

$$p \stackrel{\text{def}}{=} \frac{1}{N}\sum_{k=0}^{n} (f\,|\,\varphi_k)_{DFT}\;\varphi_k$$

due to $(\varphi_k\,|\,\varphi_k)_{DFT} = N$. The coefficients $(f\,|\,\varphi_k)_{DFT} = \sum_{j=0}^{N-1} y_j\,\omega^{jk}$ with $\omega \stackrel{\text{def}}{=} \exp(-\frac{2\pi i}{N})$ of this orthogonal expansion are exactly the Fourier coefficients of a length-N DFT for $k \in \{0, \ldots, N-1\}$.

Since a least-squares approximation on the unit circle is closely related to the construction of a trigonometric polynomial that minimizes the discrete least-squares error given data points $y_j \in \mathbb{R}$ (compare [9]), the case of real-valued functions is also covered.

2.2 The Update Algorithm

Instead of solving one isolated approximation problem we consider now continuously arriving equidistant data, for which may be even overlapping DFTs have to be determined. Using the above mentioned view to the problem of a discrete signal shifted along a fixed window, let's say by $1 \le r \in \mathbb{N}$ data points, we can state the following update formula.

Theorem 2.1 *Let $y_0, \ldots, y_t, y_{t+1}, \ldots$ be continuously arriving equidistant, real or complex measuring data. Define*

$$\hat{y}_{k,t} \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} y_{t+j}\,\omega^{jk}$$

for $k \in \{0, \ldots, n\}$, $n \le N-1$. The best approximating polynomial $p_t \stackrel{\text{def}}{=} \frac{1}{N}\sum_{k=0}^{n} \hat{y}_{k,t}\,\varphi_k$ for the data $y_t, \ldots, y_{t+N-1}$ fulfills the following modification algorithm:

For $k \in \{0, \ldots, n\}$ the new Fourier coefficients $\hat{y}_{k,t+r}$ can be computed from the former coefficients $\hat{y}_{k,t}$ by

$$\hat{y}_{k,t+r} = \omega^{-kr}\left(\hat{y}_{k,t} + \sum_{j=0}^{r-1}\left[y_{t+N+j} - y_{t+j}\right]\omega^{jk}\right). \qquad (1)$$

Proof. For any $k \in \{0, \ldots, n\}$ with $n \le N-1$ the assertion follows from

$$\begin{aligned} \hat{y}_{k,t+r} &= \sum_{j=0}^{N-1} y_{t+r+j}\,\omega^{jk} = \sum_{j=r}^{N-1+r} y_{t+j}\,\omega^{k(j-r)} \\ &= \omega^{-kr}\left(\sum_{j=0}^{N-1} y_{t+j}\,\omega^{jk} + \sum_{j=N}^{N-1+r} y_{t+j}\,\omega^{jk} - \right. \\ &\qquad \left. \sum_{j=0}^{r-1} y_{t+j}\,\omega^{jk}\right) \\ &= \omega^{-kr}\left(\hat{y}_{k,t} + \sum_{j=0}^{r-1} y_{t+N+j}\,\omega^{k(j+N)} - \sum_{j=0}^{r-1} y_{t+j}\,\omega^{jk}\right) \\ &= \omega^{-kr}\left(\hat{y}_{k,t} + \sum_{j=0}^{r-1}\left[y_{t+N+j} - y_{t+j}\right]\omega^{jk}\right). \end{aligned}$$

□

In the special case $r = 1$ the above formula simplifies to

$$\hat{y}_{k,t+1} = \omega^{-k}\left(\hat{y}_{k,t} + y_{t+N} - y_t\right). \qquad (2)$$

From the formulas (1) and (2) one can see that an r-shift of the data in the time domain corresponds to a multiplication with ω^{-kr} in the frequency domain, which is in effect a rotation on the unit circle. An essential prerequisite for hitting the defined nodes after a rotation is the equidistancy of the data.

It is remarkable that the above formula (2) is an algorithm for the computation of a DFT of **arbitrary length** N in moving time windows, which provides after each receipt of a new value the new corresponding Fourier coefficients with the aid of $N+1$ real, complex respectively, additions and $N-1$ complex multiplications (under the assumption that $y_{t+N}-y_t$ is computed only once in advance and that the trivial multiplication with $\omega^0 = 1$ is omitted). In the case of the more general formula (1) representing an $1 < r$-shift, $r(N-1)$ complex multiplications and $r(N+1)$ complex additions are required.

Even if $N = 2^n$ and the Fourier coefficients may be computed by a Fast Fourier Transform (FFT) in $O(N \log_2 N)$, the modification technique for applications in moving time windows in formula (2) requires less arithmetic operations than every time a new FFT-computation. Therefore we have advantages in the computational costs whenever we have to compute DFTs of overlapping data. If we want to compute subsequent DFTs with no overlap using the above methods, we end up at $O(N^2)$ arithmetic operations, but in this case we have won more information since we got all overlapping DFTs in between.

Furthermore theorem 2.1 provides the possibility of an update formula for a single Fourier coefficient which is independent from the computation of other Fourier coefficients due to the fact that the determination of $\hat{y}_{k,t+r}$ requires only the value of $\hat{y}_{k,t}$. This fact allows the computation of the Fourier coefficients in parallel on different independent processors suitable for hard real-time applications.

3. COMPARISON WITH RELATED METHODS

3.1 Existing Techniques

Using the up- and downdating techniques described in [9] and [1], it is in principle possible to create algorithms for the computation of Fourier coefficients in moving time windows by adding a new and deleting an old node/value-pair of an existing solution. Compared to such a try, equation (1) allows an efficient computation of a new DFT after a window shift $r \geq 1$ using **one algorithmic step** applied to the orthogonal expansion (in this case: Fourier) coefficients without modifying the orthogonal basis.

Due to the linear dependency of the computational costs on r of equation (1) it is possible to choose an arbitrary window shift r without loss of efficiency as required by the application.

The new method has advantages if it is necessary to observe specific frequencies continuously over time. Here only the required portion or set of Fourier coefficients has to be determined. This set of Fourier coefficients may be arbitrarily chosen among all the Fourier coefficients of a certain DFT-length N. Furthermore this choice does not depend on the fact that N has to fulfill some restrictions like being a power of 2 as it is required f.i. for FFT–pruning techniques (compare [8, 10]). The most efficient pruning methods reach computational costs of $O(N \log_2 K)$ for a single computation of K coefficients. Another method proposed by Sorensen and Burrus in [13] has the same complexity, but is a little more efficient than pruning techniques and more flexible with respect to the choice of the coefficients to be determined. However, all these methods have restrictions concerning the number and the distribution of the coefficients to be computed. Also the DFT-length N is a decisive factor for the efficiency of some of the methods. The update algorithm presented here is much more flexible concerning these aspects.

Furthermore it is easy to use different DFT-lengths within one application having equation (1) available. Since the costs of the computation of a single Fourier coefficient in moving time windows are constant and therefore independent from the DFT-length N, it is possible to reach a very high resolution in frequency for specific parts of the spectrum with reasonable effort. On the other hand, low frequency portions of a signal may be determined with another DFT-length.

3.2 Implementation Details

Since the DFT of signal being zero at every node has all Fourier coefficients zero, the application of equation (1) does not require an initial DFT- or FFT-computation as an initialization step. Starting with $y_0 = y_1 = \cdots = y_N = 0$ one can immediately use the modification algorithm (1). This simplicity and the fact that only additions and multiplications (e.g. no square roots etc. which are often not available as instructions on DSPs) are required make the algorithm (1) especially suitable for DSP-implementations.

Statistical tests applying the new method to different randomly generated signals were performed and proved satisfactory numerical behavior. In effect we could not find any parameter constellation (Fourier coefficient k, DFT-length N) leading to numerical instabilities of algorithm (1) even after billions of update steps compared to direct DFT-computations of the respective data.

Summing up, the new algorithm has advantages compared to known methods, if one of the following facts or a combination thereof is applicable:

- $N \neq 2^n$,
- only few Fourier coefficients are required,
- an observation of certain frequencies is continuously necessary,
- the usage of different (big) DFT-lengths is desirable.

4. APPLICATIONS

The above advantages are particularly useful for evaluating signals from an optical displacement sensor (see [11]). This sensor is used to measure the lateral position of moving parts of technical objects. The principle of operation makes use of the interference effect of laser light. Thus the lateral position can be determined precisely without special treatment of the surface of the technical object. However the signal generated by the sensor, is oscillating with a frequency which depends on the motion speed of the surface. Thus a simple band-pass filter is not sufficient to determine the position exactly. Further challenges on the evaluation software are dropouts of the signal amplitude and a low-frequency noise signal. Present algorithms eliminate the noise signal via a high-pass filter and count the zeros of the filtered signal to determine the lateral position. However, with this method, the dropouts cannot be compensated, and it is not robust to noise with higher frequency than the chop-off.

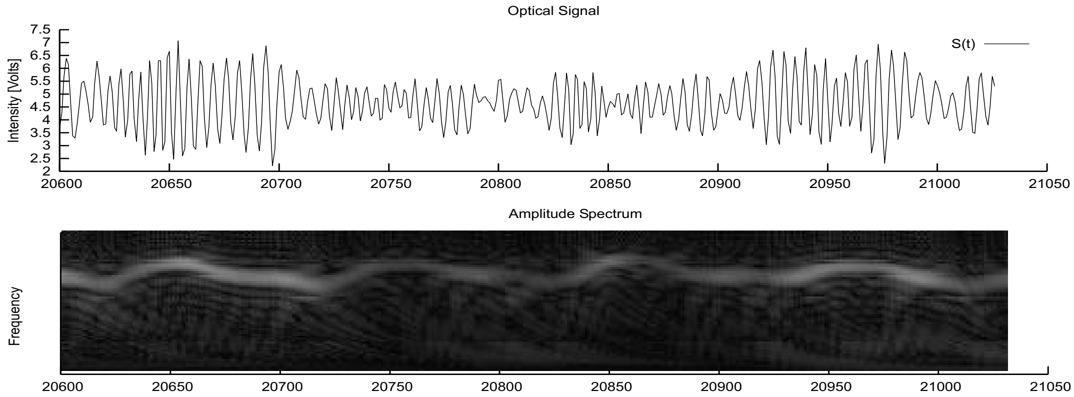

Figure 1: Optical signal and amplitude spectrum.

Using the update technique from section 2.2 it is possible to build an algorithm which determines the frequency and phase in small windows accurately. A main advantage of the update technique is that it is not bound to a specific window length, and that single coefficients can be computed independently. Hence we can analyze several windows with different lengths simultaneously, and we can choose for each window a single coefficient which is computed. Figure 4 shows a portion of a signal together with a grayscale visualization of this special type of amplitude spectrum. While the window size runs from 300 down to 16, the selected Fourier coefficient counts from 1 up to 8.

A second application is in the field of two-dimensional interferometry. The execution speed of a Fourier based phase unwrapping algorithm (see [12]) can be optimized using the update technique. With the Fourier based local phase unwrapping approach, the interferometric image is analyzed inside small overlapping $N \times N$ windows. For each window the position of the peak of the amplitude spectrum is computed. Then a non-linear filter is applied to suppress coefficients with a small absolute value. If the step width r between subsequent windows is less than $\log_2 N$, the update technique is faster than the standard FFT.

5. CONCLUDING REMARKS

From a theoretical point of view, by analogy to [5] the next step should be to investigate, whether it is possible to derive update formulas like in theorem 2.1 for other weights than equal ones. This requires the determination (especially an explicit representation) of the respective orthogonal polynomials on the unit circle as a first step.

After having proven to be useful in several applications, the algorithm could be implemented in hardware for the usage in complex real-time environments.

REFERENCES

[1] G. S. Ammar, W. B. Gragg, and L. Reichel, "Downdating of Szegö Polynomials and Data–Fitting Applications," *Linear Algebra and its Applications*, vol. 172, pp. 315–336, June 1992.

[2] Å. Björck, *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[3] S. Elhay, G. H. Golub, and J. Kautsky, "Updating and Downdating of Orthogonal Polynomials with Data Fitting Applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 12, pp. 327–353, Apr. 1991.

[4] E. Fuchs, "On Discrete Polynomial Least-Squares Approximation in Moving Time Windows," in *Applications and Computation of Orthogonal Polynomials* Birkhäuser Verlag, Basel–Boston–Berlin, 1999, pp. 93–107.

[5] E. Fuchs, *Schnelle Quadratmittelapproximation in gleitenden Zeitfenstern mit diskreten orthogonalen Polynomen*. Shaker Verlag, Aachen, 2000.

[6] G. Goertzel, "An Algorithm for the Evaluation of Finite Fourier Series," *American Mathematical Monthly*, vol. 65(1), pp. 34–35, Jan. 1958.

[7] G. H. Golub and C. F. van Loan, *Matrix Computations*. 3. Ed., John Hopkins University Press, Baltimore, 1996.

[8] J. D. Markel, "FFT Pruning," *IEEE Transactions on Audio and Electroacoustics*, vol. AU–19, pp. 305–311, Dec. 1971.

[9] L. Reichel, G. S. Ammar, and W. B. Gragg, "Discrete Least Squares Approximation by Trigonometric Polynomials," *Mathematics of Computation*, vol. 57, pp. 273–289, July 1991.

[10] D. P. Skinner, "Pruning the decimation in time FFT Algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP–24(2), pp. 193–194, Apr. 1976.

[11] P. Schreiber and U. D. Zeitner, "Design and analysis of a microoptical speckle displacement sensor," *Novel Optical Systems Design and Optimization V, Proceedings of SPIE*, vol. 4768, Jul. 2002.

[12] O. Schwarz, "Improved hybrid phase unwrapping algorithm in speckle interferometry," *Interferometry XI: Techniques and Analysis, Proceedings of SPIE*, vol. 4777, pp. 300–310, Jul. 2002.

[13] H. V. Sorensen and C. S. Burrus, "Efficient Computation of the DFT with Only a Subset of Input or Output Points," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-41(1), pp. 1184–1200, Mar. 1993.