

DECONVOLUTION, DIFFERENTIATION AND FOURIER TRANSFORMATION ALGORITHMS FOR NOISE-CONTAINING DATA BASED ON SPLINES AND GLOBAL APPROXIMATION

H. WORMEESTER, A.G.B.M. SASSE and A. VAN SILFHOUT

University of Twente, Department of Applied Physics, P.O. Box 217, 7500 AE Enschede, The Netherlands

Received 16 January 1988; in revised form 26 June 1988

One of the main problems in the analysis of measured spectra is how to reduce the influence of noise in data processing. We show a deconvolution, a differentiation and a Fourier Transform algorithm that can be run on a small computer (64 K RAM) and suffer less from noise than commonly used routines. This objective is achieved by implementing spline based functions in mathematical operations to obtain global approximation properties in our routines. The convenient behaviour and the pleasant mathematical character of splines makes it possible to perform these mathematical operations on large data input in a limited computing time on a small computer system. Comparison is made with widely used routines.

1. Introduction

Numerical deconvolution, differentiation and Fourier transformation are common practice in dataprocessing. The routines used for these operations are sensitive to noise and may even give a non-unique solution as in the case of deconvolution. Although Fast Fourier Transforms is a powerful tool to manipulate or to analyse measured data [1,2], deconvolution based on FFT routines suffer heavily from noise amplification [3,4]. Differentiation, performed in a point to point manner, can only be performed after intensively smoothing. Discrete Fourier Transform routines lay constraints on the frequency interval obtained. The numerical implementation of these mathematical exact defined operations are responsible for their sensitivity to high frequency components and limits its use and performance.

The main objective of this paper is to show an improvement of the algorithm of these routines and reduce their sensitivity to high frequency components. This goal is approached by implementing spline functions and non-linear least squares minimization. Within this scheme interesting mathematical properties are found which improve the efficiency of the routines.

Our scheme is based on global approximation of a trial function composed of splines on the measured spectrum. In global approximation the line regions over the entire spectrum become correlated. This property suppresses uncorrelated frequencies, such as noise inherent in measuring and high frequency oscillations induced by the numerical manipulations. Approximation with polynomials, such as Lagrange and Hermite polynomials (e.g. ref. [5] and references therein), cannot prevent the induction of local high frequency oscillations, because of the local character of this type of approximation.

Using splines in solving the deconvolution problem has been performed before by Dierckx [6] who used Lagrange Multipliers to obtain an approximation. According to Chornik et al. [4] this required the use of a large computer. Differentiation of noise-containing data, using splines, has been subject of study before (Gutman [7]). However, his definition of splines is numerical less convenient as ours.

As shown before [8,9] we use an iterative minimizing procedure based on a modified Levenberg–Marquardt routine (LM) [10]. The inversion of an operator as the convolution, which results in a deconvolution routine is obtained by letting an

operator, e.g. the convolution, act on the trial function before computing the minimum of the least squares difference with the measured spectrum. An inversion of the operator is obtained without explicit knowledge of the inverted operator. We show also that the self-consistent calculation of the spline coefficients in the LM routine can be expressed in terms of a constant symmetric Toeplitz matrix [11]. This leads to the ability to use larger input arrays of data in our programs and to improve the execution time significantly.

The deconvolution routine with splines [8,9] reduces the ambiguous infinite set of mathematical solutions to a finite set of physical relevant solutions, such as the set of bounded continuous functions. The continuity of the solution is important for the physical interpretation of the deconvolution result.

The use of splines in our method cannot only assure global approximation but they can be generated with a recursive sequence that can easily be implemented in a computer algorithm. Furthermore, splines can easily be manipulated; for example, the derivative of a function represented by splines can be written as a combination of a different kind of splines. The Fourier Transform of a spline representable function can be evaluated using the timeshift theorem and the FT of one spline. This modification of the FT improves its numerical behaviour significantly. The definition of splines assures that all physical relevant functions, i.e. continuous and bounded on a certain interval can be represented unambiguously by a set of spline coefficients [5].

All our tests were performed on an Olivetti M24 IBM compatible personal computer with an 8087 coprocessor (64 K RAM).

This paper is divided in seven sections. In section two the mathematical background of splines is presented. In section three the iterative minimizing method (LM) used is discussed. The implementation of the deconvolution and differentiation algorithm are given in section four and comparison with widely used methods are given. Section five deals with the continuity problem of the solutions of the deconvolution operation and a comparison of our scheme with Fourier Transform based deconvolution routines is made. The

spline based Fourier Transform implementation is shown in section six and in section seven concluding remarks are presented.

2. Splines

Global approximation of a function $D^*(x)$, can be achieved by constructing a function $D(x, \mathbf{c})$ on a basis of piecewise continuous functions [5,12]. These functions are connected to each other in such a way that their sum and their derivatives are continuous to a certain order $(k-1)$ over the entire function. k is the order of the piecewise continuous function. A very convenient definition of these piecewise continuous functions are splines [5,8,9]. These splines, $B_i^k(x)$, defined on a subinterval (x_i, x_{i+k}) of the entire interval (x_1, x_N) , can be constructed by a recursive sequence of variable order k :

$$B_i^k(x) = \frac{x - x_i}{x_{i+k-1} - x_i} B_i^{k-1}(x) + \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}} B_{i+1}^{k-1}(x) \quad (1)$$

with

$$B_i^1(x) = \begin{cases} 1; & x_i \leq x < x_{i+1}, \\ 0; & \text{elsewhere,} \end{cases}$$

$$k = 2, 3, 4, \dots, \quad i = 1, \dots, N.$$

This is a very convenient definition of splines for developing simple computer algorithms. The number of intervals influenced by a spline, defined as support, is equal to the order of the spline.

The function $D^*(x)$ can be approximated by N splines of order k (eq. (2)). A geometrical interpretation of the coefficients c_i is straightforward (see fig. 1).

$$D(x, \mathbf{c}) = \sum_i^N c_i B_i^k(x). \quad (2)$$

When using an equidistant axis, with a the length of one interval, the derivative of a spline function can easily be obtained. It is easily deduced from the definition of splines that the derivative of a function represented by splines of

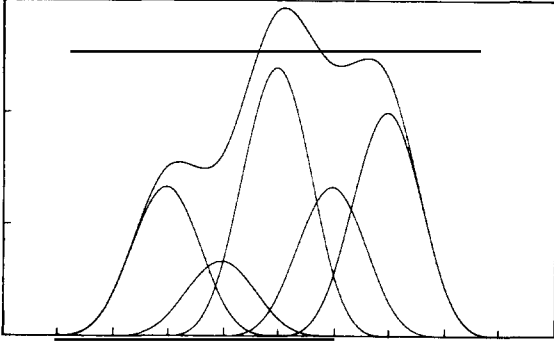


Fig. 1. Five splines of order four with different values for the coefficient on eight intervals. Also is shown the summation. From the plot the geometrical interpretation of eq. (2) is straightforward.

order k can be written as a sum of splines of order $k-1$:

$$\frac{\partial D(x, \mathbf{c})}{\partial x} = \frac{1}{a} \sum_{i=1}^{N+1} (c_i - c_{i-1}) B_i^{k-1}(x) \quad (3)$$

with c_0 and c_{N+1} equal to zero. In this way a new set of coefficients is defined.

A block function, defined on one subinterval (x_i, x_{i+1}) , is by definition a first order spline. Using eq. (3), the Fourier transform $\mathbb{B}_i^k(\omega)$ of a spline of order k can be calculated with the time-shift theorem. The spline is as an even function positioned on the zero of the x -axis,

$$\begin{aligned} \mathbb{B}^k(\omega) &= \frac{1}{i\omega a} \mathbb{B}^{k-1}(\omega) (e^{i\omega a/2} - e^{-i\omega a/2}) \\ &= \frac{2}{\omega a} \mathbb{B}^{k-1}(\omega) \sin\left(\frac{\omega a}{2}\right) \\ &= \left(\frac{2}{\omega a} \sin\left(\frac{\omega a}{2}\right)\right)^{k-1} \mathbb{B}^1(\omega) \\ &= a \left(\text{sinc}\left(\frac{\omega a}{2}\right)\right)^k. \end{aligned} \quad (4)$$

The large overlap between high order ($k > 3$) splines already suggested a smooth behaviour. Eq. (4) shows the reason why spline approximation does not suffer from high frequency oscillations. The first side maximum in the amplitude spectrum of a high order spline is $\text{sinc}(\frac{3}{2}\pi)^k \approx (-0.21)^k$ smaller than the main maximum. During the approximation of the spline function $D(x, \mathbf{c})$ to

$D^*(x)$ the high frequency part ($\omega > 2\pi/a$) cannot be enlarged without a much larger growth in the amplitude of the low frequency part and thus suppresses the build up of noise in $D(x, \mathbf{c})$.

Another convenient relationship can be deduced from eq. (4). It can be easily proven from this eq. (4) that a spline is actually a convolution of two splines of low order.

$$B^{k+l} = \frac{1}{a} B^k * B^l. \quad (5)$$

Splines of order 8 or higher can therefore be computed faster by convoluting two splines of lower order.

3. The modified Levenberg–Marquardt routine

As an iterative minimizing method we used in this work a modified Levenberg–Marquardt routine (LM) [8–10]. The LM method states that the direction of search for the minimum lies between the Gauss–Newton and the steepest descent method. It combines the speed of the Gauss–Newton and the absolute convergence of the steepest descent method. The routine minimizes the least squares sum $F(\mathbf{x}, \mathbf{c}^{(m)}) = \mathbf{f}^T \mathbf{f}$ in a self-consistent iterative way:

$$\mathbf{f}(\mathbf{x}, \mathbf{c}^{(m)}) = \mathbf{y}(\mathbf{x}, \mathbf{c}^{(m)}) - \mathbf{y}(\mathbf{x})^*, \quad (6)$$

$$\mathbf{c}^{(m+1)} = \mathbf{c}^{(m)} + \boldsymbol{\delta}^{(m)}, \quad (7)$$

$$(\mathbf{J}^T \mathbf{J} + \lambda^{(m)} \mathbf{D}) \boldsymbol{\delta}^{(m)} = -\mathbf{J}^T \mathbf{f}(\mathbf{x}, \mathbf{c}^{(m)}), \quad (8)$$

\mathbf{c}^m = the set of spline coefficients,

$\mathbf{y}(\mathbf{x}, \mathbf{c}^{(m)})$ = trial spectrum,

$\mathbf{y}(\mathbf{x})^*$ = measured spectrum,

\mathbf{J} = Jacobian defined as $J_{ij}(x) = \partial f(x_i, \mathbf{c}) / \partial c_j$,

\mathbf{D} = diagonal matrix of $\mathbf{J}^T \mathbf{J}$,

$\lambda^{(m)}$ = sequence of non-negative real constants.

For $\lambda \ll 1$ the LM iteration behaves like a Gauss–Newton method, and for $\lambda \gg 1$ like a steepest descent method. The magnitude of this constant (λ) is properly chosen by a sequence developed by Fletcher [10]. The parameter (λ) ensures that the LM iteration follows the fast Gauss–Newton method as closely as possible without losing convergence. It is obvious that the

choice of (λ) is of ultimate importance for the performance of this method.

4. Implementation

4.1. Deconvolution

According to the definition of the convolution: $S = D * V$, with V the broadening function and D the unknown spectrum, we can write:

$$\begin{aligned} S(x, c) &= \int_{-\infty}^{+\infty} D(x - \Delta) V(\Delta) d\Delta \\ &= \int_{-\infty}^{+\infty} \sum_i c_i B_i^k(x - \Delta) V(\Delta) d\Delta. \end{aligned} \quad (9)$$

The function $F(x, c)$, which is minimized in the LM method is defined as:

$$F(x, c) = f^T f, \quad f(x, c) = S(x, c) - S^*(x), \quad (10)$$

where $S^*(x)$ is the measured array of datapoints. As input for the LM routine, the array of datapoints composing the spectrum, is defined as a vector. In the LM routine a set of c_i (eq. (2)) is calculated, from which an approximation $D(x, c)$ of $D^*(x)$ can be constructed.

It can be proven that the $\mathbf{J}^T \mathbf{J}$ matrix, which is the numerical heart of our calculation (eq. (8)), is a constant symmetric Toeplitz matrix. In a Toeplitz matrix the elements of the diagonals of the matrix are equal.

Theorem 1: $\mathbf{J}^T \mathbf{J}$ is a constant symmetric Toeplitz matrix which has the form:

$$\begin{bmatrix} \diagup & & & \\ & \diagup & & \\ & & \diagup & \\ & & & \diagup \end{bmatrix}$$

Proof: According to the definition of splines we can write:

$$B_{i+j}(x) = B_i(x + j * a). \quad (11)$$

From eq. (9) it is easily seen that

$$J_i^T(x) = \frac{\partial f(x, c)}{\partial c_i} = B_i * V. \quad (12)$$

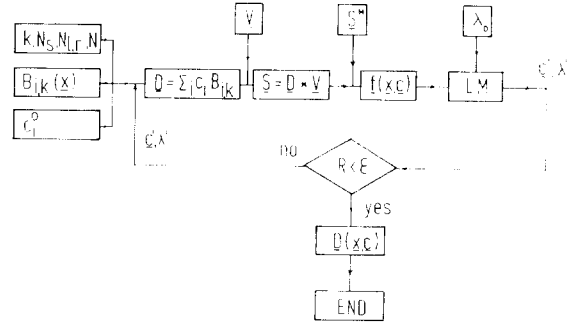


Fig. 2. The iteration scheme of the deconvolution calculation. k = order of the spline; N_s = number of splines; $N_{l,r}$ = number of left and right boundary splines; N = number of intervals; $B_{i,k}(x)$ = spline of order k starting on the i th interval; C_i^0 = initial choice of the spline coefficients C_i ; $f(x, c) = S - S^*$; V = broadening function; $D(x, c)$ = calculated deconvoluted spectrum; R = residue of the fit defined as the square root of $f^T(x, c) \cdot f(x, c)$; ϵ = constant; λ^0 = initial choice of the Levenberg–Marquardt constant, which determines the ratio of the Gauss–Newton and the steepest descent minimizing routines.

Using the definition of convolution we derive

$$\begin{aligned} J_{i+j}^T(x) &= B_{i+j} * V \\ &= \int_{-\infty}^{+\infty} V(y) B_i(x - y - j * a) dy \\ &= J_i^T(x - j * a). \end{aligned} \quad (13)$$

Taking the inner product to derive element $i, i + j$

$$J_i^T J_{i+j} = J_i^T \cdot J_{i+j}^T = J_i^T \cdot J_i^T(x - j * a), \quad (14)$$

$\mathbf{J}^T \mathbf{J}$ is a constant symmetric Toeplitz matrix.

This property is responsible for speeding up the execution of the computer program and for handling larger arrays of datapoints. Only N elements of the matrix $\mathbf{J}^T \mathbf{J}$ have to be evaluated if N is the dimension of this matrix instead of $N \times N$ elements.

If the matrix $\mathbf{J}^T \mathbf{J}$ in (eq. (8)) had $N \times N$ independent matrix elements we would have to use a Gauss elimination procedure to solve it. On a 64 K RAM personal computer this means that the largest dimension of the measured data array S was 200 points with 25 coefficients c_i . With the

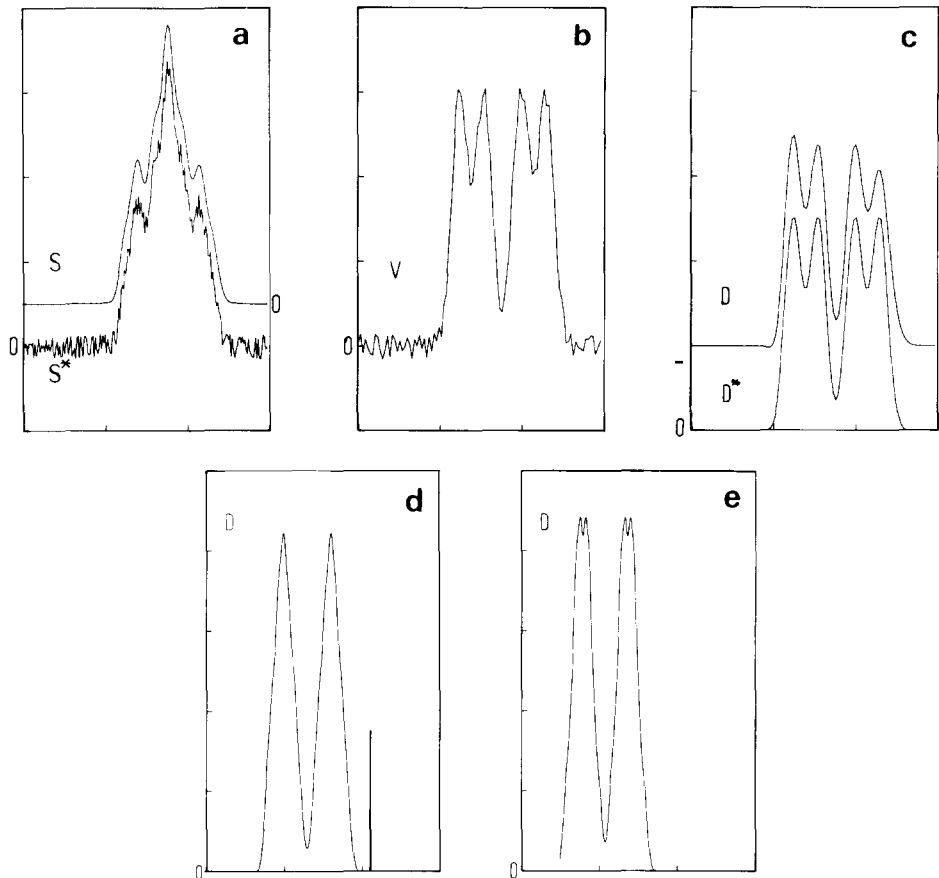


Fig. 3. Test of the (self-)deconvolution method with artificial data. (a) Top: $S(E)$, convolution of the deconvolved result $D(E)$ (top curve (c)) of spectrum $S(E)$ with the broadening function $V(E)$, bottom: $S^*(E)$ input signal of the deconvolution routine ($D^*(E) * D^*(E)$ with 10% noise added). The residue of the approximation between S and S^* is 3.8×10^{-4} per point (200 datapoints). Some characteristic parameters used in our deconvolution scheme: $k = 3$, initial $c_i = 0.2$, $\lambda_0 = 100$, $N_s = 27$, $N = 30$, $N_{i,r} = 0$; (b) broadening function $V(E)$ ($D^*(E)$ with 10% noise added); (c) top: $D(E)$, result of the deconvolution calculation, bottom: $D^*(E)$ original function. Deconvolution with the Fast Fourier Transform routine of Press et al. [11] with noiseless functions S and V ; (d) deconvolution with 256 points, 56 zero's added; (e) deconvolution with 1024 points, 824 zero's added. Only parts of the result are given in (d) and (e) as a large part is zero.

implementation of the Toeplitz inversion method [8], we were able to achieve an input length for the data arrays S and V of 2000 points in combination with 300 coefficients c_i in our program.

The scheme of our deconvolution routine is displayed in fig. 2. The implemented computer program for deconvolution was tested with artificial noise containing data. Full agreement between the outcome of the deconvolution calculation and the initial artificial data was achieved. The result, depicted in fig. 3c, was obtained after seven itera-

tions. Shown in figs. 3d and e are the results of a deconvolution with a Fast Fourier Transform based routine as described by Press et al. [11]. The input for this routine was taken without noise. It is clear from this part of the figure that even without noise this routine is not free of artifacts. The data array had to be enlarged to obtain reasonable results. We used 200 points in our deconvolution routine, which resulted in the same computation time compared to the best obtained deconvolution result with the FFT based method.

Another common used method for deconvolution is the Van Cittert method based on the Fredholm integral [13]. The application to Auger Electron Spectroscopy is fully described by Madden and Houston [14]. They formulated criterions for a reasonable result with this method. As our input data do not fulfil these criterions it is not expected that this method will give a reasonable result.

4.2. Differentiation

The differentiation routine is rather simple. In fig. 4 a test of the differentiation routine on a

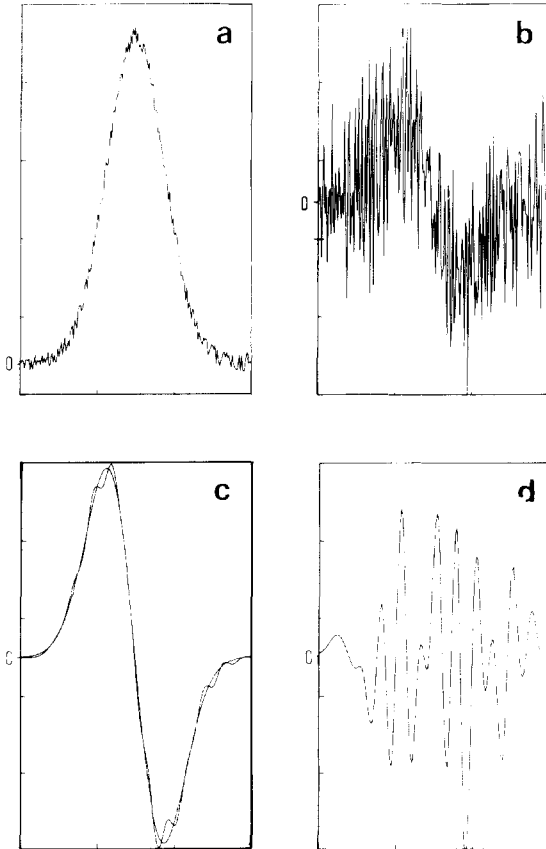


Fig. 4. Test of the differentiation routine. (a) Normalized Gaussian of 200 points containing noise; (b) result of differentiation using the method of Abramowitz and Stegun [15]; (c) result of our method after seven iterations and the analytical result. We used 21 splines of 4 order, process time 15 s; (d) absolute difference of the curve in (c) enlarged by a factor 10.

normalized Gaussian curve containing noise is given. It is performed as an approximation of the original data with splines (eq. (2)), followed by a differentiation as defined in (eq. (3)), fig. 4c. For illustration the same curve was differentiated with a routine based on a method of Abramowitz and Stegun [15] and depicted in fig. 4b.

5. Continuity in deconvolution

Sasse et al. [8] have described a self-deconvolution method $S = D * D$ based on splines. The problem of self-deconvolution has been solved with the aid of Fourier Transforms and Laplace Transforms [16–18], in which the main problem was the reconstruction of the phase. We want to show that even by solving this problem one has no guarantee that the function obtained is physical relevant. We take the more general case of deconvolution $S = V * D$. If we assume that S , V and D can be represented by splines, it can be shown that there is a relation between the spline coefficients of S , V and D , respectively s , v and d .

Theorem 2: When performing a convolution with splines $S = V * D$, with V and D composed of splines of half the order of S , there is a unique relation between the coefficients of the splines.

Proof: Using eq. (5) we write:

$$\begin{aligned} S &= \sum_{l=1}^{2N-1} s_l B_l^{2k} = V * D = \sum_{i,j=1}^N v_i d_j B_i^k * B_j^k \\ &= a \sum_{i,j=1}^N v_i d_j B_{i+j-1}^{2k}. \end{aligned} \quad (15)$$

There exist a relation between the coefficients s , v and d :

$$s_l = a \sum_{\substack{i,j=1,\dots,N \\ l=i+j}} v_i d_j, \quad l = 1, \dots, 2N-1, \quad (16)$$

Only those functions S which are able to satisfy eq. (16) can give solutions in the deconvolution

problem. It shows that the deconvolution problem is actually an overdetermined problem. N coefficients s_l already determine all coefficients d_j and there is therefore a unique relation between the first N coefficients s_l and the other $N-1$ s_l , independent of the choice of the N coefficients s_l out of the complete set s_l . This internal relation of spline representable functions is not satisfied in Fourier Transform based deconvolution methods. In Fourier Transform based deconvolution always a solution exists:

$$\mathbb{D}(\omega) = \mathbb{S}(\omega)/\mathbb{V}(\omega). \quad (17)$$

Solution of eq. (17), however, can be discontinuous and therefore physical irrelevant. In a discrete evaluation continuous and discontinuous solutions cannot be distinguished. If a solution is obtained with the spline method the continuity is assured and therefore more reliable for physical interpretation. When the solution is discontinuous our method will not be able to give a good approximation of the measured spectrum S^* and will give a continuous result as close as possible.

6. Fourier transform

In discrete Fourier Transforms there is a relation between the length and number of points in time space and the maximum frequency and the number of points in the frequency space [11]. Attention has to be given to the sampling of the signal or the addition of zero's to obtain a correct result. The addition of zero's when working with large data arrays may cause large computing times. The outcome of the mathematical exact Fourier transformation is distorted when this transformation is calculated on a computer due to the finite discrete integration of the Fourier integral. We suggest a numerical improvement by implementing splines into the FT algorithm.

The spline is an even real function. The Fourier Transform theory states that this implies that the frequency spectrum only has a real part, the phase is zero. The Fourier Transform of a spline represented function $D(x, c)$, $\mathbb{D}(\omega, c)$, can therefore

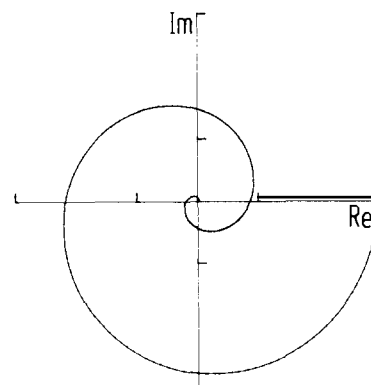


Fig. 5. Polar plot of the spline based Fourier Transform of a Gaussian function.

be calculated with the aid of the time shift theorem:

$$\mathbb{D}(\omega) = \sum_{j=1}^N c_j \mathbb{B}^k(\omega) e^{-i\omega x_j}, \quad (18)$$

x_j is the distance from the center of coordinates to the maximum of spline j . With the aid of eq. (4) the summation can be performed.

As an example of the spline based Fourier Transform the transform of a Gaussian function is depicted in fig. 5. It was performed with 50 splines of order 6. The choice of the number of splines and its order can be varied over a large interval without any influence on the result. In fig. 6 the Fourier transform of the function $\text{sinc}^2(x - x_0) \cos(x - x_0)$ is given in a polar representation for our method (1000 points) and obtained by Fast Fourier transform for 1024 and 4096 points. The theoretical result for the amplitude is a shifted triangle, with height π and a straight line for the phase. In fig. 7 the Fourier transform of the function $\text{sinc}(x - x_0) \cos(x - x_0)$ (2000 points) is given in polar representation for our method (1000 points) and FFT (2048 and 4096 points). Theoretical this should be a circle, but due to the windowed input this never can be obtained. The improvement of the scheme of Fourier transformation is depicted in figs. 6 and 7. The frequency spacing is chosen arbitrarily. The CPU time needed in our method is approximately 1 min for 1000 points. This is slow in respect to Fast Fourier Transform algorithms, which are able to perform this in

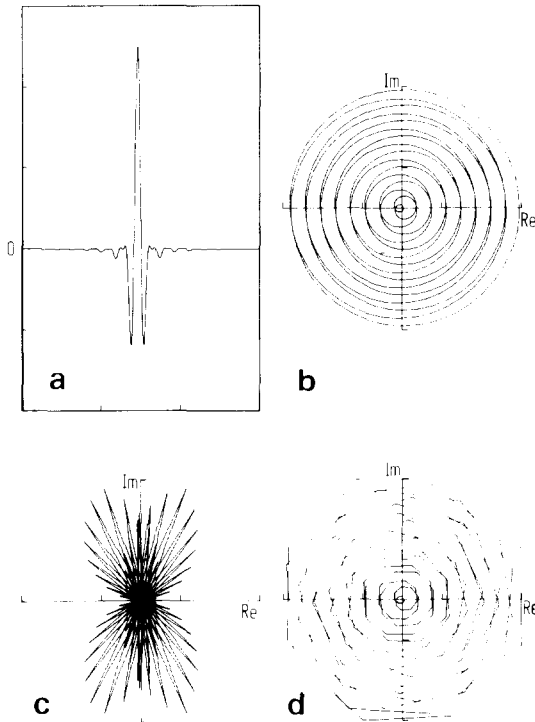


Fig. 6. Polar plots of the Fourier transform of $\text{sinc}^2(x - x_0) \cos(x - x_0)$. (a) Original function of 100 points; (b) Fourier Transform with our method, 1000 points 121 splines of order 4; (c) FFT on 1024 points (24 zero's added); (d) FFT of 4096 points (3096 zero's added).

seconds. However, to obtain the same frequency distribution with such algorithms, one would have to take a 50 time larger input array. This means a great reduction in the efficiency of such algorithms as one has also to consider the I/O time. The time

for a 4096 points containing data array is comparable to the time needed for our method. We have performed these test without noise. As long as our method can make a good approximation it will give a good performance. The introduction of noise would harm the FFT routines more than our new FT algorithm, because of the global character of the splines a kind of inherent windowing will take place [9] (see for example fig. 3).

7. Conclusion

In this paper we have presented computer algorithms for deconvolution, differentiation and Fourier Transform of data. The algorithms use splines in combination with a powerful iterative minimizing routine. Since the method has a global character, noise amplification can be suppressed as is evidently demonstrated in the differentiation and deconvolution routine. This allows us to use relatively large fluctuating and noise containing functions as input. Our deconvolution and self-deconvolution routine has been proven to be a promising tool to process noise containing data. We proved the existence of a constant symmetric Toeplitz matrix in our spline approximation routines. This made it possible to run them on a personal computer (64 K) with large sets of data and still have a reasonable computation time.

Deconvolution with Fourier Transform routines cannot guarantee continuity of the solution even if the phase problem is solved as occurring in self-

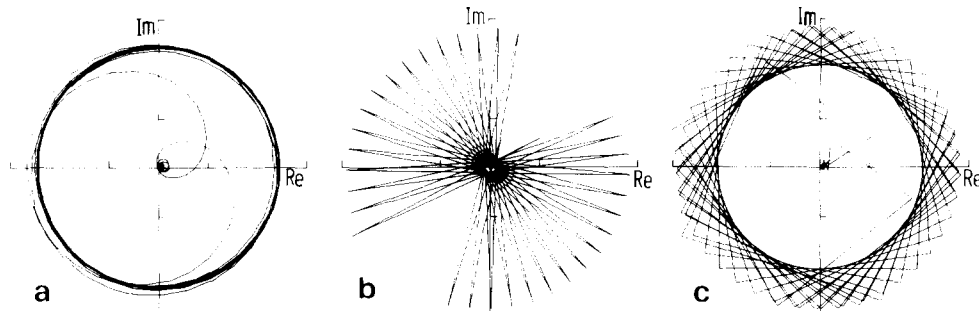


Fig. 7. Polar plots of the Fourier transform of $\text{sinc}(x - x_0) \cos(x - x_0)$ original function has 2000 points. (a) Fourier transform with our method, 1000 points 236 splines of order 4; (b) FFT on 2048 points (48 zero's added); (c) FFT on 4096 points (2096 zero's added).

deconvolution. The physical relevance of the solutions obtained is therefore ambiguous. We proved that our spline based routine will only generate continuous solutions and fails if the solution is discontinuous.

Using the convenient mathematical character of splines we were able to construct a spline based Fourier Transform routine. Although our Fourier Transform routine is slower than common discrete Fourier Transforms when using the same length of data, we are free to choose the frequency spacing. To obtain the same result with FFT algorithms a large addition of zero's would be necessary, which slows down the speed of the FFT routine extensively. Test were performed without noise to stress the strength of our method in even nice cases. The quality of the approximation is the only necessary condition for a good performance of our routine. This can be easily fulfilled for most of the physical relevant data.

Not only deconvolution problems without knowledge of the deconvolution itself can be solved with this method but many problems of the form,

$$D(x) = A^{-1}(S(x)) \quad (19)$$

with A being a well defined operation on an unknown function $D(x)$ and $S(x)$ as a known spectrum. Often it is not easy or even impossible to compute A^{-1} explicitly. This class of problems can be solved in principle by applying the iterative scheme of fig. 2.

References

- [1] M.B. Priestly, *Spectral Analysis and Time Series* (Academic Press, New York, London, 1981).
- [2] J. Cullum, *SIAM J. Numer. Anal.* 8 (1971) 254.
- [3] A.F. Carley and R.W. Joyner, *J. Electr. Spectr. Related Phenom.* 16 (1979) 1.
- [4] B. Chornik, R. Sopizet and C. le Gressus, *J. Electr. Spectr. Related Phenom.* 42 (1987) 392.
- [5] P.M. Prenter, *Splines and Variational Methods* (Wiley, New York, 1975).
- [6] P. Dierckx, *J. Comput. Phys.* 52 (1983) 163.
- [7] A.J. Guttmann, *J. Phys. C* 8 (1975) 4037.
- [8] A.G.B.M. Sasse, D.G. Lakerveld and A. van Silfhout, *J. Vac. Sci. Technol. A* 6 (1988) 1045.
- [9] A.G.B.M. Sasse, H. Wormeester and A. van Silfhout, *Surface Intern. Anal.* (in press).
- [10] R. Fletcher, *A Modified Marquardt Subroutine for Non-linear Least Squares Fitting*, UK Atomic Energy Authority Research Group, Theoretical Physics Division, AERE-R. 6799, Harwell, Berkshire (1971).
- [11] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes* (Cambridge Univ. Press, London, 1986).
- [12] G. Wahba and S. Wold, *Commun. in Stat.* 4 (1975) 1.
- [13] P.H. van Cittert, *Z. Phys.* 69 (1931) 298.
- [14] H.H. Madden and J.E. Houston, *J. Vac. Sci. Technol.* 47 (1976) 3071.
- [15] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1965).
- [16] J.A. Taggle, V. Martinez, J.M. Rojo and M. Salmeron, *Surface Sci.* 79 (1978) 93.
- [17] H.D. Hagstrum and G.E. Becker, *Phys. Rev. B* 4 (1971) 4187.
- [18] H.D. Hagstrum and G.E. Becker, *Phys. Rev. B* 8 (1973) 1580.