Approximating Functions from Sampled Fourier Data Using Spline Pseudofilters

Ana Gabriela Martínez, and Alvaro Rodolfo De Pierro, *Member*

***Abstract*— Recently, new polynomial approximation formulas were proposed for the reconstruction of compactly supported piecewise smooth functions from Fourier data. Formulas for zero and first degree polynomials were presented. For higher degree approximations, polynomial formulas become extremely complicated to be handled. In this paper we solve this problem by introducing spline approximations. The new approach can be used in the same way as the polynomial one but producing computable formulas for any degree of approximation in Fourier reconstruction. We present general error estimates and numerical experiments.**

***Index Terms*— Discrete Fourier transform, filters, Fourier series, Fourier tranform, interpolation**

I. Introduction

Recovering functions from a finite number of Fourier coefficients is an essential task in signal processing and many other areas of application as well like Nuclear Magnetic Ressonance, Spectral Methods in Fluid Dynamics and Computed Tomography (see for example [1], [4], [9], [12], [15]). It is well known that when the function is smooth (analytic), the pointwise convergence of the Fourier series is exponential. Otherwise, it is only polynomial or even worse, and it depends on its degree of smoothness. Furthermore, if the function is not continuous, convergence is extremely slow and large oscillations show up. That is the so called Gibbs phenomenon [11] that causes a very poor approximation close to the jumps. For the resolution of the Gibbs phenomenon there are two groups of methods: one consists of filtering in the Fourier domain (equivalent to smoothing in the space domain) and the other of approximating the function in the space domain for each interval between discontinuities, using some special basis. The first group is able to obtain only polynomial convergence of the filtered Fourier expansion, with a degree of acceleration depending on the degree of smoothness of the filter. This convergence, as expected, deteriorates close to the discontinuities and the polynomial bounds for the truncation error can only be obtained outside small intervals containing the discontinuities (see [11]). The second group, before 1999, contained only one method and it consists of expanding the Fourier approximation using Gegenbauer polynomials, thus obtaining exponential convergence (truncation error tending to zero exponentially fast with the number of available coefficients). A detailed description of all these methods can be found in the Review by Gottlieb and Shu [11]. The Gegenbauer approximation approach resolves in theory the Gibbs phenomenon, because of the exponential convergence proofs, but in practice, it is computationally expensive and unstable, suggesting the use of hybrid approaches that combines it with classical filtering (see [6]), still complicated and not very stable. Also, the Gegenbauer expansion approach needs to discard intervals containing the discontinuities, in order to get the exponential bounds. In 1999, Yin, De Pierro and Wei [24] introduced a new approach that essentially belongs to the second group and could be described as follows: estimate first the discontinuity points and approximate the function inside the intervals between discontinuities by polynomials of a given degree in such a way that the reconstruction is exact for functions that are piecewise polynomials of that degree. This polynomial approach proved to be more stable and better than the classical filtering approach, as shown by the experiments in [24]. Formulas and error estimates for zeroth and first degree approximations were presented in [24] and [21]. General formulas for higher degrees (≥ 2) proved to be extremely complicated and computationally unfeasible. It is easy to see this, just by trying to extend from zeroth to first degree (that is not trivial) and then to second degree (that gives rise to a several pages formula). In this paper, we solve that problem by introducing a new family of approximation methods based on splines. This new approach makes possible to deduce approximation formulas for the retrieval of functions from Fourier coefficients for any given degree allowing very accurate reconstructions of piecewise smooth signals from Fourier data. We illustrate this for the case of second degree splines. Higher degree cases, although more complicated, could be now deduced straightaway. Regarding the previous step, that is, the estimation of the discontinuities, fast methods for this task were based on modifications of the conjugate Fourier series, work done by A. Gelb and E. Tadmor [7], [8]. A more stable and fast method based on polynomial approximations can be found in [22] (Section III, Theorem 1). In [23] it was shown that the zeroth degree detection method of [22] works extremely well and it is much better (better approximations, separating better the jumps, more stable) than the existing ones based on the conjugate Fourier series.

In what follows we introduce some assumptions and notation, needed for the next sections. For the sake of simplicity we consider functions in the interval $[0, 1]$.

The Fourier Transform (FT) $\hat{f}$ of $f(x)$ is defined as

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi x\omega} dx = \int_{0}^{1} f(x) e^{-i2\pi x\omega} dx \qquad (1)$$

and its inverse (IFT) as

Authors are with the Applied Mathematics Department, State University of Campinas, IMECC-UNICAMP, CP 6065, 13081-970, Campinas, Brazil. Phone 55-19-32515962, Fax 55-19-32515766. E-mails: gabrik@ime.unicamp.br (A. G. Martínez) and alvaro@ime.unicamp.br (A. R. De Pierro).

$$f(x) = \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i2\pi x\omega} d\omega, \ a.e. \tag{2}$$

Let

$$0 = x_0 < x_1 < \cdots < x_N = 1, \ \Delta x_j = x_{j+1} - x_j = h = \frac{1}{N} \tag{3}$$

and

$$f_j = f(\eta_j), \ j = 0, \ldots, N-1 \text{ with } \eta_j = x_j + \delta, \tag{4}$$

where $0 \leq \delta \leq \ h$ is a constant.

We need to derive an approximate relation between the discrete Fourier transform (DFT) and the Fourier transform of the function $f(x)$, so that we can use the Fast Fourier Transform (FFT) to evaluate $\{f_j, j = 0, \cdots, N-1\}$ efficiently and accurately. Next we present our definitions and notation for the Discrete Fourier Transform, the Fourier Transform (FT) and the Fourier Expansion (FE). Properties and relations between them can be found in any textbook like [3].

For a given function $f(x)$ with support in $[0, 1]$ and an even integer $N > 0$, let $\{x_j, \ j = 0, \cdots, N\}$ and $\{f_j, \ j = 0, \cdots, N-1\}$ be defined as in (3) and (4), respectively. Then the DFT of $\{f_j, \ j = 0, \cdots, N-1\}$ is defined by

$$\tilde{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i2k\pi \frac{j}{N}}, -N/2 \leq k \leq N/2 - 1 \tag{5}$$

and the inverse formula by:

$$f_j = \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{i2k\pi \frac{j}{N}}, j = 0, \ldots, N-1. \tag{6}$$

The DFT is the mapping between the N complex numbers $\{f_j, j = 0, \ldots, N-1\}$ and the N complex numbers $\{\tilde{f}_k, k = -N/2, \ldots, N/2-1\}$. The FFT can be used to compute them.

From (5) and (6), in order to establish the relation between the DFT and the FT of $f(x)$, we just need to establish the relation between the DFT and the FT with frequency k. From (1) we obtain

$$\hat{f}_k \equiv \hat{f}(k) = \int_0^1 f(x) e^{-i2k\pi x} dx, \ k = 0, \pm 1, \pm 2, \ldots. \tag{7}$$

If f is a real valued function, $\hat{f}_{-k} = \overline{\hat{f}_k}$.

The set of functions $\{e^{i2k\pi x}, k = 0, \pm 1, \pm 2, \ldots\}$ is an orthogonal system over the interval [0,1]. Because of the fact that the support of $f(x)$ is in the interval $[0, 1]$, we can also consider $f(x)$ as defined in [0,1], and we can obtain the Fourier expansion of $f(x)$:

$$Sf(x) = \sum_{k=-\infty}^{+\infty} c_k e^{i2k\pi x} \tag{8}$$

with the Fourier coefficients

$$c_k = \int_0^1 f(x) e^{-i2k\pi x} dx, \ k = 0, \pm 1, \pm 2, \ldots. \tag{9}$$

$Sf(x)$ represents the formal expansion of f in terms of the Fourier orthogonal system $\{e^{i2k\pi x}, k = 0, \pm 1, \pm 2, \ldots\}$. If f is a real valued function, $c_{-k} = \overline{c_k}$.

Notice that the Fourier coefficient c_k in (9) is exactly the same as $\hat{f}_k$, that is, the Fourier transform of $f(x)$ with frequency k in (7).

The truncated Fourier expansion of $f(x)$ is

$$P_N f(x) = \sum_{k=-N/2}^{N/2-1} c_k e^{i2k\pi x} \tag{10}$$

Equation (10) is different from the theoretical discussion of truncated Fourier transforms, but it corresponds directly to the way that practical computation is actually programmed.

In order to make this expansion rigorous, one has to cope with some problems: when and in what sense is the transform convergent, what is the relation between the transform and the function $f(x)$, and how rapidly does the series converge. It is well known that if $f(x) \in L_2([0, 1])$ (square integrable in $(0, 1)$), the series converges to the function f. However, pointwise convergence could be far from reasonable. If $f(x) \in C^\infty(0, 1)$ and $f^{(p)}(0) = f^{(p)}(1)$ for all $p = 0, 1 \cdots$, then

$$P_N f(x) \to f(x) \text{ exponentially for } N \to \infty, \ \forall x \in [0, 1].$$

But when $f(x)$ has points of discontinuity, or even when $f(x) \in C^\infty(0, 1)$ but it is not periodic, convergence rate is poor and the Gibbs phenomenon occurs. As mentioned in the beginning, in this case, one needs to use window functions (also called smoothing functions) to reduce oscillations close to points of discontinuity [14], [9], [4]. In [10] a different but expensive approach is suggested for the problem. In [16] a comprehensive description of all these approaches can be found. In [24], a family of filters (that we prefer now to call pseudofilters, because they are no longer associated with convolutions in the space domain) was introduced based on the property of producing an exact reconstruction for piecewise constant functions. The formulas extending this result to piecewise linear functions appeared in [21]. Further extensions of the formulas for filters associated with higher degree polynomials proved to be too complicated, even for degree two. The solution of this problem turns out to be the substitution of polynomials by splines, giving rise to computationally feasible formulas for the filters of any degree of approximation, as it is mathematically shown in the Appendix, where error estimates are deduced.

In §2 we present a brief introduction to splines, needed to derive the new spline pseudofilters of §3. §4 illustrates with numerical experiments the behavior of the filters and finally in §5 we present some conclusions and our current and future research directions on this topic.

II. A Brief Introduction to Splines

It follows a brief presentation of some basic results and notation on spline functions. More detailed information could be found in [2] and [18].

A. B-Splines

B-splines are symmetric 'bell-shaped' functions, built by iteratively convolving the rectangular pulse, that is,

$$\beta^n(x) = \beta^0 * ... * \beta^0(x), \qquad n+1-times,$$

where

$$\beta^0(x) = \begin{cases} 1 & -\frac{1}{2} < x < \frac{1}{2}, \\ 0 & \text{otherwise} . \end{cases}$$

These symmetric splines of order n, $\{\beta^n(x)\}_{n\in N}$, also known as B-splines are basis functions used to build the spline functions $s(x)$. B-splines can be manipulated (differentiated and integrated) very easily. They are compact supported and the simplest way to construct them explicitly is through the following formula:

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \begin{pmatrix} n+1 \\ j \end{pmatrix} (x+\bar{n}-j)^2 \mu(x+\bar{n}-j),$$

where $\bar{n} = \frac{n+1}{2}$ and

$$\mu(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$$

The Fourier Transform of the $\beta^n(x)$'s can be computed using the convolution theorem:

$$\beta^n(x) = \beta^0 *...* \beta^0(x),\ n+1-times \ \Rightarrow\ \hat{\beta}^n(w) = (\hat{\beta}^0(w))^{n+1}.$$

From the fact that $\hat{\beta}^0(w) = \frac{e^{-iw/2} - e^{iw/2}}{-iw}$, we get that $\hat{\beta}^n(w) = (\frac{\sin(w/2)}{w/2})^{n+1}$.

B. Interpolating Splines

Spline functions are piecewise polynomials connected in a smooth way at the points x_k's, called knots . A degree "n" spline corresponds to a degree n polynomial in each of the segments determined by the knots. In the interpolation process, the differentiability conditions imposed to the approximation allow the calculation of the $n+1$ coefficients for the polynomials in each segment. So, we have the following:

Definition: Let $\{x_i\}$ be a decreasing or increasing set of real numbers, where $a = \inf(\{x_i\})$ and $b = \sup(\{x_i\})$. If n is an integer ≥ 2, $s(x)$ is a spline function of order n or degree $n-1$ with knots $\{x_i\}$, if:

(i) $s(x)|_{[x_i,x_{i+1}]} \in P_{n-1}$, i.e., $s(x)$ constrained to the interval $[x_i, x_{i+1}]$ is a polynomial of at most degree $n-1$.

(ii) $s(x) \in C^{n-2}(a,b)$.

(iii) Additionally, if the number of knots is finite and $s^{(k)}(a) = s^{(k)}(b), k = 0,1,\ldots,n-2,$ $s(x)$ is called a periodic spline.

If $g(x)$ is a function defined in the interval $[a,b]$, $s(x)$ will be its interpolating spline if it satisfies $s(x_i) = g(x_i), \forall i$.

We will call S_h^n, the generic space of polynomial splines of order n, where n stands for the degree of the polynomials in each segment and h the spacing between knots, that is,

$$S_h^n = \{s \in L_2(R); s \in C^{n-1},\ s|_{I_k} \in P_n, k \in Z\},$$

where $I_k = [x_k, x_k+h)$ if n is odd, and $I_k = [x_k - h/2, x_k + h/2)$ if n is even.

Interpolating Splines are uniquely determined by their B-splines expansion; in the case of equally spaced knots, this expansion is given by

$$s(x) = \sum_{k\in Z} c(k)\beta^n(x-k),$$

with unique coefficients $c(k)$. In the case of polynomials of degree less or equal to one, the relation between the sample points and the coefficients is simple and given by $c(k) = s(k)$. For $n \geq 2$, it was introduced in [17] an efficient technique using digital filtering for the computation of the coefficients. In order to describe this technique we need some additional notation. The *discrete B-spline kernel* of degree n, expanded by a factor m is defined by a sequence of values that correspond to equidistant samples of the B-spline of degree n, expanded by a factor of m, that is

$$b_m^n(k) = \beta^n(x/m)|_{x=k}.$$

Once defined $b_m^n(k)$, we denote by $B_m^n(z)$ its z transform; so

$$B_m^n(z) = \sum_{k\in Z} b_m^n(k)\ z^{-k}.$$

The interpolation condition for the approximation by splines at integer values corresponds to:

$$s(k) = \sum_{l\in Z} c(l)\beta^n(x-l).$$

that can be rewritten as

$$s(k) = (c * b_1^n)(k).$$

We denote by "$(b_1^n)^{-1}$" the inverse convolution operator defined through the z transform:

$$(b_1^n)^{-1} \rightarrow 1/B_1^n(z),$$

that is, $(b_1^n)^{-1}$ is the one whose z transform corresponds to the inverse of $B_1^n(z)$. In the z space, we have that $S(z) = C(z)\ B_1^n(z)$, applying the convolution theorem. Therefore, $C(z) = S(z)\ [\frac{1}{B_1^n(z)}]$ and we obtain the following expression for the coefficients,

$$c(k) = [s * (b_1^n)^{-1}](k).$$

It is worth mention that the previous procedure is not anly stable and fast, but also very easy to implement [17].

Substituting the last expression in $s(x)$, we obtain,

$$s(x) = \sum_{k\in\mathbf{Z}} \ [s*(b_1^n)^{-1}](k)\ \beta^n(x-k),$$

or, equivalently,

$$s(x) = \sum_{k\in\mathbf{Z}} \ s(k)\ \eta^n(x-k),$$

where $\eta^n(x) = [(b_1^n)^{-1} * \beta^n](x)$ is called the *Cardinal Spline* of order n.

III. New Spline based Pseudofilters

Our goal is to derive recontruction methods of the point values of f from a finite number of Fourier coefficients $\{\hat{f}_k\}_{k=-N/2}^{N/2-1}$. In our presentation of the new spline filters, for the sake of simplicity, we will assume first that f has a single jump at $x = z \in [0,1]$.

We start describing the zeroth degree method, then the first degree method and finally the general formula for splines filters of any given degree. The reason for this is that this sequence could help the reader to understand how it is possible to build algorithms for any degree of approximation (accuracy). Algorithm 1, 2 and 3 give exact reconstructions for piecewise constant functions, piecewise linear splines and piecewise splines of second degree respectively.Also, for the sake of better understanding, the descriptions of the cases when the discontinuity point belongs to the mesh and when it does not are presented separately.

We denote by β_h^n the symmetric spline of order n defined by,

$$\beta_h^n(x) = \frac{1}{h}\beta^n(\frac{x}{h}).$$

Its Fourier transform is given by :

$$\hat{\beta_h}^n(w) = [2\ \sin(w\frac{h}{2})/wh]^{n+1}.$$

A. Zeroth Degree Approximation

Uniform Mesh

Let $\eta_i = \frac{x_i + x_{i+1}}{2}$ and $f_i = f(\eta_i)$ for $i = 1, \ldots, N-1$. We define the approximation by a zeroth degree polynomial of the function f in $[0,1]$ as:

$$p_f^{(0)}(x) = h \sum_{j=0}^{N-1} f(\eta_j)\ \beta_h^0(x - \eta_j).$$

So, in each interval $[x_i, x_{i+1}]$, the value of the polynomial is constant and equal to the function's value at the mid point of the interval. The Fourier coefficients of this polynomial are given by

$$\begin{aligned}(\hat{p}_f^{(0)})_k &= h \sum_{j=0}^{N-1} f(\eta_j)\ (\hat{\beta_h}^0)_k\ e^{-i2\pi k(x_j+h/2)},\\ &= \hat{\beta}^0(kh)\ e^{-i\pi kh}\ h \sum_{j=0}^{N-1} f(\eta_j)\ e^{-i2\pi k x_j},\\ &= [\frac{\sin(\pi kh)}{\pi kh}]\ e^{-i\pi kh}\ \tilde{f}_k.\end{aligned}$$

When the discontinuity points of f are the same as the knot points, then the values of $\hat{f}_k$ can be approximated by the Fourier coefficients of the interpolating polynomial of zeroth degree. So, if the g_j's are the approximations of the exact values of the function f, we have that:

$$\hat{f}_k \approx [\frac{\sin(\pi kh)}{\pi kh}]\ e^{-i\pi kh}\ \tilde{g}_k.$$

Therefore the approximations for the values of f at the points η_j can be calculated by applying an Inverse Discrete Fourier Transform (IDFT):

$$g_j = (IDFT)\{[\frac{\pi kh}{\sin(\pi kh)}]\ e^{i\pi kh}\ \hat{f}_k\}_j \quad j = 0, 1, \ldots, N-1.$$

This gives the expression of the *zeroth degree filter* for the reconstruction formula:

$$\sigma_k^{(0)} = [\frac{\pi kh}{\sin(\pi kh)}]\ e^{i\pi kh},\ k \neq 0 \quad and \quad \sigma_0^{(0)} = 1. \qquad (11)$$

So, the formula for the reconstruction in the Fourier space for the discontinuity points that coincide with the mesh points is given by:

$$\tilde{g}_k = \sigma_k^{(0)}\ \hat{f}_k, \qquad (12)$$

and the pointwise reconstruction of f could be obtained through an Inverse Fast Fourier Transform (IFFT).

Nonuniform Mesh

When the mesh points do not coincide with the knots, the formula should be corrected with a term that comes from the substitution of the uniform mesh by a nonuniform one that includes the discontinuity point as a knot, as in [24]. In order to obtain the new fromula, let $z \not\subset \{x_0, \ldots, x_N\}$ be the discontinuity point and let x_{q_z} be the closest point to z in the uniform mesh. The new nonuniform mesh is obtained by substituting x_{q_z} by z, i.e.: $\{x_0, \ldots, x_{q_z-1}, z, x_{q_z+1}, \ldots, x_N\}$. In the new resulting subintervals (x_{q_z-1}, z) and (z, x_{q_z+1}) we define

$$f_{q_z-1} = f(\frac{x_{q_z-1}+z}{2}) \quad and \quad f_{q_z} = f(\frac{x_{q_z+1}+z}{2}).$$

The new zeroth degree polynomial approximating the function in the nonuniform mesh is given by

$$p^{(0)}(x) = h \sum_{j=0}^{N-1} f_j\ \beta_h^{(0)}(x - \eta_j) + (f_{q_z-1} - f_{q_z})\ \chi_{[x_{q_z}, z]}.$$

where $\chi_I(x)$ is the characteristic function of the interval I:

$$\chi_I(x) = \begin{cases} 1 & \text{for} \quad x \in I, \\ 0 & \text{otherwise} \ . \end{cases}$$

Taking into account that

$$\begin{aligned} p^{(0)}(x) &= h \sum_{j=0}^{N-1} f_j\ \beta_h^{(0)}(x - \eta_j) \\ &-h f_{q_z-1}\ \beta_h^{(0)}(x - \eta_{q_z-1}) - h f_{q_z} \beta_h^{(0)}(x - \eta_{q_z}) + \\ &+ f_{q_z-1}\ \chi_{[x_{q_z-1}, z]} + f_{q_z}\ \chi_{[z, x_{q_z+1}]}, \end{aligned}$$

in the Fourier space we get

$$\hat{f}_k \approx \hat{p}_k^{(0)} = \hat{\beta}_{kh}^{(0)}\ e^{-i\pi kh}\ \tilde{f}_k + (f_{q_z-1} - f_{q_z})\ \hat{A}_k^{(0)}(z),$$

where $\hat{A}_k^{(0)}(z) = (\hat{\chi}_{[x_{q_z}, z]})_k = \int_{x_{q_z}}^{z} e^{-i2\pi kx} dx.$

As before, the reconstructed point values are denoted by g_j and the reconstruction formula in the Fourier space is given by:

$$\tilde{g}_k = \sigma_k^{(0)} \ (\hat{f}_k + [g](z) \ \hat{A}_k^{(0)}(z)), \tag{13}$$

where $[g](z) = (g_{q_z} - g_{q_z-1})$ denotes the approximation of the jump amplitude of f at the point z. The values of $\{g_j\}_{j=0}^{N-1}$ are obtained through the IDFT:

$$g_j = IDFT\{\sigma_k^{(0)} \ (\hat{f}_k + [g](z) \ \hat{A}_k^{(0)}(z))\}_j,$$

The generalization of the reconstruction formula for L discontinuity points is straightforward. These L points substitute the closest ones in the mesh and the corresponding expression in the Fourier space is given by:

$$\tilde{g}_k = \sigma_k^{(0)} \ (\hat{f}_k + \sum_{l=1}^{L} [g](z_l) \ \hat{A}_k^{(0)}(z_l)),$$

So, the reconstruction formula for L points is the following:

$$g_j = f_j^{(0)} + \sum_{l=1}^{L} [g](z_l) \ a_{j,l}^{(0)}, \quad j = 0, 1, \ldots, N-1, \tag{14}$$

where $f_j^{(0)} = IDFT\{\sigma_k^{(0)} \ \hat{f}_k\}_j$ and $a_{j,l}^{(0)} = IDFT\{\sigma_k^{(0)} \ \hat{A}_k^{(0)}(z_l)\}_j$.

From (14), it is clear that, to obtain approximations for the point values of the function, it is necessary the knowledge of the discontinuities $\{z_l\}_{l=1}^{L}$ as well as their amplitudes. Methods for this computation can be found in [22]. Observe that the $\{[g](z_l)\}_{l=1}^{L}$ can be obtained by solving an $L \times L$ linear system, because if we consider in (14) $j = q_r$ and $j = q_r - 1$, we have that:

$$g_{q_r} = f_{q_r}^{(0)} + \sum_{l=1}^{L} [g](z_l) \ a_{q_r,l}^{(0)},$$

$$g_{q_r-1} = f_{q_r-1}^{(0)} + \sum_{l=1}^{L} [g](z_l) \ a_{q_r-1,l}^{(0)}.$$

And for $r = 1, 2, \ldots, L$,

$$[g](z_r) - \sum_{l=1}^{L} (a_{q_r,l}^{(0)} - a_{q_r-1,l}^{(0)}) \ [g](z_l) = f_{q_r}^{(0)} - f_{q_r-1}^{(0)}.$$

These equations define a linear system for the L unknowns $[g](z_1), \ldots, [g](z_L)$.
Summarizing, we describe next the zeroth degree reconstruction method for the nonuniform mesh in the following algorithm (ifft will denote the Inverse Fast Fourier Transform).

Algorithm 1
Given N Fourier coefficients $\{\hat{f}_k\}_{k=-N/2}^{N/2-1}$ and L discontinuity points $\{z_1, \ldots, z_L\}$,
Step 1: Compute

$$f_j^{(0)} = ifft_j\{\sigma_k^{(0)} \ \hat{f}_k\}, \ for \ j = 0, \ldots, N-1,$$

$$a_{j,l}^{(0)} = ifft_j\{\sigma_k^{(0)} \ A_k^{(0)}(z_l)\}, \ for \ j = 0, \ldots, N-1, \ l = 1, \ldots, L.$$

Step 2: Solve for $\{[g](z_l)\}_{l=1}^{L}$ the linear system,

$$[g](z_r) + \sum_{l=1}^{L} (a_{q_r,l}^{(0)} - a_{q_r-1,l}^{(0)}) \ [g](z_l) = f_{q_r}^{(0)} - f_{q_r-1}^{(0)},$$

for $r = 1, \ldots, L$.

Step 3: Compute,

$$g_j = f_j^{(0)} + \sum_{l=1}^{L} [g](z_l) \ a_{j,l}^{(0)}, \ for \ j = 0, \ldots, N-1.$$

The computational cost of the algorithm is low because only $N(2L+1)$ complex multiplications are necessary plus $L+1$ applications of the FFT.

We derive next the first degree filter and the corresponding reconstruction formula.

B. First Degree Approximation

Let $x_j = \frac{j}{N}$, and $f_j^+ = f(x_j^+) = \lim_{x \to x_j^+} f(x)$, $f_j^- = f(x_j^-) = \lim_{x \to x_j^-} f(x)$. $\{x_{q_l}\}_{l=1}^{L}$ are the L points of the mesh closest to the discontinuity points $\{z_l\}_{l=1}^{L}$.

Using the saw function, we introduce for each z_l a new piecewise smooth function with support localized in the interval (x_{q_l-1}, x_{q_l+1}) that we denote by $A_{z_l}^1(x)$ and has a unique discontinuity point at $x = z_l$.

$$A_{z_l}^1(x) = \begin{cases} (x_{q_l-1} - x)/2h & \text{for } x \in [x_{q_l-1}, z_l], \\ (x_{q_l+1} - x)/2h & \text{for } x \in (z_l, x_{q_l+1}]. \\ 0 & \text{otherwise.} \end{cases}$$

The value of the jump amplitude at that point is given by $[A_{z_l}^1](z_l) = A_{z_l}^1(z_l^+) - A_{z_l}^1(z_l^-) = 1$, and the k-th corresponding Fourier coefficient by ,

$$(\hat{A_{z_l}^1})_k = \frac{e^{-i2\pi k z_l}}{i2\pi k} + \frac{e^{-i2\pi k x_{q_l}}}{i2\pi k} \ (\frac{\sin(2\pi k h)}{2\pi k h}).$$

This is a very useful function because it allows the construction of another continuous function $u(x)$ associated with $f(x)$. That is,

$$f(x) = u(x) + \sum_{l=1}^{L} [f](z_l) \ A_{z_l}^1(x).$$

The function $u(x)$ is not only continuous, but as regular as f in the intervals not containing discontinuity points, that is, for $x \in [0,1] - \bigcup_{z \in Z} [x_{q_z-1}, x_{q_z+1}]$. It is continuous because

$$[u] \ (z_r) = [f](z_r) - \sum_{l=1}^{L} [f](z_l) \ [A_{z_l}^1](z_r)$$

$$= [f](z_r) - \sum_{l=1}^{L} [f](z_l) \ \delta_{l,r} = [f](z_r) - [f](z_r) = 0.$$

Now denote by $p_u^{n+}(x)$ the degree n interpolator of the function $u(x)$ at the points $(x_j, u^+(x_j))$, where $u^+(x_j) = f_j^+ - \sum_{l=1}^{L} [f](z_l) \ A_{z_l}^1(x_j)$.

Observe that for each l, $u(z_l^+) = u(z_l^-) = \frac{[f(z_l^+)(z-x_{q_l-1})+f(z_l^-)(x_{q_l+1}-z)]}{2h}$.
The representation of the polynomial $p_u^{n+}(x)$ in the B-spline basis, for $n=1$ corresponds to:

$$p_u^{1+}(x) = h \sum_{j=0}^{N-1} u(x_j)\ \beta_h^1(x-x_j).$$

Replacing the values $u(x_j)$ in the expression above gives,

$$p_u^{1+}(x) = h \sum_{j=0}^{N-1} f_j^+\ \beta^1(x-x_j) - \sum_{l=1}^{L} [f](z_l)$$
$$h\ \textstyle\sum_{j=0}^{N-1}\ A_{z_l}^1(x_j)\ \beta_h^1(x-x_j),$$

The first part of the above expression will be denoted by $p^1{}_{f^+}(x)$, which corresponds to the first degree interpolating spline at the points $(x_j, f^+(x_j))$. So, the k-th Fourier coefficient is given by:

$$(\hat{p}_u^{1+})_k = (\hat{p}_{f^+}^1)_k - \sum_{l=1}^{L} [f](z_l) \sum_{j=0}^{N-1} A_{z_l}^1(x_j)\ \hat{\beta}_{kh}^1\ e^{-i2\pi k x_j}$$
$$= h \sum_{j=0}^{N-1} f_j^+ \hat{\beta}_{kh}^1\ e^{-i2\pi k x_j} - \sum_{l=1}^{L} [f](z_l)\ \hat{\beta}_{kh}^1 h$$
$$\textstyle\sum_{j=0}^{N-1}\ A_{z_l}^1(x_j)\ e^{-i2\pi k x_j} = \hat{\beta}_{kh}^1\ (\tilde{f}_k^+ - \displaystyle\sum_{l=1}^{L} [f](z_l)\ (\tilde{A}_{z_l}^1)_k\ ),$$

where $(\tilde{A}_{z_l}^1)_k$ corresponds to the DFT of the sequence $\{A_{z_l}^1(x_j)\}_{j=0}^{N-1}$ and $\hat{\beta}_{kh}^1 = (\frac{\sin(\pi k/N)}{\pi k/N})^2$.

Using the fact that $\hat{f}_k = \hat{u}_k + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}_{z_l}^1)_k$, and approximating $\hat{u}_k$ by $(\hat{p}_u^{1+})_k$, we have that:

$$\hat{f}_k \approx (\hat{p}_u^1)_k + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}_{z_l}^1)_k,$$
$$= \hat{\beta}_{kh}^1\ (\tilde{f}_k^+ - \sum_{l=1}^{L} [f](z_l)\ (\tilde{A}_{z_l}^1)_k\ ) + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}_{z_l}^1)_k,$$
$$= \hat{\beta}_{kh}^1\ \tilde{f}_k^+ + \sum_{l=1}^{L} [f](z_l)\ ((\hat{A}_{z_l}^1)_k - \hat{\beta}_{kh}^1\ (\tilde{A}_{z_l}^1)_k\ ).$$

This provides us with a filter and a reconstruction formula for the point values of f. If we call, as before, g_j the reconstructed values of $f(x_j^+)$, then

$$\tilde{g}_k = \sigma_k^{(1)}\ \hat{f}_k - \textstyle\sum_{l=1}^{L}\ [f](z_l)\ (\sigma_k^{(1)}\ (\hat{A}_{z_l}^1)_k - (\tilde{A}_{z_l}^1)_k\ ),$$
$$k = -N/2, \ldots, 0, \ldots, N/2-1,$$

where

$$\sigma_k^{(1)} = 1/\hat{\beta}_{kh}^1 = \begin{cases} (\frac{\pi k h}{\sin(\pi k h)})^2 & k \neq 0, \\ 1 & k = 0. \end{cases}$$

Observe that the magnitude of the jumps could be obtained by solving the linear system obtained from the N-periodicity of the DFT. Because, from $\tilde{g}_k = \tilde{g}_{k+N}$, we can deduce that

$$\hat{\beta}_k^{(1)} \hat{\beta}_{k+N}^{(1)} \tilde{g}_k = \hat{\beta}_k^{(1)} \hat{\beta}_{k+N}^{(1)} \tilde{g}_{k+N},$$

and using the N-periodicity of $(\tilde{A}_{z_l}^1)_k$ we obtain the linear system

$$\sum_{l=1}^{L} [f](z_l) [\hat{\beta}_{k+N}^{(1)}\ \hat{A}_k^{z_l} - \hat{\beta}_k^{(1)} \hat{A}_{k+N}^{z_l}] = \hat{\beta}_k^{(1)} \hat{f}_{k+N} - \hat{\beta}_{k+N}^{(1)} \hat{f}_k,$$
$$k = k_1, \ldots, k_L.$$

So, assuming known the location of the discontinuity points and their amplitudes, we have the following reconstruction algorithm.

Algorithm 2:
Given the L discontinuity points $z = (z_1, \ldots, z_L)$ of the function f and its Fourier coefficients $\{\hat{f}_k\}_{k=-N/2}^{N/2-1}$:

Step 1: Compute $f_j^{(0)} = ifft_j(\sigma_k^{(1)}\ \hat{f}_k)$.

Step 2: Compute

$$(\hat{A}_{z_l}^1)_k = \frac{e^{-i2\pi k z_l}}{i2\pi k} + \frac{e^{-i2\pi k x_{q_l}}}{i2\pi k}\ (\frac{\sin(2\pi k h)}{2\pi k h},$$
$$k = 0, 1, \ldots, N-1, \text{ and } l = 1, 2, \ldots, L.$$

Step 3: Solve the $(L \times L)$ linear system to compute the jumps amplitude: $[f](z_1), [f](z_2), \ldots, [f](z_L)$,

$$\sum_{l=1}^{L} [f](z_l)\ [\hat{\beta}_{k+N}^{(1)}\ \hat{A}_k^{z_l} - \hat{\beta}_k^{(1)}\ \hat{A}_{k+N}^{z_l}] = \hat{\beta}_k^{(1)}\ \hat{f}_{k+N} - \hat{\beta}_{k+N}^{(1)}\ \hat{f}_k,$$

$$for\ k = k_1, \ldots, k_L.$$

Step 4: Apply the reconstruction formula in the Fourier space,

$$\tilde{g}_k = \sigma_k^{(1)}\ \hat{f}_k - \sum_{l=1}^{L} [f](z_l)\ (\sigma_k^{(1)}\ (\hat{A}_{z_l}^1)_k - (\tilde{A}_{z_l}^1)_k\ ),$$
$$k = -N/2, \ldots, 0, \ldots, N/2-1.$$

Step 5: Apply the IFFT ,

$$g_j = ifft_j(\tilde{g}_k).$$

The vector $\{g_j\}_{j=0}^{N-1}$ contains the approximations for the point values of f at the mesh knots.

C. General Case: $n \geq 2$

The case $n \geq 2$ differs from the previous ones because the coefficients of the interpolating spline no longer correspond to the values of the interpolated function at the mesh points. Now they are given by the discrete convolution of these values with the filter. In what follows we present a derivation of the general filter for any value of n.

As before $p_u^n(x)$ will be the interpolating spline of degree n of the function $u(x)$ at the points (x_j, u_j). So,

$$p_u^n(x) = h \sum_{l \in \mathbf{Z}} c(l)\ \beta_h^n(x-x_l),$$

where $c(l) = [u * (b_1^n)^{-1}](l)$ and $(b_1^n)^{-1}(l)$ is the inverse z Transform of $1/B_1^n(z)$. The k-th Fourier coefficient of this polynomial is given by:

$$
\begin{aligned}
(\hat{p}_u^n)_k &= h \sum_{l\in \mathbf{Z}} c(l)\ \hat{\beta}^n_{kh}\ e^{-i2\pi k x_l}, \\
&= h\ \hat{\beta}^n_{kh} \sum_{l} c(l)\ \ z_k^{-l}; \qquad z_k = e^{i2\pi k/N} \\
&= h\ \hat{\beta}^n_{kh}\ C(z)|_{z=z_k},
\end{aligned}
$$

where $C(z)$ denotes the z Transform of the sequence $c(l)$.

Using the fact that the z Transform of a convolution is the product of the z Transforms we get that:

$$(\hat{p}_u^n)_k = h\ \hat{\beta}^n_{kh}\ [U(z)\ \frac{1}{B_1^n(z)})]|_{z=z_k},$$

Then it holds,

$$(\hat{p}_u^n)_k \approx \frac{\hat{\beta}^n_{kh}}{B_1^n(z_k)}\ \tilde{u_k}.$$

Taking into account the fact that $(\hat{p}_u^n)_k$ is an approximation of $\hat{u}_k$ we obtain,

$$\hat{u}_k \approx \frac{\hat{\beta}^n_{kh}}{B_1^n(z_k)}\ \tilde{u_k},$$

So, we have a general expression for the filters when $n \geq 2$,

$$\sigma_k^{(n)} = \frac{B_1^n(z_k)}{\hat{\beta}^n_{kh}}. \tag{15}$$

The general expression for the z Transform of $\beta^n(x)$ is given by

$$B_1^n(\omega) = b_1^n(0) + \sum_{k=1}^{[n/2]} 2b_1^n(k)\ \cos(2\pi\omega k).$$

where $b_1^n(k)$ are the coefficients of the B-spline of degree n and ω corresponds to any given frequency. So, the general form of the degree n filters is given by,

$$\sigma_k^{(n)} = (\hat{\beta}^{(n)}_{kh})^{-1}\ [b_1^n(0) + \sum_{k=1}^{[n/2]} 2b_1^n(k)\ \cos(2\pi\omega k)].$$

In order to establish a formula for the case $n = 2$, assuming known the locations and amplitudes of the discontinuities of the function and its derivatives, we proceed as before. We know that, in this case, f has the following form:

$$f(x) = u(x) + \textstyle\sum_{l=1}^{L}\ [f](z_l)\ A^1_{z_l}(x) + [f'](z_l)\ A^2_{z_l}(x),$$

where $A^1_{z_l}(x)$ corresponds to the function previously defined for the first degree method and $A^2_{z_l}(x)$ is defined by:

$$
A^2_{z_l}(x) = \begin{cases} \frac{x(x-x_{q_l-1})(z-x_{q_l+1})}{2hz} & \text{for } \ x_{q_l-1} < x \leq z\ , \\ \frac{x(x-x_{q_l+1})(z-x_{q_l-1})}{2hz} & \text{for } \ z < x < x_{q_l+1}\ , \\ 0 & \text{otherwise.} \end{cases}
$$

We observe that this function satisfies $A^2_{z_l}(x_{q_l-1}) = A^2_{z_l}(x_{q_l+1}) = 0$, $[A^2_{z_l}](z) = 0$ and $[A^{2'}_{z_l}](z) = A^{2'}_{z_l}(z^+) - A^{2'}_{z_l}(z^-) = \frac{(z-x_{q_l-1})(2z-x_{q_l+1})}{2hz} - \frac{(z-x_{q_l+1})(2z-x_{q_l-1})}{2hz} = 1.$

The properties of this auxiliary function, together with the previously defined one $A^1_{z_l}(x)$ make $u(x)$ a continuous function because,

$$
\begin{aligned}
[u](z_r) &= [f](z_r) - \sum_{l=1}^{L} [f](z_l)\ [A^1_{z_l}](z_r) + [f'](z_l)\ [A^2_{z_l}](z_r), \\
&= [f](z_r) - \sum_{l=1}^{L} [f](z_l)\ \delta_{lr} = 0.
\end{aligned}
$$

Also observe that $[u'](z_r) = 0$. At the subintervals not containing discontinuity points, that is, at $[0,1] - \bigcup_l [x_{q_l-1}, x_{q_l+1}]$, the function u is as regular as f.

Using the function $u(x)$, the approximation given by the second degree spline in the Fourier space is given by

$$\hat{u}_k \approx (\hat{p}_u^2)_k = \tau_k^{(2)}\ \tilde{u}_k, \quad \tau_k^{(2)} = 1/\sigma_k^{(2)}.$$

Using this approximation for $n = 2$ we have that,

$$
\begin{aligned}
\hat{f}_k &= \hat{u}_k + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}^1_{z_l})_k + [f'](z_l)\ (\hat{A}^2_{z_l})_k \\
&\approx \tau_k^{(2)}\ \tilde{u}_k + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}^1_{z_l})_k + [f'](z_l)\ (\hat{A}^2_{z_l})_k, \\
&= \tau_k^{(2)}\ h \sum_{j=0}^{N-1}\ e^{-i2\pi k x_j} (f_j^+ - \sum_{l=1}^{L} [f](z_l)\ A^1_{z_l}(x_j)\ + \\
&+ [f'](z_l) A^2_{z_l}(x_j)\ ) + \sum_{l=1}^{L} [f](z_l)\ (\hat{A}^1_{z_l})_k + [f'](z_l)\ (\hat{A}^2_{z_l})_k,
\end{aligned}
$$

and the approximation,

$$
\begin{aligned}
\hat{f}_k \approx \tau_k^{(2)}\ \tilde{f}_k^+ &- \textstyle\sum_{l=1}^{L}\ [f](z_l)\ (\tau_k^{(2)}\ (\tilde{A}^1_{z_l})_k - (\hat{A}^1_{z_l})_k) \\
&+ [f'](z_l)\ (\tau_k^{(2)}\ (\tilde{A}^2_{z_l})_k - (\hat{A}^2_{z_l})_k).
\end{aligned}
$$

This gives rise to the second degree formula

$$
\begin{aligned}
\tilde{g}_k^+ = \sigma_k^{(2)}\ \hat{f}_k &- \textstyle\sum_{l=1}^{L}\ [g](z_l)\ (\sigma_k^{(2)}\ (\hat{A}^1_{z_l})_k - (\tilde{A}^1_{z_l})_k\ ) + \\
&+ [g'](z_l)\ (\sigma_k^{(2)}\ (\hat{A}^2_{z_l})_k - (\tilde{A}^2_{z_l})_k\ ),
\end{aligned}
$$

where, as before, the $\{g_j^+\}_{j=0}^{N-1}$ are the approximations provided by the method to the values of $f(x_j^+)$ for $j = 0, 1, \ldots, N-1$, after applying the IDFT to the last expression.

In the same way as the first degree case, that is, using the N-periodicity of the DFT, we can build a linear system that gives the approximations for the amplitudes of the jumps at the singular points. Therefore, from the fact that $\tilde{g}^+_{k+N} = \tilde{g}^+_k$, we have that for $k = k_1, \ldots, k_{2L}$,

$$
\begin{aligned}
&\sigma_{k+N}^{(2)}\ \hat{f}_{k+N} - \sigma_k^{(2)}\ \hat{f}_k = \\
&\sum_{l=1}^{L} (\ \sigma_{k+N}^{(2)}\ \hat{A}^1_{k+N,z_l} - \sigma_k^{(2)}\ \hat{A}^1_{k,z_l}\ )\ [g](z_l) + \\
&+ (\ \sigma_{k+N}^{(2)}\ \hat{A}^2_{k+N,z_l} - \sigma_k^{(2)}\ \hat{A}^2_{k,z_l}\ )\ [g'](z_l).
\end{aligned} \tag{16}
$$

It follows the algorithm that corresponds to the second degree filtering

Algorithm 3:
Given the Fourier coefficients $\{\hat{f}_k\}_{k=-N/2-L}^{N/2-1+L}$ and the L discontinuity points of the function and its derivatives:

Step 1: Compute $\quad f_j^{(2)} = ifft_j\ (\sigma_k^{(2)} \hat{f}_k),$

$$a_{jl}^{(1)}(z_l) = ifft_j\ (\sigma_k^{(2)} \hat{A}^1_{k,z_l} - \tilde{A}^1_{k,z_l}),$$

$$a_{jl}^{(2)}(z_l) = ifft_j\ (\sigma_k^{(2)} \hat{A}^2_{k,z_l} - \tilde{A}^2_{k,z_l}).$$

Step 2: Approximate the jumps $[g](z_l)$ and $[g'](z_l)$ through solving the linear system (16).

Step 3: Compute the approximations of the point values of f using the reconstruction formula,

$$g_j = f_j^{(2)} - \sum_{l=1}^{L} [g](z_l)\ a_{jl}^{(1)}(z_l) + [g'](z_l)\ a_{jl}^{(2)}(z_l).$$

Remark 1: In all the examples considered, the discontinuity points of the function and its derivative are the same. Approximations of the discontinuity points of f' could be obtained using the same algorithm that determines the approximations for the discontinuities of f, but modifying the Fourier coefficients in the following way: $\widehat{df}_k = ik\pi\hat{f}_k + \sum_{l=1}^{L_1}[f](z_l)e^{-ik\pi z_l}$ for $k \neq 0$, and $\widehat{df}_0 = -\sum_{l=1}^{L_1}[f](z_l)$.

Remark 2: The choices of $A^0_{z_l}$, $A^1_{z_l}$ and $A^2_{z_l}$ are not unique, but are the simplest ones for the calculations.

IV. Numerical Examples

In this section, we show the results obtained when applying the new filters to the following functions.

$$f_1(x) = \begin{cases} x^2 & x \in [0,1], \\ 0 & x \notin [0,1]. \end{cases}$$

$$f_2(x) = \begin{cases} x^2 & x \in [0, 0.5 + 1/256], \\ \cos(x) & x \notin [0, 0.5 + 1/256]. \end{cases}$$

$$f_3(x) = \begin{cases} \exp 5x & 0 \leq x \leq 0.3, \\ 2 & 0.3 < x \leq 0.5, \\ -4\cos(\pi x) & 0.5 < x \leq 1, \end{cases}$$

Notice that f_1 is analytic in the interval $(0,1)$ but $f_1(1) \neq f_1(0)$. Figures 1-3 show, for each function, the graphics of the logarithm of the pointwise error of the reconstruction for each of the algorithms described in the previous sections. Also we show comparisons with the errors obtained when using the polynomial filters of [24]. The reconstructions are not shown because of the high resolution obtained, that does not allow a visual comparison. In all cases $N = 64$. For the piecewise quadratic function, as expected, accuracy was the order of the machine precission when using the second degree spline filter (Figure 1(c)), better than accuracy obtained by zeroth and first degree spline filters (Figure 1(a-b)). The error for the first degree polynomial and spline filters were indistinguishable. Comparison of these methods with the direct IFFT approach (with different filters) appeared in [24] and [21].

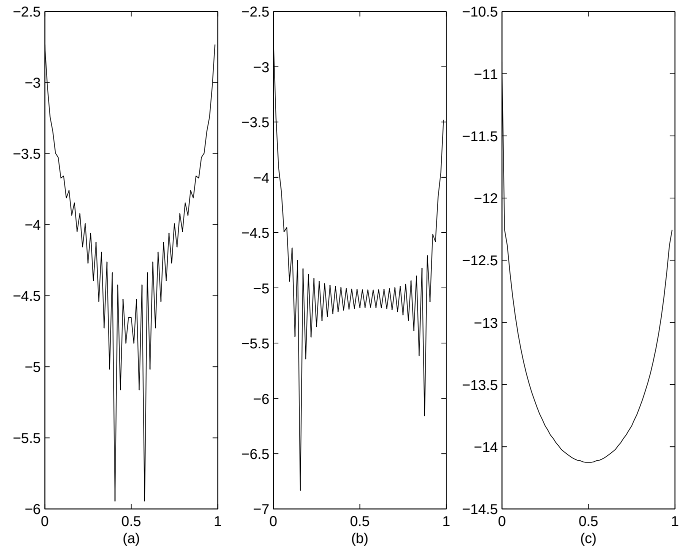

Fig. 1. Log_{10} of the reconstruction error for f_1: (a) zeroth degree spline filter (b) first degree polynomial/spline filter (c) second degree spline filter

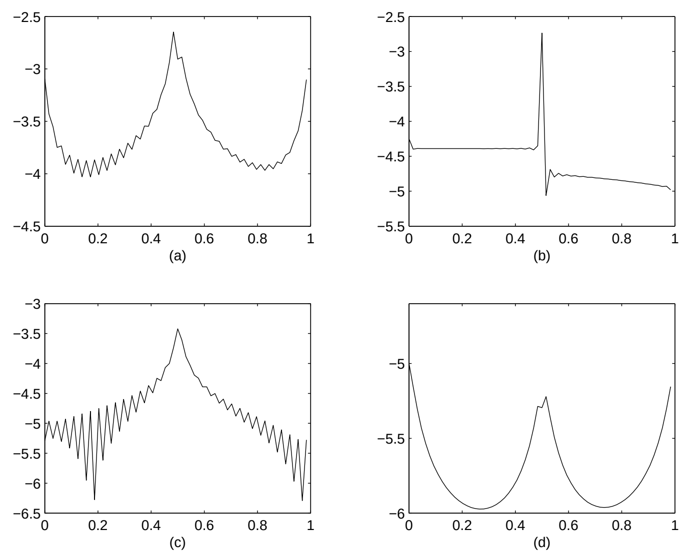

Fig. 2. Log_{10} of the reconstruction error for f_2: (a) zeroth degree spline filter (b) first degree polynomial filter (c) first degree spline filter (d) second degree spline filter

For f_2 the reconstruction error for the second degree spline filter (Figure 2(d)) looks significantly lower than the others. The same can be said for f_3; this is clearly seen in Figure 3(a-d).

For a global comparison we define the Mean Square Error (MSE) as

$$\sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (f_j - g_j)^2}.$$

The g_j's correspond to the point values of the function.

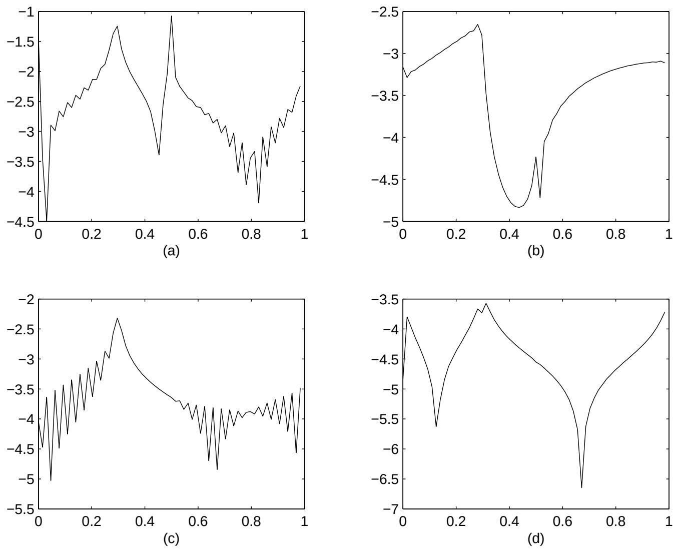

Fig. 3. Log_{10} of the reconstruction error for f_3: (a) zeroth degree spline filter (b) first degree polynomial filter (c) first degree spline filter (d) spline filter degree two

N	$\sigma_k^{(0)}$	$\sigma_k^{p(1)}$	$\sigma_k^{sp(1)}$	$\sigma_k^{sp(2)}$
64	4.0675e-004	—	4.0619e-005	1.5600e-012
128	1.4535e-004	—	1.0149e-005	5.5160e-013
256	5.1663e-005	—	2.5539e-006	1.9503e-013

TABLE I

MSE FOR THE RECONSTRUCTION OF THE FUNCTION f_1

Tables I, II and III show the MSE's for the three functions and three different values of N. $\sigma_k^{(0)}$ denotes the zeroth degree method, $\sigma_k^{p(1)}$ the first degree polynomial method, $\sigma_k^{sp(1)}$ the first degree spline method, and $\sigma_k^{sp(2)}$ the second degree spline method. The results obtained are consistent with the estimates shown in the Appendix.

We define the signal-to-noise ratio (SNR) as

$$SNR = \sqrt{\frac{\sum_j (f_j - \bar{f})^2}{\sum_j (\xi_j - \bar{\xi})^2}}, \tag{17}$$

where $\bar{f} = \frac{1}{N}\sum_j f_j$ is the signal average and $\hat{\eta} = \frac{1}{N}\sum_j \eta_j$ the corresponding noise.

N	$\sigma_k^{(0)}$	$\sigma_k^{p(1)}$	$\sigma_k^{sp(1)}$	$\sigma_k^{sp(2)}$
64	4.8671e-004	2.3148e-004	3.4991e-004	2.9100e-006
128	3.2773e-004	1.6907e-004	1.6611e-004	3.4484e-007
256	5.3404e-005	2.0768e-006	2.0420e-006	9.2083e-008

TABLE II

MSE FOR THE RECONSTRUCTION OF THE FUNCTION f_2

N	$\sigma_k^{(0)}$	$\sigma_k^{p(1)}$	$\sigma_k^{sp(1)}$	$\sigma_k^{sp(2)}$
64	0.0157	8.0745e-004	6.1055e-004	8.2598e-005
128	0.0091	2.3852e-004	1.3852e-004	1.0258e-005
256	0.0015	5.0275e-005	3.5651e-005	2.7998e-006

TABLE III

MSE FOR THE RECONSTRUCTION OF THE FUNCTION f_3

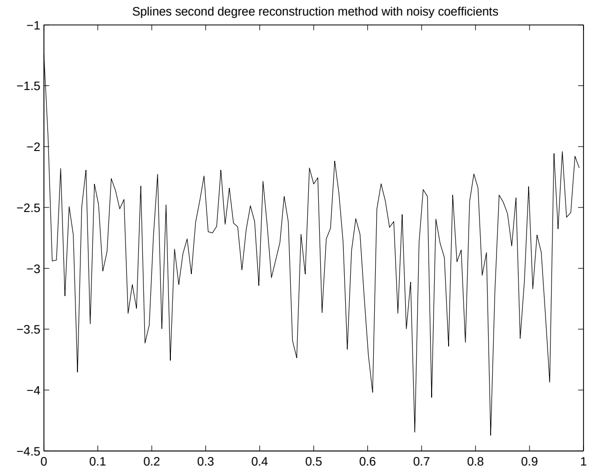

Fig. 4. Pointwise reconstruction error $log_{10}|g(x) - f_1(x)|$ with N=128, g(x) is the second degree splines reconstruction formula. SNR=14

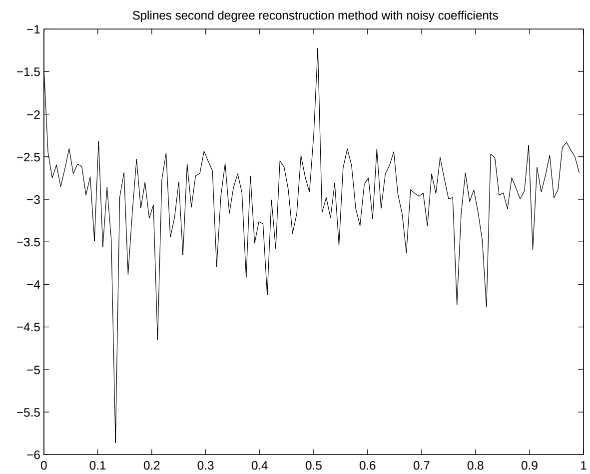

Fig. 5. Pointwise reconstruction error $log_{10}|g(x) - f_2(x)|$ with N=128, g(x) is the second degree splines reconstruction formula. SNR=22

Figures 4-6 show the errors for the second degree spline filter applied to the three functions with the respective SNR's (around 20%). With those levels of errors, the results deteriorate significantly as expected for all methods trying to perform this task, but still better for higher degree filters.

V. CONCLUSION

We have presented general spline filters that are able to reconstruct piecewise smooth functions from a finite number of their Fourier coefficients with any given accuracy, assuming known the discontinuity points and their amplitudes. These results solve a problem posed in a previous paper [24]. The natural continuation is the development of iterative methods based on the new filters to detect the discontinuities as in [22]. This is the subject of a forthcoming article. We are now working on extensions of our approach to a broader family of functions with discontinuous derivatives as well as on applications where data is given by two dimensional Fourier coefficients. Another research direction (suggested by a reviewer) is the possibility of using other interpolating

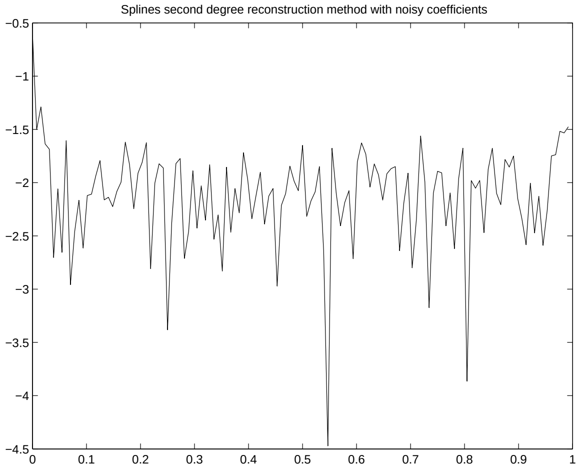

Fig. 6. Pointwise reconstruction error $log_{10}|g(x) - f_3(x)|$ with N=128, g(x) is the second degree splines reconstruction formula. SNR=21

kernels like, for example, those generated by wavelets, as in [20].

Appendix
Error Estimates

In what follows we assume that

- f has at most a finite number $L \geq 0$ of jumps, $\{z_1, \ldots, z_L\}$.
- If $Z = \{z_1, \ldots, z_L\}$ denotes the set of jumps, then for $x \in [0,1]$ and $x \notin Z$, $f'(x), f''(x), f'''(x)$ exist and are bounded, that is, : $\sup_{x\in[0,1]} |f(x)| < C$, $\sup_{x\notin Z} |f'(x)| < C_1$, $\sup_{x\notin Z} |f''(x)| < C_2$, and $\sup_{x\notin Z} |f'''(x)| < C_3$ for some constants C, C_1, C_2 and C_3.
- If $L > 1$, jumps should be enough separated, i.e., $N \min_{r\neq l} d_{q_l,q_r} \gg 1$, where q_l denotes the subindex of the mesh point z_l and

$$d_{jl} = \min_{k\in\{-1,0,1\}} ||x_j - x_l - k| - \frac{1}{2N}|.$$

We present error estimates for the reconstruction methods using degree 2 splines that can be generalized to higher degrees. The estimates will be obtained in the Fourier space. So we will estimate

$$\hat{e}_k^{(2)} = \hat{f}_k - (\hat{p}_f^2)_k = \hat{u}_k - (\hat{p}_u^2)_k.$$

where $p_u^2(x)$ corresponds to the spline interpolator of degree 2 of the continuous function $u(x)$. Therefore,

$$\hat{e}_k^{(2)} = \int_0^1 (u(x) - p_u^{(2)}(x)) e^{-i2\pi kx}\ dx.$$

Because of the fact that the approximation given by $p_u^{(2)}(x)$ is an interval approximation, we consider the difference $d_2(x) = u(x) - p_u^{(2)}(x)$, for each subinterval $[x_j, x_{j+1}]$, for $j = 0, 1, \ldots, N-1$. Let z be a discontinuity point and x_{q_z} the closest point to z in the uniform mesh. For those subintervals without discontinuity points it is possible to assume that the function u is regular enough, so, the following estimate given by the approximation error for the interpolating splines of degree 2 are valid:

$$\max_{x\in(x_j,x_{j+1})/j\neq q_z, q_z-1} |d_2(x)| \leq K\ C_3 h^3,$$

where K is a constant (an explanation for the origin of this constant can be found in [13] and [5]). So, we have that,

$$\begin{aligned} |\hat{e}_k^{(2)}| = &| \sum_{j=0, j\neq q_z, q_z-1}^{N-1} \int_{x_j}^{x_{j+1}} d_2(x)\ e^{-i2\pi kx}\ dx + \\ &+ \int_{x_{q_z-1}}^{x_{q_z+1}} d_2(x)\ e^{-i2\pi kx}\ dx|, \\ &\leq (N-2)KC_3h^4 + |\int_{x_{q_z-1}}^{x_{q_z+1}} d_2(x)\ e^{-i2\pi kx}\ dx|. \end{aligned}$$

It remains now to find bounds for the second part of the expression. In order to do this we will assume that the point $z \in (x_{q_z}, x_{q_z+1})$; so, it is still possible to assume the regularity of the function for the interval (x_{q_z-1}, x_{q_z}), thus obtaining

$$\begin{aligned} &|\int_{x_{q_z-1}}^{x_{q_z+1}} d_2(x)\ e^{-i2\pi kx}\ dx| \leq KC_3h^4 + \\ &|\int_{x_{q_z}}^{x_{q_z+1}} d_2(x)\ e^{-i2\pi kx}\ dx|. \end{aligned}$$

For $x \in (x_{q_z}, x_{q_z+1})$, the approximation given by the degree two splines can be written as $p_u^2(x) = L_{q_z}(x) + (x - x_{q_z})(x - x_{q_z+1})\ M_q$, where $L_{q_z}(x)$ corresponds to the linear approximation of the function $u(x)$ at the interval (x_{q_z}, x_{q_z+1}), and $M_q = [x_{q_z+1}, x_{q_z}, x]u$ is the second order divided difference of the function $u(x)$, that is

$$M_q = \tfrac{1}{(x-x_{q_z})}[\tfrac{p_u^2(x)-u(x_{q_z})}{(x-x_{q_z})} - \tfrac{u(x_{q_z})-u(x_{q_z+1})}{(x_{q_z}-x_{q_z+1})}].$$

$p_u^2(x) \in C^1[0,1]$, so, from the continuity condition for the first derivative at $x = x_{q_z}$, we get

$$L'_{q_z-1}(x) + 2(x - \frac{(x_{q_z-1} + x_{q_z})}{2})\ M_{q-1} =$$

$$L'_{q_z}(x) + 2(x - \frac{(x_{q_z+1} + x_{q_z})}{2})\ M_q.$$

And we obtain that,

$$hM_{q-1} + (x - x_{q_z})M_q =$$

$$\frac{u(x_{q_z+1}) - 2u(x_{q_z}) + u(x_{q_z-1})}{h^2}(x-x_{q_z}) + (x_{q_z+1}-x)M_{q-1},$$

then,

$$M_q = \frac{u(x_{q_z+1}) - 2u(x_{q_z}) + u(x_{q_z-1})}{h^2} - M_{q-1}.$$

$$\Delta^2 h = \frac{1}{h^2}\{(u(x_{q_z+1}) - u(z)) + (u(z) - u(x_{q_z}))+$$
$$+(u(x_{q_z-1}) - u(x_{q_z}))\}$$
$$= \frac{1}{h^2}\{u'(z^+)(x_{q_z+1} - z) + \frac{1}{2}u''(z^+)(x_{q_z+1} - z)^2+$$
$$+\frac{1}{6}u'''(\xi_1)(x_{q_z+1} - z)^3 \quad - (u'(z^-)(x_{q_z} - z)+$$
$$+\frac{1}{2}u''(z^-)(x_{q_z} - z)^2 + \frac{1}{6}u'''(\xi_2)(x_{q_z} - z)^3)$$
$$-(h\ u'(x_{q_z}) - \frac{1}{2}u''(x_{q_z})\ h^2 + \frac{1}{6}u'''(\xi_3)\ h^3)\},$$
$$= \frac{1}{h^2}\{(x_{q_z+1} - z)\ [u'](z) + h\ (u'(z^-) - u'(x_{q_z}))+$$
$$+\frac{1}{2}(u''(z^+)(x_{q_z+1} - z)^2 - u''(z^-)(x_{q_z} - z)^2+$$
$$+u''(x_{q_z})\ h^2) + \frac{1}{6}(u'''(\xi_1)(x_{q_z+1} - z)^3-$$
$$-u'''(\xi_2)(x_{q_z} - z)^3 - u'''(\xi_3)\ h^3)\},$$
$$= \frac{1}{h^2}\{(x_{q_z+1} - z)\ [u'](z) + h\ (u''(z^-)(x_{q_z} - z)+$$
$$+\frac{1}{2}u'''(\xi_4)(x_{q_z} - z)^2) + \frac{1}{2}(u''(z^+)(x_{q_z+1} - z)^2-$$
$$-u''(z^-)(x_{q_z} - z)^2 + u''(x_{q_z})\ h^2)+$$
$$+\frac{1}{6}(u'''(\xi_1)(x_{q_z+1} - z)^3-$$
$$-u'''(\xi_2)(x_{q_z} - z)^3 - u'''(\xi_3)\ h^3)\},$$

where we have used the following expressions,

$$u(x_{q_z+1}) = u(z) + u'(z^+)(x_{q_z+1} - z)+$$
$$\frac{1}{2}\ u''(z^+)(x_{q_z+1} - z)^2 + \frac{1}{6}\ u'''(\xi_1)(x_{q_z+1} - z)^3;$$
$$u(x_{q_z}) = u(z) + u'(z^-)(x_{q_z} - z) + \frac{1}{2}\ u''(z^-)(x_{q_z} - z)^2+$$
$$\frac{1}{6}\ u'''(\xi_2)(x_{q_z} - z)^3);$$
$$u(x_{q_z-1}) = u(x_{q_z}) - h\ u'(x_{q_z}) + \frac{1}{2}\ u''(x_{q_z})\ h^2+$$
$$+\frac{1}{6}\ u'''(\xi_3)\ h^3;$$

where $\xi_1 \in (z, x_{q_z+1}),\ \xi_2 \in (x_{q_z}, z)$ and $\xi_3 \in (x_{q_z}, x)$.

Assuming that $[u'](z) = 0$, we have that, $|\Delta^2 h| \leq 2C_2 + hC_3$, then $|M_q| \leq 3C_2 + hC_3$.

Considering now the values of x at the subinterval $[x_{q_z}, z)$, we will estimate the difference $d_2(x) = u(x) - p_u^2(x)$:

$$d_2(x) = u(x) - L_{q_z}(x) - (x - x_{q_z})(x - x_{q_z+1})\ M_q$$

At the subinterval $[x_{q_z}, z), u(x)$ is regular enough and the following inequalities are valid:

$$u(x) = u(z) + u'(z^-)(x - z) + \frac{1}{2}\ u''(\theta_1)(x - z)^2;$$

for $\theta_1 \in (x, z)$,

$$u(x_{q_z+1}) = u(z) + u'(z^+)(x_{q_z+1} - z)+$$
$$+\frac{1}{2}\ u''(\theta_2)(x_{q_z+1} - z)^2;$$ for $\theta_2 \in (z, x_{q_z+1})$,

$$u(x) = u(x_{q_z}) + u'(x_{q_z})(x - x_{q_z}) + \frac{1}{2}\ u''(\theta_3)(x - x_{q_z})^2;$$

for $\theta_3 \in (x_{q_z}, x)$.

So,

$$u(x) - L_{q_z}(x) = \frac{1}{h}[(u(x) - u(x_{q_z+1}))(x - x_{q_z})+$$
$$+(u(x) - u(x_{q_z}))(x - x_{q_z+1})$$
$$= \frac{1}{h}[(x - x_{q_z})(u'(z^-)(x - z) - u'(z^+)(x_{q_z+1} - z)+$$
$$+\frac{1}{2}\ u''(\theta_1)(x - z)^2 - \quad -\frac{1}{2}\ u''(\theta_2)(x_{q_z+1} - z)^2)+$$
$$+(x - x_{q_z+1})(u'(x_{q_z})(x - x_{q_z}) + \frac{1}{2}\ u''(\theta_3)(x - x_{q_z})^2)],$$
$$= \frac{1}{h}[(x - x_{q_z})(x_{q_z+1} - z)\ (u'(z^-) - u'(z^+))+$$
$$+(u'(z^-) - u'(x_{q_z}))(x - x_{q_z})(x - x_{q_z+1})+$$
$$+\quad \frac{(x - x_{q_z})}{2}[u''(\theta_1)(x - z)^2 - u''(\theta_2)(x_{q_z+1} - z)^2-$$
$$-u''(\theta_3)(x - x_{q_z})(x - x_{q_z+1})]],$$
$$= \frac{1}{h}[(x - x_{q_z})(x_{q_z+1} - z)\ [u'](z)+$$
$$+u''(\theta_4)(x_{q_z} - z)(x - x_{q_z})(x - x_{q_z+1})+$$
$$+\frac{(x - x_{q_z})}{2}[u''(\theta_1)(x - z)^2 - u''(\theta_2)(x_{q_z+1} - z)^2-$$
$$-u''(\theta_3)(x - x_{q_z})(x - x_{q_z+1})]],$$

because $u'(x_{q_z}) = u'(z^-) + u''(\theta_4)(x - x_{q_z});\quad \theta_4 \in (x_{q_z}, z)$. Substituting the expression obtained for $u(x) - L_{q_z}(x)$ in $d_2(x)$ we get that,

$$d_2(x) = \frac{1}{h}\ [(x - x_{q_z})\ (x_{q_z+1} - z)\ [u'](z)+$$
$$+u''(\theta_4)\ (x_{q_z} - z)(x - x_{q_z})(x - x_{q_z+1})+$$
$$+\frac{(x - x_{q_z})}{2}\ [u''(\theta_1)(x - z)^2 - u''(\theta_2)(x_{q_z+1} - z)^2-$$
$$-u''(\theta_3)\ (x - x_{q_z})(x - x_{q_z+1})]\ ]-$$
$$-(x - x_{q_z})(x - x_{q_z+1})\ M_q.$$

We have that,

$$|d_2(x)| \leq h\ [u'](z) + h^2(u''(\theta_4) + \frac{1}{2}(u''(\theta_1) + u''(\theta_2)+$$
$$+u''(\theta_3)\ ) + h^2\ M_q,$$
$$\leq h\ [u'](z) + \frac{5}{2}\ C_2 h^2 + h^2(3C_2 + hC_3) = \frac{11}{2}\ C_2 h^2 + h^3 C_3.$$

So, the contribution of $d_2(x)$ for the spectral error at this subinterval is of order $O(h^3)$,

$$|\int_{x_{q_z}}^{z}\ d_2(x)\ e^{-i2\pi kx}\ dx\ | \leq \frac{11}{4}\ C_2 h^3 + O(h^4).$$

Similarly we obtain that, $|\int_{z}^{x_{q_z+1}}\ d_2(x)\ e^{-i2\pi kx}\ dx\ | \leq R\ C_2 h^3$.

For the case in which the discontinuity point $z \in [x_{q_z-1}, x_{q_z}]$, using L_{q_z-1} instead of L_{q_z}, and proceeding in the same manner it is possible to obtain a similar estimate. So, the error estimate, in the presence of L discontinuity points, for the approximation in the frequency space for the degree 2 method is of order h^3:

$$|\hat{e}_k^{(2)}| \leq (N-1)KC_3 h^4 + \frac{11}{4}\ C_2 h^3 + RC_2 h^3 + O(h^4)$$
$$\approx (KC_3 + C_2(\frac{11}{4} + R))L\ h^3.$$

In the same way, for the reconstruction formulas using filters of higher degrees, say r, it is possible to obtain error estimates of order h^{r+1}.

ACKNOWLEDGMENT

A. G. Martínez was supported by FAPESP Grant No 2005/60892-6 and A. De Pierro was partially supported by FAPESP Grant No 2002/07153-2 and CNPq Grants 300969/2003-1 and 476825/2004-0.

REFERENCES

[1] R. Archibald and A. Gelb, A method to reduce the Gibbs ringing artifact in MRI while keeping tissue boundary integrity, IEEE Trans. Med. Imag., 21, 4, 305-319, 2002.

[2] C. de Boor, *A Practical Guide to Splines* , New York, Springer Verlag, 1978.

[3] E.O. Brigham, *The Fast Fourier Transform and its Applications* , Prentice Hall, 1988.

[4] C. Canuto, M.Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.

[5] F. Dubeau and J. Saboie, Best error bounds for odd and even degree deficient splines, SIAM J. Numer. Analysis, 34, 3, 1167-1184, 1997.

[6] A. Gelb, A hybrid approach to spectral reconstruction of piecewise smooth functions, J. Sci. Comput., 15, 3, 293-322, 2000.

[7] A. Gelb and E. Tadmor, Detection of edges in spectral data, Appl. Comp. Harmonic Analysis 7, 101-135, 1999.

[8] A. Gelb and E. Tadmor, Detection of edges in spectral data II: nonlinear enhancement, SIAM J. Numer. Analysis, 38, 4, 1389-1408, 2000.

[9] D. Gottlieb and S. Orszag,*Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia, PA, 1977.

[10] D. Gottlieb, C.W. Shu, A. Solomonoff and H. Vandeven, " On the Gibbs phenomenon I: recovering exponential accuracy from the Fourier partial sum of the a nonperiodic analytic function", J. Comput. Appl. Math., Vol. 43,pp. 81-98, 1992.

[11] D. Gottlieb and C.W. Shu, On the Gibbs phenomenon and its resolution, SIAM Rev, **39**, 644-668, 1997

[12] D. Gottlieb, Gustafsson and P. Forssen, On the direct Fourier method for computed tomography, IEEE Trans. Med. Imag., 19, 3, 223-233, 2000.

[13] G. Hammerlin, K.-H. Hoffmann, Numerical Mathematics, Springer-Verlag, 1991.

[14] G. Hamming, *Numerical Methods for Scientists and Engineers*, 2nd Edit., McGraw-Hill, New York, 1973.

[15] A. K. Jain, *Fundamentals of Digital Image Processing* , Englewood Cliffs, NJ: Prentice-Hall, 1989.

[16] A. J. Jerri, *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations* , Kluwer Academic Publishers, 1998.

[17] M. Unser, A. Aldroubi and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation", IEEE Trans. Pattern Anal. Machine Intell., vol. 13 N0 3, 277-285, 1991

[18] M. Unser, A. Aldroubi and M. Eden, "B-spline signal approximation filter design and asymptotic equivalence with Shannon's sampling theorem", IEEE Trans. Information Theory, 38, 1, 95-103, 1992.

[19] D. H. Vandeven, " Family of spectral filters for discontinuous problem", J. Sci. Comp, **6**, 159-192, 1991.

[20] G.G. Walter, "A sampling theorem for wavelets subspaces", IEEE Trans. Info. Theory, 38, 881-884, 1992.

[21] M. Wei, A.R. De Pierro and Yin J. "Error estimates for two filters based on polynomial interpolation for recovering a function from its Fourier coefficients", Numerical Algorithms, 35, 205-232, 2004.

[22] M. Wei, A.R. De Pierro and J. Yin, "Iterative methods based on polynomial interpolation filters to detect discontinuities and recover point values from Fourier data", IEEE Transactions on Signal Processing, 53, 1, 136-146. January 2005.

[23] M. Wei, A.G. Martínez and A.R. De Pierro, "Detection of edges from spectral data: new results", Applied and Computational Harmonic Analysis, 2007 (available on-line 11-16-06).

[24] J. Yin, A.R. De Pierro and M. Wei, "Reconstruction of a compactly supported function from the discrete sampling of its Fourier transform", IEEE Transactions on Signal Processing, vol 47, No 12, 3356-3364, 1999.

[25] G. Zelniker and F.J. Taylor, *Advanced Digital Signal Processing, Theory and Applications* , Marcel Dekker, New York, 1994.