UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Data Compression Algorithms for Locally Oscillatory Data

Rosa Donat
Ami Harten

July 1993

CAM Report 93-26

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

DATA COMPRESSION ALGORITHMS FOR LOCALLY OSCILLATORY DATA

ROSA DONAT[1,3] AND AMI HARTEN[2,4]

Abstract. The efficiency of a data compression algorithm is very dependent on the nature of the data to which it is applied. Smooth, periodic signals with high frequency components are best compressed using global algorithms derived from Fourier techniques, while piecewise smooth, non periodic functions are efficiently compressed with local algorithms based on multi-scale representations of the data. In this report we are concerned with the compression of locally oscillatory signals, that is, piecewise smooth signals that exhibit high frequency oscillatory behavior in some portions of their domain of definition. For this type of data, neither local not global algorithms give acceptable compression rates. We design compression algorithms that combine both: local techniques for the slowly varying part of the signal and global techniques (applied locally) to represent the oscillatory components.

Key Words.: Multi-resolution, locally oscillatory signal, Fourier transform.

AMS(MOS) subject classifications. Primary 65T99, secondary 42C10

1. Introduction. In many occasions, one is faced with the problem of storing or transmitting large amounts of data in an optimal way. To do so, one designs data compression algorithms that select or compute a much smaller set of data, from which the original one can be recovered up to a given precision, and store or transmit this smaller set.

The nature of the data should determine the kind of compression algorithm to be used. Take for instance a smooth, purely oscillatory signal, that is, a smooth periodic signal with high frequency components. Data compression algorithms based on local properties will not give good compression rates because they fail to see the global, repetitive structure of the signal.

Let us consider a periodic sine wave, $f(x) = \sin(2\pi x)$, $x \in [0,1]$. It can be represented by one coefficient in the trigonometric system or approximated by a polynomial that interpolates a set $\{f(x_i)\}$ of at least four equally spaced data. In this case, it does not make much difference which representation we use. However, suppose our signal is $f(x) = \sin(100\pi x)$. In the Fourier domain, we still need just one coefficient to represent the function, but a representation by means of a local interpolation would now require at least 200 equally spaced data.

It is thus obvious that purely oscillatory signals are best compressed using Fourier techniques. However, the performance of compression algorithms based on these techniques depends heavily on the periodicity of the given function. Lack of periodicity results in an infinite number of base functions in the Fourier representation. Ignoring some of them (usually the high frequency base elements) leads to representations of the signal with an unpleasant Gibbs' phenomenon around discontinuities.

In this report, we are mainly concerned with *locally oscillatory signals* , that is, piecewise smooth signals which exhibit high frequency oscillatory behavior in some portions of their domain of definition.

[1] Research supported by a University of Valencia Grant, DGICYT PS90-0265 and ONR-N00014-92-J-1890

[3] Departament de Matemàtica Aplicada Universitat de València (Spain)

[2] Research supported by ONR-N00014-92-J-1890 and NSF-DMS91-03104

[4] School of Mathematical Sciences Tel-Aviv University and Department of Mathematics, UCLA

From our previous comments, it is clear that neither local techniques (based on multiscale analysis) nor global ones (based on Fourier Collocation) will give acceptable compression rates for this type of data.

An approach to the design of compression algorithms for locally oscillatory signals is to combine the best of both techniques, namely local techniques for the slowly varying parts and the discontinuities of the signals and global techniques (applied locally) to represent its oscillatory components.

This can be done by using reconstructions based on the multiresolution analysis of the signal. On a coarse level, these will be oscillation-free, thus, subtracting the original signal from an appropriate coarse level reconstruction will produce an *essentially* flat signal with the highly oscillatory behavior of the original one. The lack of periodicity problem is (to a good extent) solved and we can apply now global techniques to compress the data in each oscillatory region.

A similar idea has been used in [13] to remove noise from piecewise smooth data.

Coifman and Meyer approach the problem in a different manner. In [3] they describe local trigonometric bases that can be used to analyze a signal in a local, yet Fourier-like, manner. We refer to [3, 1] for details on this approach.

Data compression algorithms can be used to design numerical algorithms for the rapid application of a matrix (possibly dense) to a vector. The key observation is that if it is possible to approximate the given matrix by a sparse one by a computationally viable procedure, one has a fast solution algorithm (see [2, 4, 5, 11] and references therein). Obviously, the more efficient the data compression algorithm is, the faster will be the associated 'rapid' numerical procedure.

We expect that our data compression algorithms will lead to the design of fast numerical algorithms for the solution of integral equations with (locally) oscillatory kernels and in general integral equations with solutions that exhibit a locally oscillatory behavior

The rest of the paper is organized as follows: In section 2 we review the concept of multi-resolution analysis and the role of the reconstruction techniques introduced in [9]. In section 3 we concentrate on the interpolatory multi-resolution analysis and describe the basis associated to interpolatory reconstructions that are data-independent. We explore the possibility of using trigonometric polynomials in this multi-resolution set-up to obtain better compression rates for locally oscillatory signals. Section 4 is devoted to the multi-resolution analysis of cell averages. Section 5 displays various numerical experiments showing the performance of these two types of multi-resolution analysis for the type of data under consideration. Although the performance of both algorithms is comparable, multi-resolution algorithms based on cell averages lead to coarse level reconstructions of the signal that keep only the low, long range, frequency components of the original signal. This will be used in section 6 to describe a compression algorithm that combines a multi-resolution analysis with a Fourier decomposition of the local oscillations of the signal. We obtain the local oscillations by subtracting the original signal from a coarse level reconstruction in the underlying multi-resolution analysis. This 'combined' algorithm gives very high compression rates for locally oscillatory data. Numerical experiments confirm that cell-average based multi-resolution leads to compression algorithms that are more robust than the ones derived from the interpolatory multi-resolution. Section 7 deals with signals that are locally oscillatory and discontinuous. We observe that the combined algorithm, using multi-resolution based on cell-averages and ENO-reconstruction with Subcell Resolution ([7, 6]) can also achieve high compression rates for this type of signals.

Several numerical experiments confirm these observations.

2. Discretization and Reconstruction. Following [9], we consider the discrete case in a finite domain. Therefore we associate the various levels of resolution to grids rather than to function spaces, as was done in Mallat's [12] original development.

We start with a finite number of values $\bar{f}_j^0$, $N_0 = 2^{n_0}$, which represent a sampling of weighted averages of a periodic (for simplicity) function $f(x)$, corresponding to a uniform partition of $[0,1]$, i.e.

$$x_j^0 = j \cdot h_0, \qquad 0 \le j \le N_0, \qquad h_0 = \frac{1}{N_0}$$

$$\bar{f}_j^0 = < f, \frac{1}{h_0}\varphi(\frac{x - x_j^0}{h_0}) > .$$

Here $< \cdot, \cdot >$ denotes the Euclidean inner product and $\varphi(x)$ is the weight function. Introducing the nested grids

$$\{\{x_j^k\}_{j=1}^{N_k}\}_{k=1}^{L}, \qquad x_j^k = j \cdot h_k, \qquad h_k = 2^k h_0, \qquad N_k = \frac{1}{h_k} \quad L < n_0$$

and the corresponding scaled quantities

$$\bar{f}_j^k = < f, \varphi_j^k >, \tag{1}$$

where

$$\varphi_j^k(x) = \frac{1}{h_k}\varphi(\frac{x - x_j^k}{h_k}) \tag{2}$$

$$\int \varphi(x)dx = 1 \tag{3}$$

$$\varphi(x) = 2\sum_l \alpha_l \varphi(2x - l), \tag{4}$$

we say (following [9]) that

$$\{\{\bar{f}_j^k\}_{j=1}^{N_k}\}_{k=1}^{L}$$

constitutes a discrete multiresolution analysis if knowledge of the discrete values at level k determines the corresponding values at level $k+1$. It is shown in [9] that (4) is equivalent to

$$\bar{f}_j^{k+1} = \sum_l \alpha_l \bar{f}_{2j+l}^k \tag{5}$$

Thus, in a discrete multi-resolution analysis, the values at the k-th level of resolution determine the values of all larger scales.

At the heart of the data compression algorithms based on a multi-resolution analysis lies a reconstruction technique, $R(x; \bar{f}^k)$ which approximates $f(x)$ from the discrete set of values $\{\bar{f}_j^k\}_{j=1}^{N_k}$. The reconstruction procedure is required to satisfy the following two properties:

$$R(x; \bar{f}^k) = f(x) + O((h_k)^r) \tag{6}$$

for a given r, wherever $f(x)$ is smooth and

$$< R(\cdot;\bar{f}^k), \varphi_j^k >= \bar{f}_j^k \qquad \forall k. \tag{7}$$

We say that the reconstruction is conservative if it satisfies (7), and r is its order of the accuracy.

On the functional side, the multiresolution analysis plus the reconstruction procedure allow us to decompose $f(x)$, or rather its reconstruction in the finest grid, into scales as

$$R(x;\bar{f}^0) = R(x;\bar{f}^L) + \sum_{k=1}^{L} Q_k(x,f) \tag{8}$$

where Q_k, the k-th scale component of $f(x)$ is

$$Q_k(x,f) = R(x;\bar{f}^{k-1}) - R(x;\bar{f}^k). \tag{9}$$

It is easy to see that

$$< Q_k(\cdot,f), \varphi_j^m >= 0 \qquad m \geq k, \tag{10}$$

and for $m = k-1$

$$< Q_k(\cdot,f), \varphi_j^{k-1} >= \bar{f}_j^{k-1} - < R(\cdot;\bar{f}^k), \varphi_j^{k-1} > . \tag{11}$$

We define the k-th scale coefficients $d_j^{k-1}, j = 1, \ldots, N_{k-1}$ as

$$d_j^{k-1} =< Q_k(\cdot,f), \varphi_j^{k-1} >= \bar{f}_j^{k-1} - < R(x;\bar{f}^k), \varphi_j^{k-1} >, \tag{12}$$

thus d_j^{k-1} measures our success in using the reconstruction procedure to predict $\{\bar{f}_j^{k-1}\}$ from our knowledge of $\{\bar{f}_j^k\}$. Notice that (5) implies

$$\sum_l \alpha_l d_{2j+l}^{k-1} = 0 \tag{13}$$

thus, only half of these values are independent. Following [9], we store the k-th scale coefficients with odd indexes and recover the rest form the system of equations (13). Therefore we set

$$\hat{d}_j^k = d_{2j-1}^{k-1} \qquad 1 \leq j \leq N_k, \tag{14}$$

$$d^{k-1} = \mathbf{D}\hat{d}^k \qquad \mathbf{D} \in M_{2N_k \times N_k}. \tag{15}$$

The encoding part of the data compression algorithms described in [9] proceeds by computing first the multiresolution analysis of the original sequence $\{\bar{f}_j^0\}_{j=1}^{N_0}$, $\{\bar{f}^0, \bar{f}^1, \ldots, \bar{f}^L\}$ and then computing the essential k-th scale coefficients:

$$\begin{cases} \text{Do} & k = 1, L \\ \text{Do} & j = 1, N_k \\ \hat{d}_j^k = \bar{f}_{2j-1}^{k-1} - < R(\cdot;\bar{f}^k), \varphi_{2j-1}^{k-1} > \end{cases} \tag{16}$$

At the end of this stage we have obtained the multi-resolution representation of f:

$$f^{MR} = \{\bar{f}^L, (\hat{d}^L, \ldots, \hat{d}^1)\} \tag{17}$$

which, as noted in [9], has the same dimensionality as the original sequence and is completely equivalent to it via the following decoding algorithm:

$$(18)\qquad \begin{cases} \text{Do} \quad k = L, 1 \\ d^{k-1} = \mathbf{D}\hat{d}^k \\ \text{Do} \quad j = 1, N_{k-1} \\ \bar{f}_j^{k-1} =< R(\cdot; \bar{f}^k), \varphi_j^{k-1} > + d_j^{k-1} \end{cases}$$

Data compression can be achieved due to the possible smallness of elements in $(\hat{d}^L, \ldots, \hat{d}^1)$. If f is properly resolved in the k-th grid at a certain location, the coefficients $\hat{d}^l, l = k-1, \ldots, 1$ corresponding to this location (which are just approximation errors) will be small in absolute value and can be set to zero, thus reducing the dimensionality of f^{MR}. Therefore, on the discrete side the multi-resolution analysis allows us to work with a different representation of the original data in which the number of elements can be reduced considerably with very little loss of significant information. Adequate choices of truncation parameters at each level of resolution, together with the stability of the data compression algorithm guarantee the recovery of a high quality reconstruction of the original signal.

3. Multiresolution by Interpolation . We shall pay now special attention to the interpolatory multiresolution analysis of [9] and its associated compression algorithms.

If we take $\varphi(x) = \delta(x)$, (1) becomes

$$(19)\qquad \bar{f}_j^k =< f, \varphi_j^k >= f(x_j^k)$$

and (5)

$$(20)\qquad \bar{f}_j^{k+1} = \bar{f}_{2j}^k.$$

In this case, the sense of different levels of resolution is achieved by sampling $f(x)$ with different frequencies. The conservation property (7) reads

$$(21)\qquad \bar{f}_j^k =< R(\cdot; \bar{f}^k), \frac{1}{h_k}\delta(\frac{x - x_j^k}{h_k}) >= R(x_j^k; \bar{f}^k)$$

which means that the reconstruction $R(x; \bar{f}^k)$ interpolates $f(x)$ on the k-th grid. As in [9], we shall stress these points by changing the notation slightly; we shall use f^k instead of $\bar{f}^k$ and $I_k(x; f)$ instead of $R(x; \bar{f}^k)$, i.e.

$$(22)\qquad R(x; \bar{f}^k) \;=\; I_k(x; f)$$

$$(23)\qquad I_k(x_j^k; f) \;=\; f_j^k$$

We use I_k to emphasize that this framework allows us to choose any interpolation technique, even different techniques for different levels. This proves to be advantageous in many situations and we will see some examples later on.

Data compression algorithms for this type of multiresolution analysis are as follows:
Encoding

$$(24)\qquad \begin{array}{l} Do \quad k = 1, L \\ \quad Do \quad j = 1, N_k \\ \qquad\qquad f_j^k = f_{2j}^{k-1} \\ \qquad\qquad \hat{d}_j^k = f_{2j-1}^{k-1} - I_k(x_{2j-1}^{k-1}; f^k) \end{array}$$

At the end of this stage we have obtained f^{MR}, the interpolating multi-representation of f

$$f^{MR} = \{f^L, (\hat{d}^L, \cdots, \hat{d}^1)\}.$$

Note that $\hat{d}_j^k$ is the error between the value of $f(x)$ and the value of its interpolant, based on the data at the k-th grid, at the point x_{2j-1}^{k-1} , which is the center of the interval $[x_{j-1}^k, x_j^k]$. Obviously, if the interpolation procedure based on the samples of $f(x)$ on the k-th grid gives a good approximation to $f(x)$, the interpolation errors will be small. We compress this representation truncating to zero the interpolation errors that fall below a certain specified tolerance.

$$\text{tr}(x, \epsilon) = \begin{cases} 0 & \text{if } |x| < \epsilon \\ x & \text{otherwise} \end{cases} \tag{25}$$

$$f^{MR} = \{f^L, (\hat{d}^L, \cdots, \hat{d}^1)\} \xrightarrow{\text{tr}} \{f^L, (\tilde{d}^L, \cdots, \tilde{d}^1)\}$$

The decoding part of the algorithm inverts the compressed representation.
Decoding

$$\tilde{f}^L = f^l$$

$$\begin{array}{ll} Do & k = L, 1, -1 \\ \quad Do & j = 1, N_k \\ & \qquad \tilde{f}_{2j}^{k-1} = \tilde{f}_j^k \\ & \qquad \tilde{f}_{2j-1}^{k-1} = \tilde{d}_j^k + I_k(x_{2j-1}^{k-1}; \tilde{f}^k) \end{array} \tag{26}$$

Notice that we have not assumed linearity of the interpolation procedure and therefore the interpolation strategy may depend on the local nature of the data. Thus we can incorporate in these compression algorithms adaptive procedures such as ENO interpolation ([6, 7]).

When we consider data-independent interpolatory techniques, $I_k(\cdot; f)$ is a linear functional of f. In this case, we can associate a multiresolution basis of functions to the representation f^{MR} as follows:

Define

$$\bar{\varphi}_j^k(x) = I_k(x; e_j^k) \qquad 1 \le j \le N_k \tag{27}$$

where e_j^k denotes the unit vector of the k-th grid, i.e.

$$(e_j^k)_i = \delta_{j,i} \qquad 1 \le j \le N_k \tag{28}$$

clearly

$$< \bar{\varphi}_j^k, \varphi_i^k >= \bar{\varphi}_j^k(x_i^k) = \delta_{j,i}. \tag{29}$$

Let $\bar{V}_k$ denote the linear span of $\{\bar{\varphi}_j^k\}$, $0 \le j \le N_k$ and let $\bar{P}_k$ be the interpolatory projection into $\bar{V}_k$, i.e.

$$(\bar{P}_k g)(x) = \sum_{j=1}^{N_k} g(x_j^k) \bar{\varphi}_j^k(x) \tag{30}$$

obviously

$$\bar{P}_k I_k = I_k. \tag{31}$$

If

$$\bar{P}_{k-1} I_k = I_k \tag{32}$$

then

$$Q_k = \bar{P}_{k-1} Q_k = \sum_{j=1}^{N_{k-1}} Q_k(x_j^{k-1}) \bar{\varphi}_j^{k-1} = \sum_{j=1}^{N_k} \hat{d}_j^k \bar{\varphi}_{2j-1}^{k-1}, \tag{33}$$

since

$$Q_k(x_{2j}^{k-1}) = 0.$$

Thus

$$I_0(x; f) = I_L(x; f) + \sum_{k=1}^{L} Q_k(x; f) = I_L(x; f) + \sum_{k=1}^{L} \sum_{j=1}^{N_k} \hat{d}_j^k \psi_j^k \tag{34}$$

where

$$\psi_j^k = \bar{\varphi}_{2j-1}^{k-1}. \tag{35}$$

This representation tells us that reducing the dimensionality of the multi-resolution representation amounts to dropping those basis elements whose contribution is small.

We should notice that the requirement (32) is equivalent to

$$\bar{P}_{k-1} \bar{\varphi}_j^k \equiv \bar{\varphi}_j^k, \tag{36}$$

or in other words, $\bar{\varphi}_j^k \in \bar{V}_{k-1}$. When there is a "mother function" $\bar{\varphi}$ such that

$$\bar{\varphi}_j^k = \bar{\varphi}(\frac{x - x_j^k}{h_k}), \tag{37}$$

(36) implies that $\bar{\varphi}(x)$ must also satisfy a dilation equation.

When the reconstruction is a piecewise linear interpolation

$$\bar{\varphi}_j^k = \bar{\varphi}(\frac{x - x_j^k}{h_k}), \tag{38}$$

where $\bar{\varphi}(x)$ is the hat function

$$\bar{\varphi}(x) = \begin{cases} 1 - |x| & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}.$$

Since the hat function satisfies the dilation equation

$$\bar{\varphi}(x) = \frac{1}{2} \left[\bar{\varphi}(2x - 1) + 2\bar{\varphi}(2x) + \bar{\varphi}(2x - 1) \right],$$

equation 34 holds.

Spectral collocation methods also satisfy (32). To see this, let $\{\eta_n(x)\}$, $1 \leq n < \infty$, be an infinite sequence of linearly independent functions, and let

$$I_k(x;f) = \sum_{n=1}^{N_k} a_n^k \eta_n(x), \qquad a_n^k = a_n^k(f) \tag{39}$$

where the sequence $\{a_k^n\}_{n=1}^{N_k}$ is uniquely determined by the N_k linear equations

$$I_k(x_j^k;f) = \sum_{n=1}^{N_k} a_n^k \eta_n(x_j^k) = f_j^k, \qquad 1 \leq j \leq N_k. \tag{40}$$

Notice that $\bar{P}_{k-1} I_k$ is determined by the solution of

$$I_{k-1}(x_j^{k-1}; I_k(\cdot;f)) = \sum_{n=1}^{N_{k-1}} \tilde{a}_n^{k-1} \eta_n(x_j^{k-1}) = I_k(x_j^{k-1};f), \qquad 1 \leq j \leq N_{k-1}. \tag{41}$$

and that the sequence

$$\tilde{a}_n^{k-1} = \begin{cases} a_n^k & \text{for} \quad 1 \leq n \leq N_k \\ 0 & \text{for} \quad N_k < n \leq N_{k-1} \end{cases} \tag{42}$$

satisfies the above system. Since the solution of (41) is unique, we must have

$$\bar{P}_{k-1} I_k = I_k$$

thus there is a multi-resolution basis of functions associated to the spectral collocation multi-resolution analysis.

The standard way to reduce the dimensionality of spectral approximations is to eliminate components $\eta_n(x)$ for which $a_n^k(f)$ is small in absolute value. Unfortunately, the size of the spectral coefficients depends on the global behavior of f, and its elimination affects the approximation everywhere. Rewriting the spectral approximation in its multi-resolution basis enables us to reduce its dimensionality by neglecting terms ψ_j^k for which $\hat{d}_j^k$, the local approximation error, is small in absolute value.

For purely oscillatory data, the highest compression rate is attained by a Fourier transformation of the original sequence, however, when the original data are oscillatory only in small regions of their domain of definition, the Fourier technique looses efficiency due to the fact that discontinuities and flat regions between oscillations are the result of very delicate and balanced cancellations between a large number of basis functions.

The multi-resolution basis provides a different representation of the original data in which the basis functions have a more local character in time and frequency. Compression algorithms based on spectral collocation might be able to attain better compression rates than regular Fourier collocation algorithms in those cases when the global character of the latter proves to be disadvantageous.

Let us investigate these issues further. Consider the trigonometric interpolant of the set of data $\{f_j^k\}_{j=1}^{N_k}$

$$I_k(x;f) = \frac{1}{2}a_0 + \sum_{n=1}^{N_{k+1}-1} a_n \cos(2\pi n x) + b_k \sin(2\pi n x) + \frac{1}{2} a_{N_{k+1}} \cos(2\pi N_{k+1} x) \tag{43}$$

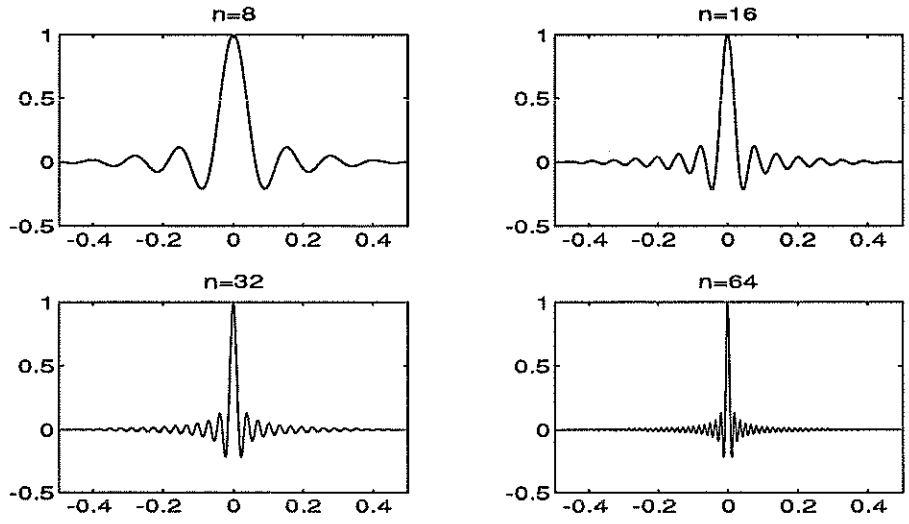


FIG. 1. *Dirichlet Kernels*

Since

$$D_n(x) = \frac{1}{2} + \sum_{l=1}^{n} \cos(2\pi l x) + \frac{1}{2}\cos(2\pi n x) = \frac{\sin(2\pi n x)\cos(\pi x)}{2\sin(\pi x)}, \tag{44}$$

$D_{N_{k+1}}(x)$ is a trigonometric polynomial that satisfies

$$D_{N_{k+1}}(x_j^k) = \delta_{0,j} \cdot N_{k+1} \tag{45}$$

thus, in our notation

$$\bar{\varphi}_0^k(x) = \frac{1}{N_{k+1}} D_{N_{k+1}}(x) \tag{46}$$

$$= \frac{1}{N_k} \frac{\sin(\pi N_k x)\cos(\pi x)}{\sin \pi x} \tag{47}$$

and

$$\bar{\varphi}_j^k(x) = \bar{\varphi}_0^k(x - x_j^k) \tag{48}$$

Notice that

$$\hat{\psi}_j^k(x) = \bar{\varphi}_{2j-1}^{k-1} \tag{49}$$

decays away from x_{2j-1}^{k-1} (see Figure 1); consequently, it is conceivable that the error introduced by dropping it from the expansion will be restricted to a neighborhood of this point. However, the experiments confirm that the multi-resolution spectral base has the same drawbacks as the classical one.

We take the following signal

$$f(x) = \begin{cases} .1\sin(50\pi x) & x \in [.2,.3] \bigcup [.7,.8] \\ 0 & \text{otherwise} \end{cases}$$

and we try to recover it after compressing the sequence $\{f(x_i^0)\}_{i=0}^{N_0}$ with $N_0 = 1024$. We use three different compression algorithms,

1. A multi-resolution analysis based on a centered polynomial interpolation of degree six (see [10]). The tolerance in each level is constant= cf for the truncation operation.
2. A regular spectral representation, compressed keeping only the largest coefficients.
3. A multiresolution analysis based on Fourier interpolation. Tolerance= cf at each level for the truncation operation.

Some experiments are shown in figure 2. We observe that, for a comparable amount of data in each compressed representation, the algorithm based on polynomial interpolation (labeled *polyg*) recovers a signal which is essentially oscillation-free, but both the standard Fourier Collocation and the multi-resolution based on Fourier interpolation recover "wiggles" in the flat region between the oscillations. This clearly indicates that the oscillations on the tail of the basis functions, $\hat{\psi}_j^k$, also contribute in an essential way to the cancellations that produce flat regions. We conclude, thus, that a straightforward application of the multi-resolution analysis based on Fourier interpolation is still inappropriate for the compression of locally oscillatory signals.

4. Multiresolution analysis of Cell-Averages . When the weight function $\varphi(x)$ is the box function

$$\varphi(x) = \begin{cases} 1 & -1 \leq x < 0 \\ 0 & \text{otherwise} \end{cases}$$

the associated multi-resolution analysis

$$\{\{\bar{f}_j^k\}_{j=1}^{N_k}\}_{k=0}^{L}$$

is given by the dilation relation (see [9])

$$\bar{f}_j^{k+1} = \frac{1}{2}(\bar{f}_{2j-1}^k + \bar{f}_{2j}^k). \tag{50}$$

where

$$\bar{f}_j^k =< f, \varphi_j^k >= \frac{1}{h_k}\int_{x_{j-1}^k}^{x_j^k} f(y)dy \tag{51}$$

is the average of $f(x)$ in the j-th cell of the k-th grid. Defining the cell-averaging operator $A(I)$ as

$$A(I)f = \frac{1}{|I|}\int_I f(x)dx,$$

we can write

$$\bar{f}_j^k = A(I_j^k)f \qquad I_j^k = [x_{j-1}^k, x_j^k].$$

The multiresolution representation of the initial data using this multiresolution analysis is obtained as follows:

Encoding

(52)
$$\begin{aligned} &\text{Do} \quad k = 1, L \\ &\quad \text{Do} \quad j = 1, N_k \\ &\qquad \bar{f}_j^k = \tfrac{1}{2}(\bar{f}_{2j}^{k-1} + \bar{f}_{2j-1}^{k-1}) \\ &\qquad \hat{d}_j^k = f_{2j-1}^{k-1} - A(I_{2j-1}^{k-1}) R_k(\cdot; \bar{f}^k) \end{aligned}$$

Decoding

$$\tilde{f}^L = f^l$$

(53)
$$\begin{aligned} &\text{Do} \quad k = L, 1, -1 \\ &\quad \text{Do} \quad j = 1, N_k \\ &\qquad \tilde{f}_{2j-1}^{k-1} = \hat{d}_j^k + A(I_{2j-1}^{k-1}) R_k(\cdot; \tilde{f}^k) \\ &\qquad \tilde{f}_{2j}^{k-1} = 2\tilde{f}_j^k - \tilde{f}_{2j-1}^{k-1} \end{aligned}$$

There are various ways to construct $R(\cdot; \bar{f}^k)$ so that the conservation requirement (7), which in this case means

$$A(I_j^k) R(\cdot; \bar{f}^k) = \bar{f}_j^k, \tag{54}$$

is satisfied. Perhaps the best known technique is the 'reconstruction via primitive function' (see [7, 8]). This technique is based on the observation that the cell averages of a function $f(x)$ give the point values of its primitive

$$F(x) = \int_0^x f(y) dy,$$

since

$$F(x_j^k) = \sum_{i=1}^{j} (h_k \bar{f}_i^k) \qquad 1 \le j \le N_k \qquad F(x_j^0) = 0.$$

The reconstruction is then defined as

$$R_k(x; \bar{f}^k) = \frac{d}{dx} I_k(x; F^k), \tag{55}$$

where $I_k(x; F^k)$ is a polynomial interpolant of the primitive function $F(x)$. Typically, if I_k is an interpolation method with formal order of accuracy $r+1$,

$$I_k(x; F^k) = F(x) + O(\| F^{(r+1)} \| (h_k)^{r+1}).$$

then (see [7, 9])

$$R_k(x; \bar{f}^k) = f(x) + O(\| f^{(r)} \| (h_k)^r).$$

It is easy to check that (55) satisfies the conservation property (54).

A	ω	tol	error	cd
.1	0	.05	.097	41
.1	0	.01	.01	88
.2	0	.01	.008	104
.3	0	.02	.01	100
.1	40	.01	.014	95
.2	40	.02	.028	95

TABLE 1
Point values. r=6

5. Numerical experiments . We apply the compression algorithms described in the previous two sections to signals of the following type:

$$f(x) = \sin(2\pi x) + \Phi(x) \tag{56}$$

where $\Phi(x)$ describes the rapidly oscillatory behavior of the signal and the smooth, slowly varying part is just a periodic sine wave.

The reconstruction procedures used are described in [10]. They correspond to interpolatory polynomials based on a centered stencil of grid points around each grid cell.

Figure 3 displays the outcome of the two compression algorithms superimposed to the original signal, for which

$$\Phi(x) = \begin{cases} .1\cos(100\pi x) & .2 \le x \le .5 \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

we also show the error between the original signal and the signal obtained from the compressed multi-resolution representation. We have intentionally displayed poorly resolved as well as high quality reconstructions to show the limitations of the compression algorithms for this type of data.

In all experiments $n_0 = 10$, $L = 10$. In the figures, tol represents the tolerance for the truncation procedure. For interpolatory compression an adequate policy is to set $\epsilon_k = \text{tol} \quad \forall k$, while for cell average compression the policy is $\epsilon_k = \text{tol}/2^k \quad \forall k$. We refer to [9] for details on these policies.

Also, 'cd' is the number of non-zero data in the compressed representation, and 'max-error' is the $||\cdot||_\infty$ of the depicted error.

We also played with a slightly more general class of 'local oscillations' where the function $\Phi(x)$ can be expressed in the following way

$$\Phi(x) = \begin{cases} A \cdot \cos(100\pi x) + \sin(\omega\pi x) & .2 < x < .5 \\ 0 & \text{otherwise} \end{cases}$$

For comparison purposes, we applied both algorithms to the signal $f(x) = \sin(2\pi x) + \Phi(x)$ for various values of A and ω. Some of the results are compiled in tables 1 and 2

The first rows in each table correspond to the experiment in Figure 3 The second rows correspond to the experiments in Figure 4. Starting with the second row, all compressed representations in both tables, produce a recovered signal that is virtually indistinguishable from the original one, wich gives us an idea of the maximal compression rate that one can obtain with these algorithms for locally oscillatory data.

A	ω	tol	error	cd
.1	0	.3	.057	38
.1	0	.1	.035	79
.2	0	.1	.03	93
.3	0	.1	.046	95
.1	40	.1	.03	83
.2	40	.2	.05	93

TABLE 2
cell averages. r=5

Of course most of the interpolation errors kept correspond to grid points inside the oscillatory region.

The number of data to obtain an acceptable reconstruction appears to be of the same order for both compression algorithms. However, we observe a significant difference in the quality of coarse level reconstructions for these two types of multi-resolution analysis. Observe the data in Figure 5 where we display the original signal, a reconstruction based on the data at a level with 8 points and the error between the two.

In the interpolatory multi-resolution analysis, the low level reconstruction and, consequently, the error depend heavily on the point-valued samples of the original signal, while in the cell average case, the local oscillations 'average out' and we are left purely with the low frequency components of the original signal.

This observation will be used in the following section to design an algorithm that, while still in the multi-resolution setting, combines these techniques with the best compression algorithm for oscillatory signals: the discrete FFT.

6. A Combined Approach. As we mentioned before, the framework in [9] enables us to use different reconstruction techniques at each level of resolution. We shall move one step further: We would like to use different encoding and decoding mechanisms in different regions of the domain at different levels of resolution, depending on the nature of the data.

The basic idea is indeed a simple one: At a low resolution level we sample with very few points thus only the large scale components of the signal remain. Subtracting the original signal from its low resolution reconstruction will produce an essentially flat signal with the oscillations of the original one. A graphic demonstration of the above is given in Figure 5. The original signal of Figure 3 is sampled with 1024 points, i.e. $n_0 = 10$ and the low level reconstruction is based on 8 points.

The most economical representation of a purely oscillatory signal is its spectral representation. However, we have learned in section 3 that a "global" spectral representation of the error is not well suited for compression purposes, due to the flat regions between the oscillations. However, the spectral representation of *only* the oscillatory part of the error would lead to optimal compression rates in this part of the domain. Notice that by Fourier transforming the local oscillations of the error we largely avoid the problems deriving of lack of periodicity.

Taking into account the above considerations, we propose the following algorithm:

1. Determine and label the limits of each one of the oscillatory intervals. For simplicity, we shall assume that the oscillatory region is composed of just one interval: $I_O = [y, z]$. In general, one has an oscillatory region which is the

union of a finite number of oscillatory intervals. The extension to this general case is self-evident.

2. Compute the multi-resolution representation of $\{f^0\}$
3. Compress this representation. For this purpose we introduce a special truncation operation:

$$\tilde{d}_j^k = str(\hat{d}_j^k; \epsilon_k) = \begin{cases} 0 & \text{if } |\hat{d}_j^k| \leq \epsilon_k \\ 0 & \text{if } x_{2j-1}^{k-1} \in I_O \text{ and } 1 \leq k \leq KC \\ \hat{d}_j^k & \text{otherwise} \end{cases} \tag{58}$$

The user decides a priory what level is to be considered "the low resolution level", KC.

4. Compute the error between the reconstruction at level KC, $R_{KC}(x) = R(x; \bar{f}^{KC})$ (which is supposed to be oscillation free) and the original signal.

$$E_j = \bar{f}_j^0 - R_{KC}(x_j^0) \tag{59}$$

5. Compute the Discrete Fourier Transform of the set of values $\{E_j\}_{j=iy}^{iz}$, where $x_{iy}^0 = y$ and $x_{iz}^0 = z$ are the endpoints of the oscillatory interval. One has then the equivalent set $\{\hat{E}_j\}_{j=iy}^{iz}$, such that if

$$E(x) = \sum_{iy}^{iz} \hat{E}_j \nu_j(x) \qquad x \in I_O$$

then

$$E(x_j^0) = E_j \qquad x_j^0 \in I_O.$$

6. Compress this representation as follows:

$$\tilde{E}_j = tr(\hat{E}_j, \epsilon_j^f) = \begin{cases} 0 & \text{if } |\hat{E}_j| < \epsilon_j^f \\ \hat{E}_j & \text{otherwise} \end{cases} . \tag{60}$$

This is just a truncation procedure in which the truncation parameters may depend on the frequency of the corresponding base element. One can also quantize the set $\{\hat{E}_j\}$ to reduce its size.

At the end of this stage we have the following representation of $\{f^0\}$:

$$f^{FMR} = (f^L, (\tilde{d}^1, \ldots, \tilde{d}^L), \tilde{E}_j) \tag{61}$$

whose cardinality is slightly larger than that of the original signal. In fact, since $\tilde{d}_j^k = 0$ when $x_{2j-1}^{k-1} \in I_O$, the number of nonzero elements in f^{FMR} is at most $2^{n_0} + [(z-y) \cdot 2^{m-KC}]$, where $[\cdot]$ stands for integer part, but for the type of signal under investigation, most of these coefficients should be very small, so they would be set to zero in the truncation process.

We describe now the decoding procedure for multi-resolution based on cell averages. With the appropriate changes, we can adapt the algorithm to work with interpolatory multi-resolution instead.

$$\tilde{f}^L = \bar{f}^L$$

Do $k = L, 1, -1$

$$\begin{aligned} \tilde{f}^{k-1}_{2j-1} &= \tilde{d}^k_j + A(I^{k-1}_{2j-1})R_k(\cdot, \tilde{f}^k) \quad 1 \le j \le N_k \\ \tilde{f}^{k-1}_{2j} &= 2\tilde{f}^k_j - \tilde{f}^{k-1}_{2j-1} \qquad\qquad\quad 1 \le j \le N_k \end{aligned}$$

(62)

$k = 1$

Do $j = 1, N_0$

If $x^0_j \in I_0$

$$\tilde{f}^0_j = R_{KC}(x^0_j) + \sum \tilde{E}_j \nu_j(x^0_j)$$

endif

Let us carry out several numerical experiments testing the performance of the above algorithm. We consider again locally oscillatory signals of the form (56) with $\Phi(x)$ as in (57). As in the previous section, we start with $A = .1$ and $\omega = 0$. Using the combined algorithm with the cell average multi-resolution, we get optimal compression. Figure 6 shows the recovered signal superimposed to the original one and the error between the two, shifted down for displaying purposes.

The legends in the figures should be interpreted as follows:

- nzd is the number of non zero d's, ie, the number of non zero interpolation errors ($\tilde{d}^k_j$) after step 3.
- nzfc is the number of non zero fourier coefficients after step 6.
- cd is the total number of non-zero data in the compressed representation, f^{FMR}, i.e.

$$\text{cd} = 2^{n_0 - L} + \text{nzd} + \text{nzfc}$$

We have taken $\epsilon_k = 1/2^k$, which is appropriate for the slowly varying part, and $\epsilon^f_j = \epsilon_f = .08$. Only one Fourier coefficient is kept in the compressed representation (as it should be, since only one frequency is present) and the recovered signal is indistinguishable form the original one. Observe that there is a relatively large error (of the size of the oscillations) at the left endpoint of the oscillatory region. This is due to some imprecision in determining computationally the endpoints of the oscillatory region. We have depicted here one of the best possible scenarios, xb=.5 is a grid point in each mesh of the multi-resolution set-up, thus the cell average framework treats it almost perfectly. In this experiment, the computational oscillatory interval starts and ends at the points in the original grid which are closest to $xa = .2$ and $xb = .5$. Missing these points by a larger amount leads to the same type of end-point errors, now over a larger region, and the same type of recovered signal.

Figure 7 displays the original and recovered signal for $A = .3$ and $\omega = 0$ as well as the error. The tolerances are as before and the results are also comparable. Missing the enpoints now, leads, of course, to end-point errors of larger magnitude (the magnitude of the local oscillations).

The combined approach with the interpolatory multiresolution proves to be less robust, even in this 'best' scenario. Figures 6 and 7 show the reconstructed signals and the errors corresponding to $A = .1$ and $A = .3$ and $\omega = 0$. The inaccuracies in the low level reconstruction, which are due to the sampling process, lead to recovered signals with less 'visual' quality, although the maximum norm of the error is comparable (and in both cases attained close to the endpoints of the oscillatory interval). The number

of data in the compressed representation is also more sensitive to the amplitude of the local oscillations, or to the number of frequencies present. On the other hand, the cell average combined approach keeps its properties when various frequencies are present. In figure 8, $A = .2$ and $\omega = 40$.

7. Discontinuous signals. Suppose that $f(x)$ has $(p-1)$ continuous derivatives and $f^{(p)}(x)$ is discontinuous but bounded. If we want to reconstruct f from its point values on a grid with step-size h, the maximal accuracy that can be achieved $O(h^p)$. The same can be said about reconstructions of f from the set of its cell-averages on the same grid. Thus there is no difference between these two multi-resolution algorithms for continuous data.

However, there is a significant advantage, in using cell-averages rather than point values when $f(x)$ is discontinuous at a finite number of points. To see this let us assume that $f(x)$ is discontinuous at $x_d \in (x_{j-1}^k, x_j^k)$ and that in $[a, x_d) \bigcup (x_d, b]$, $0 \leq a < x_d < b$, f has $(p-1)$ continuous derivatives while $f^{(p)}$ is discontinuous but bounded, $p \geq 1$. Let I^L and I^R denote interpolation of $F(x)$ at grid points in $[a, x_d)$ and $(x_d, b]$ respectively. We note that $F(x)$ is continuous in $[a, b]$, but has a discontinuous derivative at x_d. Consequently, if $F(x)$ is properly resolved on the k-th grid $I^L(x, F^k)$ and $I^R(x, F^k)$ will intersect at some point $\tilde{x}_d \in I_j^k$. Using interpolation with $r \geq p$ we get that this point is a good approximation to the location of the discontinuity within the cell I_j^k, in fact,

$$\tilde{x}_d - x_d = O(h_k^p).$$

On the other hand, the knowledge of the point values $\{f(x_j^k)\}$ in $[a, b]$ conveys no information about the location of a discontinuity within a cell.

We recall here that in the multi-resolution representation, data compression is achieved by dropping the small interpolation errors. Choosing cell-averages and ENO reconstruction with Subcell Resolution ([6, 9]) we can obtain absolute compression of piecewise polynomials with polynomial degree $p < r$, the order of accuracy of the reconstruction. In general, the ENO reconstruction with Subcell Resolution is the only reconstruction technique that satisfies (6) in each cell in the presence of discontinuities, thus it is the best choice for compression of discontinuous signals.

In the previous section we have designed an algorithm that combines a multi-resolution analysis of a signal and a spectral decomposition of its oscillatory behavior. The idea behind the "combined approach" is that one should apply to each portion of the signal the algorithm that gives the best compression rates. Thus, if the original signal is discontinuous at a finite number of points, one should combine reconstruction techniques of ENO with Subcell Resolution type with a fast fourier transform of the oscillations.

Numerical Experiments: Discontinuous signals

Let us compress now slowly varying data, locally oscillatory and discontinuous at a finite number of points using our "combined approach". The basic algorithm is the one we have specified in the previous section. The multiresolution analysis corresponds to cell averages and we use the ENO reconstruction with Subcell Resolution, as described in [6] as our reconstruction procedure. The decoding procedure (62) should be modified accordingly following (52), (53).

We perform numerical experiments on the following signal:

$$f(x) = \begin{cases} \sin(\pi x) & & & 0 \le x < .2 \\ \sin(\pi x) & + & A \cdot \cos(100\pi x) + \sin(\omega \pi x) & .2 \le x < .6 \\ -\sin(\pi x) & - & A \cdot \cos(100\pi x) + \sin(\omega \pi x) & .6 \le x < .8 \\ -\sin(\pi x) & & & .8 \le x < .1 \end{cases}$$

which consists of local periodic perturbations of a discontinuous but otherwise smooth function. Figures 9 and 10 display some of our results. In Figure 9 $A = .1$ and $\omega = 0$. In figure 10 $A = .1$ and $\omega = 40$.

Summary: By combining a spectral Fourier representation of oscillatory regions with a multi-resolution analysis of smooth regions of a given signal, we have been able to recover good quality approximations to the original data with minimal storage requirements.

Cell average multi-resolution techniques based on centered reconstructions (see [10]) lead to near optimal compression algorithms for continuous locally oscillatory data. For discontinuous signals we use the information about the location of the discontinuity contained in the cell averages to apply the Subcell Resolution technique on reconstructions of ENO type.

The emphasis in this report has been on the properties, possibilities and limitations of these algorithms with respect to data compression. It appears that it is possible to combine the compression properties of multi-resolution techniques and Fourier algorithms. This opens a field of experimentation on the design of fast algorithms for the solution of integral equations with oscillatory kernels.

REFERENCES

[1] P. Auscher, G. Weiss and M. V. Wickerhauser, *Local Sine and Cosine Bases of Coifman and Meyer and the Construction of Smooth Wavelets,* in Wavelets- A Tutorial C.K. Chui (ed.) Academic Press,(1991)

[2] G. Beylkin, R. Coifman and V. Rokhlin, *Fast Wavelet Transforms and Numerical Algorithms I,* Comm. Pure and Applied Math., v. 64 (1991), pp. 141-183 .

[3] R. Coifman, Y. Meyer, *Remarques sur l'analyse de Fourier à fenêtre,* sèrie I,C.R. Acad. Sci. Paris 312 (1991), 259-261.

[4] B. Engquist, S. Osher, S. Zhong, *Fast Wavelet Algorithms for Linear Evolution Equations,* ICASE Report # 92-14

[5] F. Arandiga, V. Candela, R. Donat, *Fast Multiresolution algorithms for Linear Evolution Equations: A Comparative Study,* CAM Report 92-52.

[6] A. Harten, *ENO schemes with Subcell Resolution,* J. Comp. Physics, v. 83 (1989), pp. 148-184; also ICASE Report # 87-56

[7] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, *Some results on Uniformly High Order Accurate Essentially Non-oscillatory Schemes* , Applied Numerical Mathematics No. 2, North-Holland, 1986, pp. 347-377.

[8] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, *Uniformly High Order Accurate Essentially Non-oscillatory Schemes III,* J. Comp. Physics, v. 71, (1987) pp. 231-303.

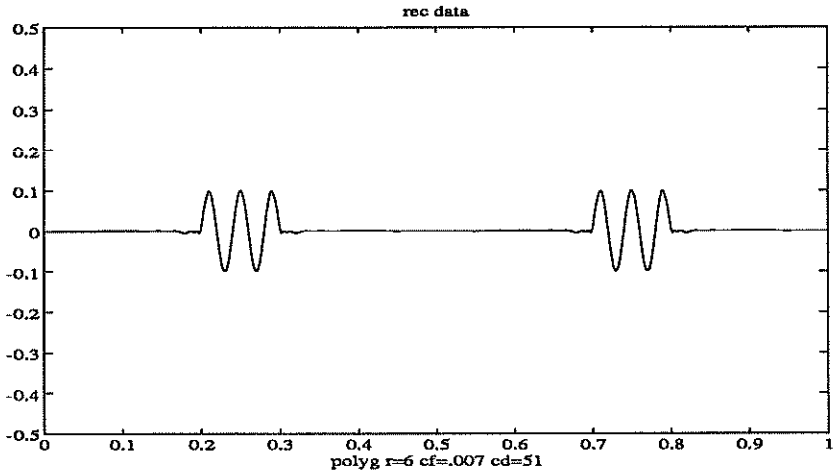
[9] A. Harten, *Discrete Multi-Resolution Analysis and Generalized Wavelets,* J. of Applied Num. Math. v. 12, pp. 153-193 (1993); also UCLA CAM Report 92-08, 1992.

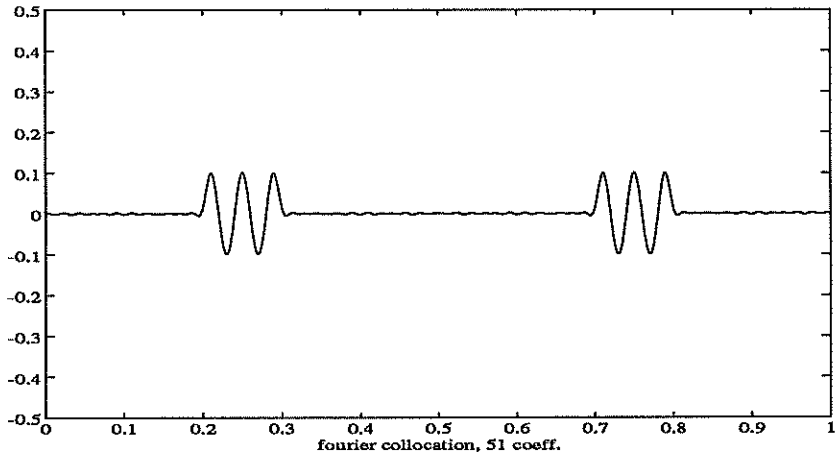
[10] A. Harten I. Yad-Shalom, *Fast Multiresolution Algorithms for Matrix-Vector Multiplication,* To appear in SINUM; also UCLA CAM Report 92-08, 1992.

[11] A. Harten, *Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws,* UCLA Computational and Applied Mathematics Report 93-03 (1993)

[12] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* $L^2(R)$, Trans. Amer. Math. Soc. (1989)

[13] D. Mizrachi, *Removing Noise from Discontinuous Data,* M. Sc. Thesis (1991) Tel-Aviv University.


rec data
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
polyg r=6 cf=.007 cd=51


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
fourier collocation, 51 coeff.

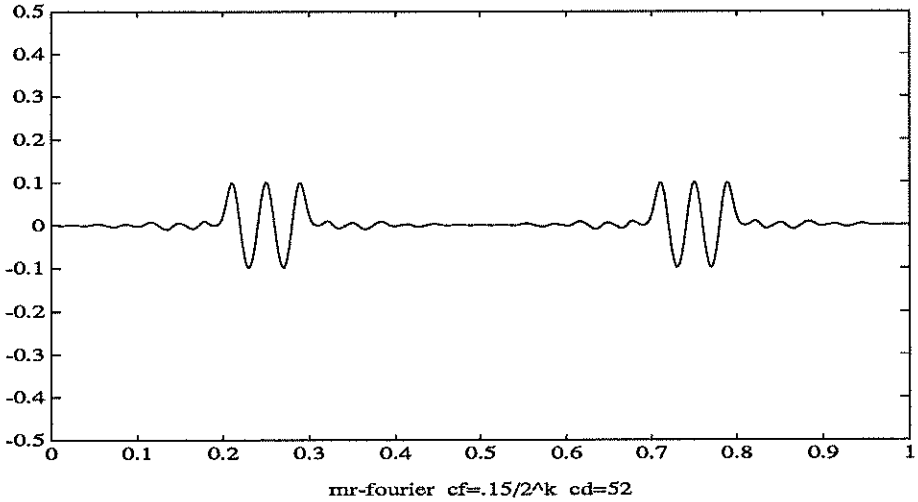
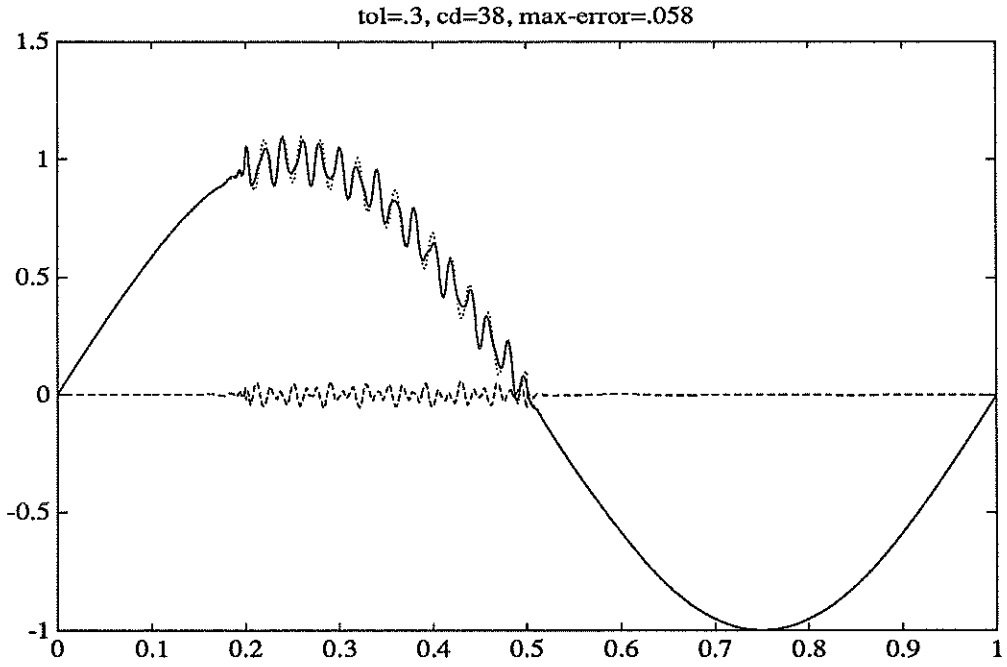
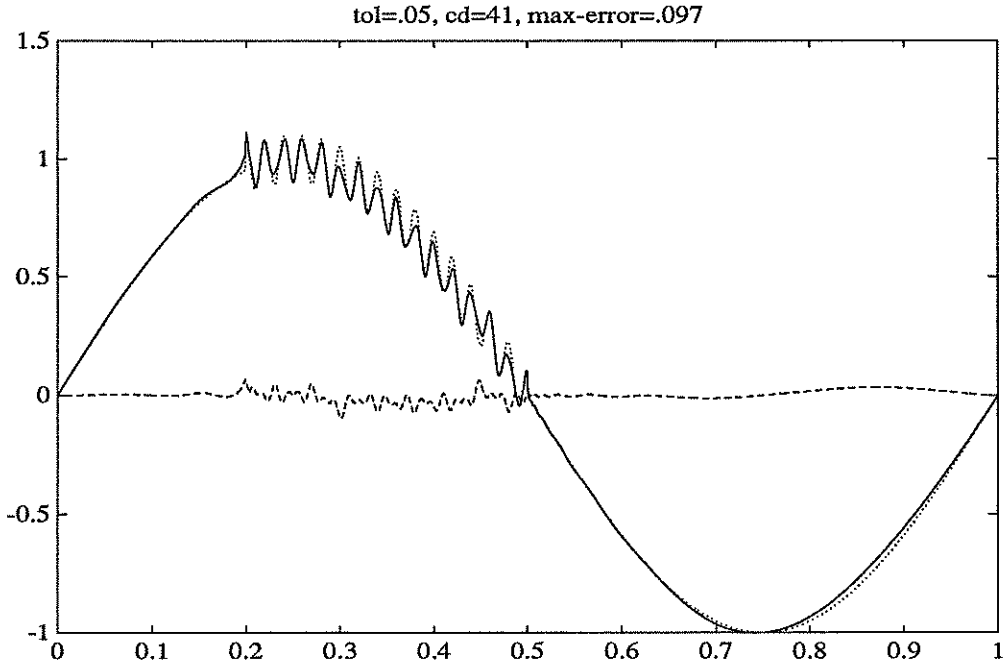

0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
mr-fourier cf=.15/2^k cd=52

FIG 2.

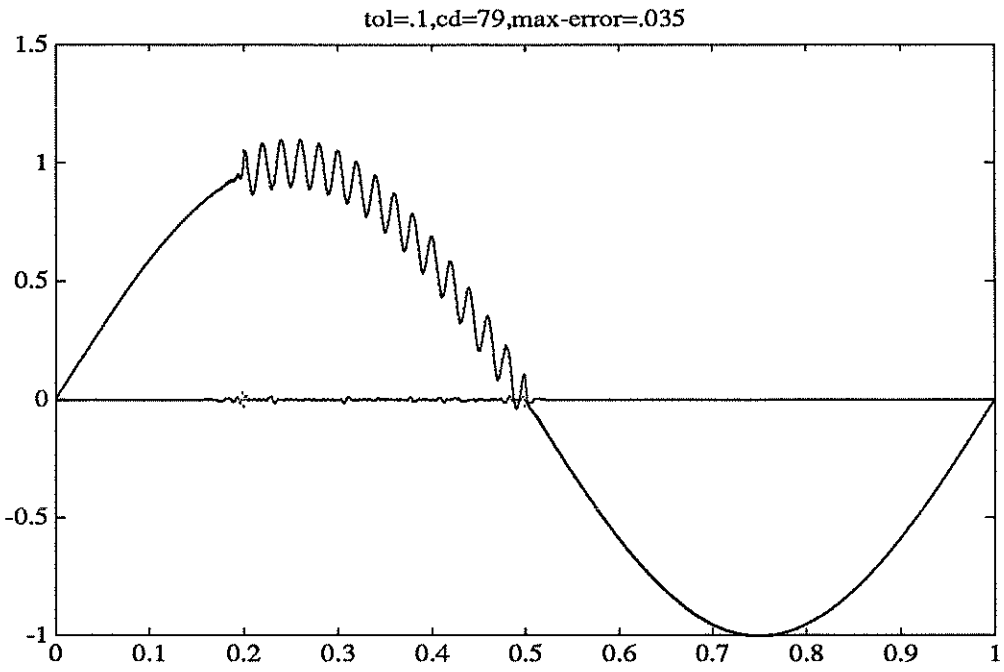


reconstructed, original and error, cell-averages

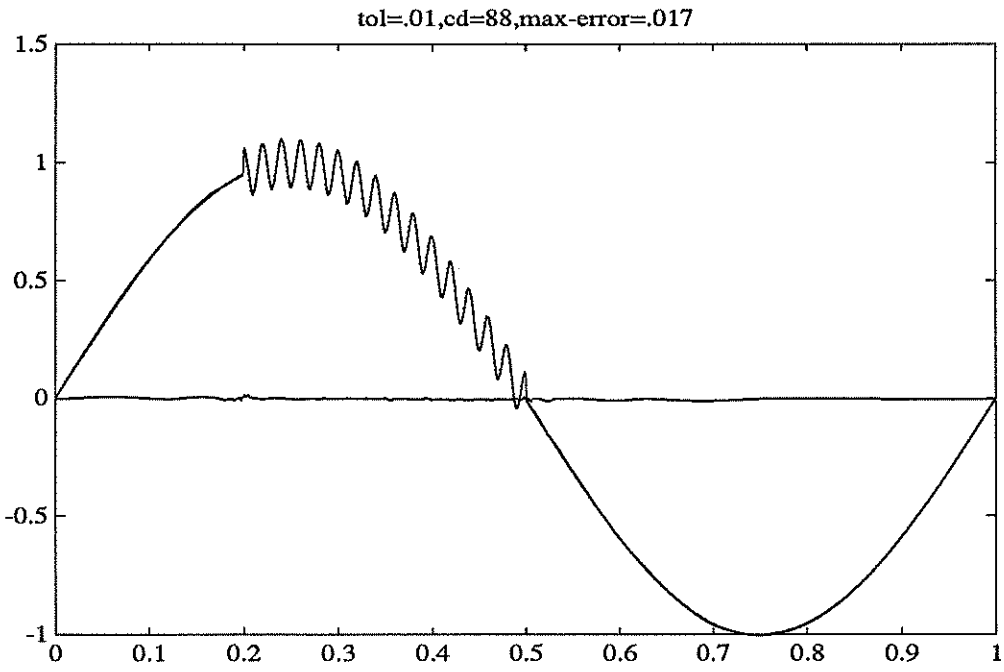


reconstructed, original and error, point values

FIG. 3.

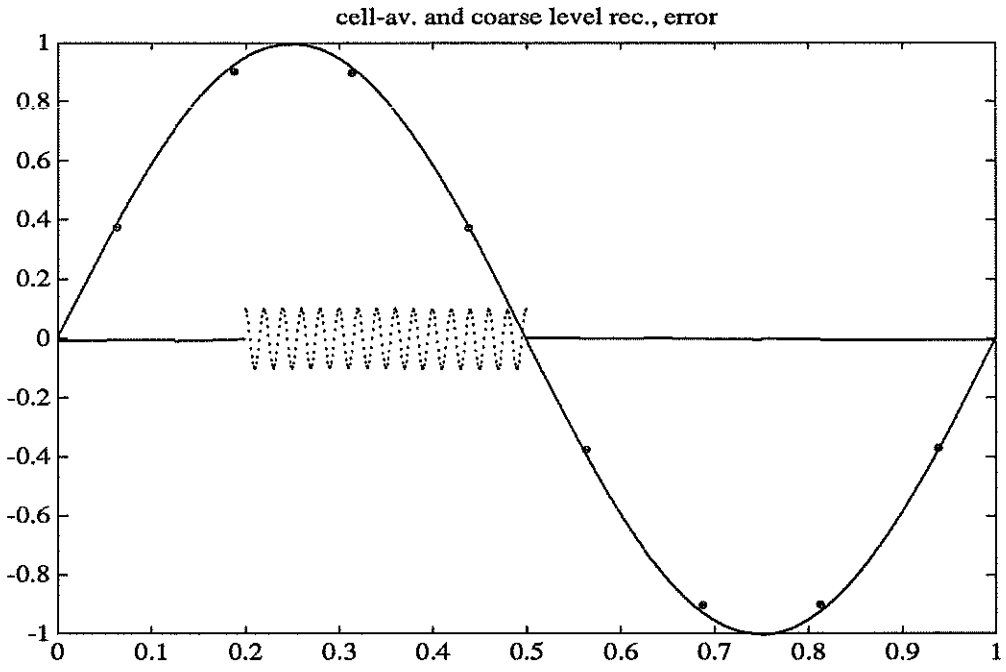


reconstructed, original and error, cell-averages



reconstructed, original and error, point values

FIG. 4.


cell-av. and coarse level rec., error
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

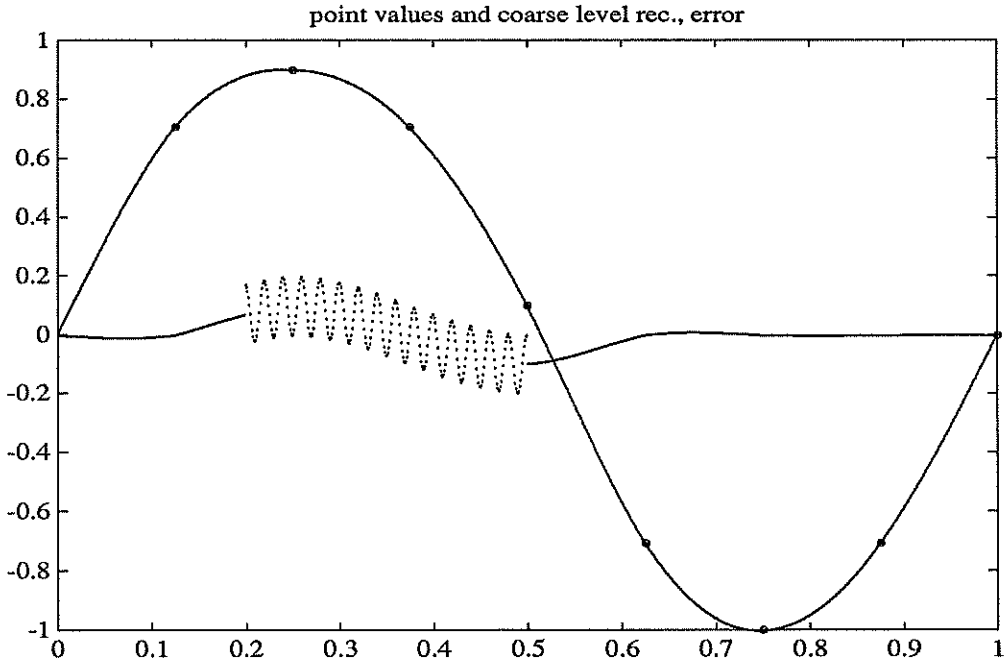
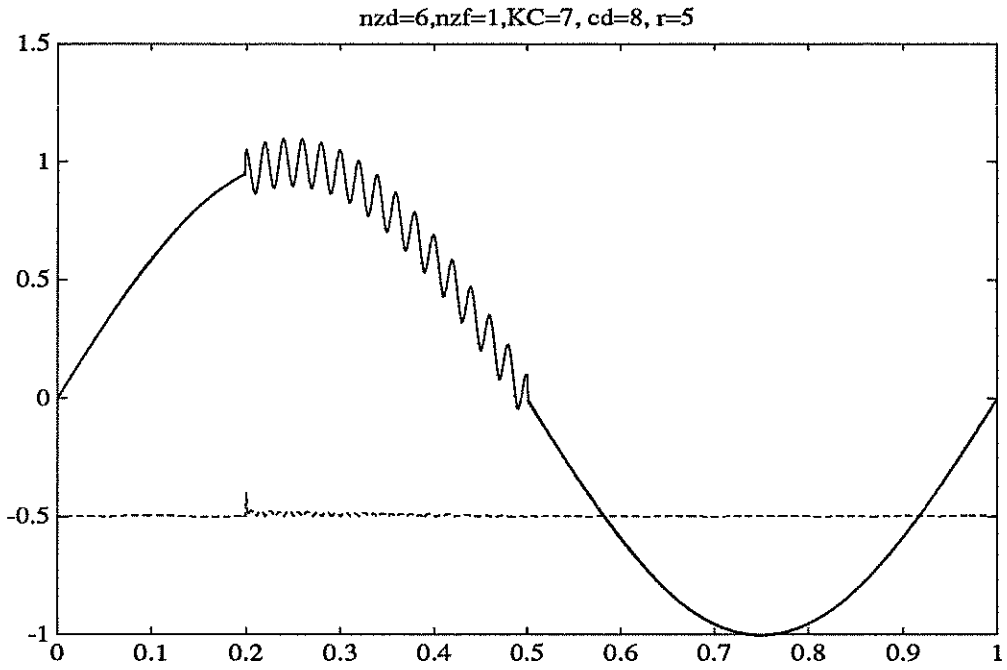
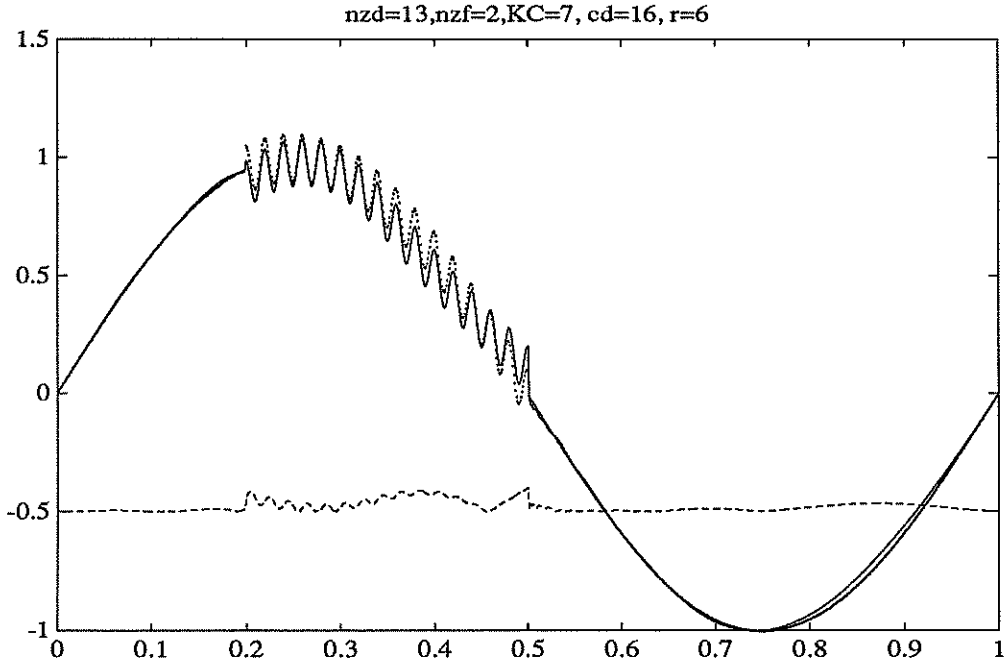

point values and coarse level rec., error
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

FIG. 5.

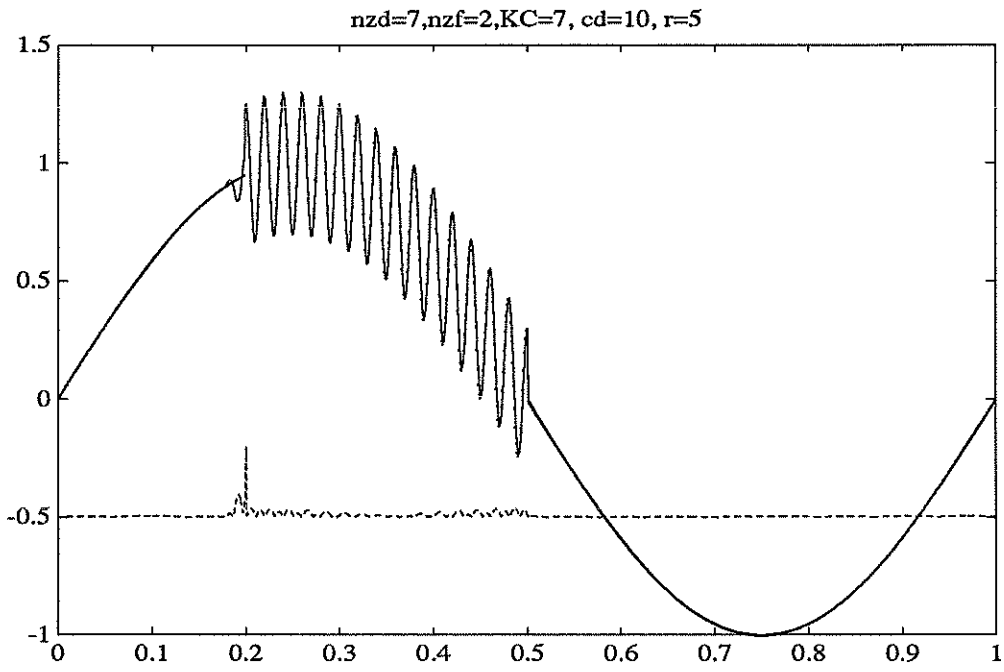


reconstructed, original and error. cell averages

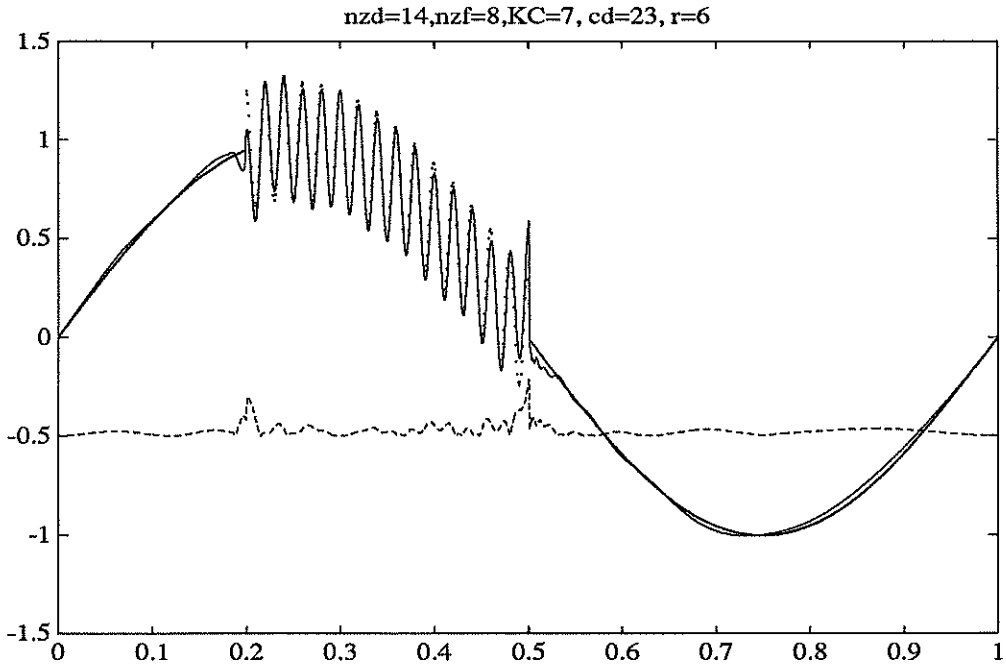


reconstructed, original and error. point values

FIG. 6.

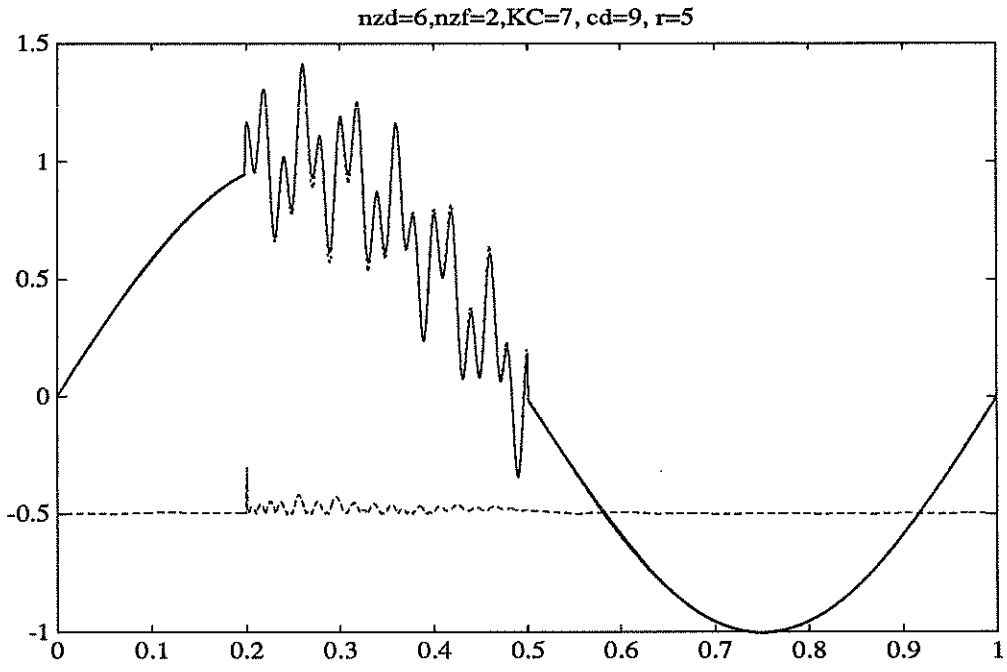


reconstructed, original and error. cell averages

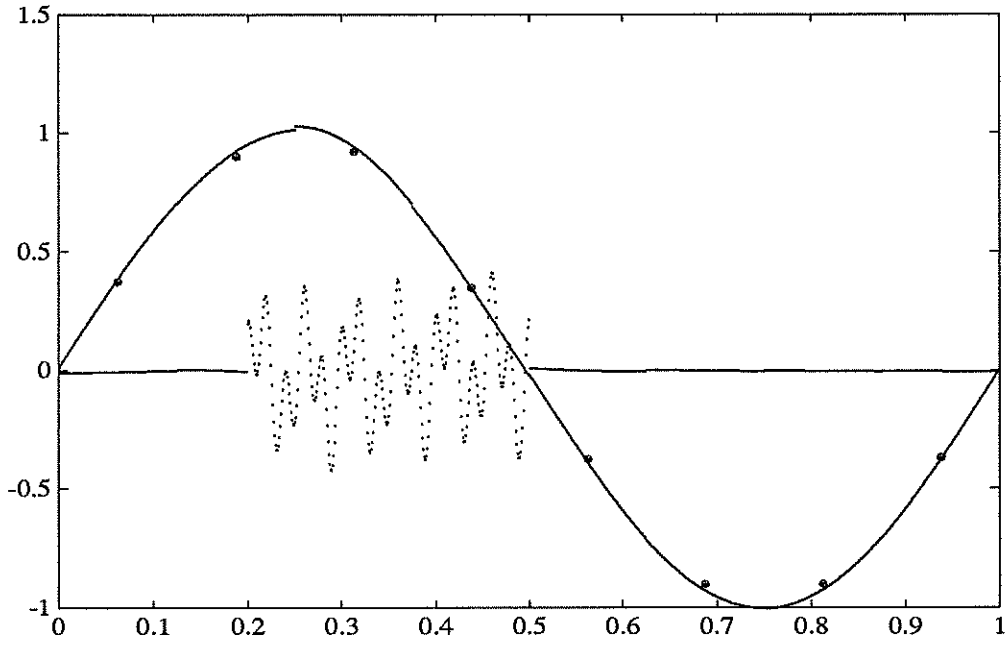


reconstructed, original and error. point values

FIG. 7.

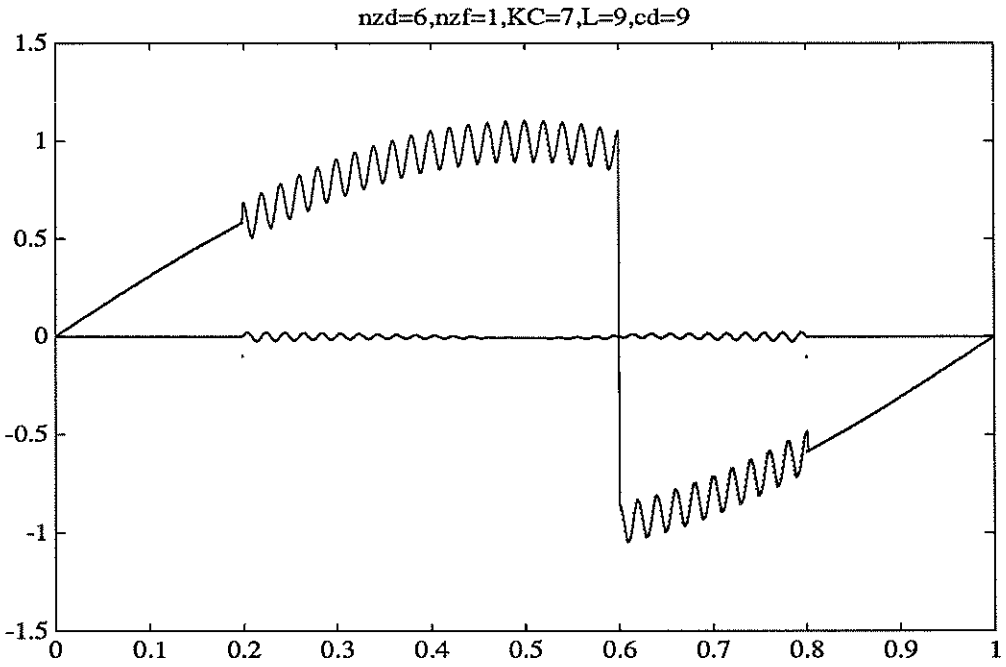


reconstructed, original and error. cell averages

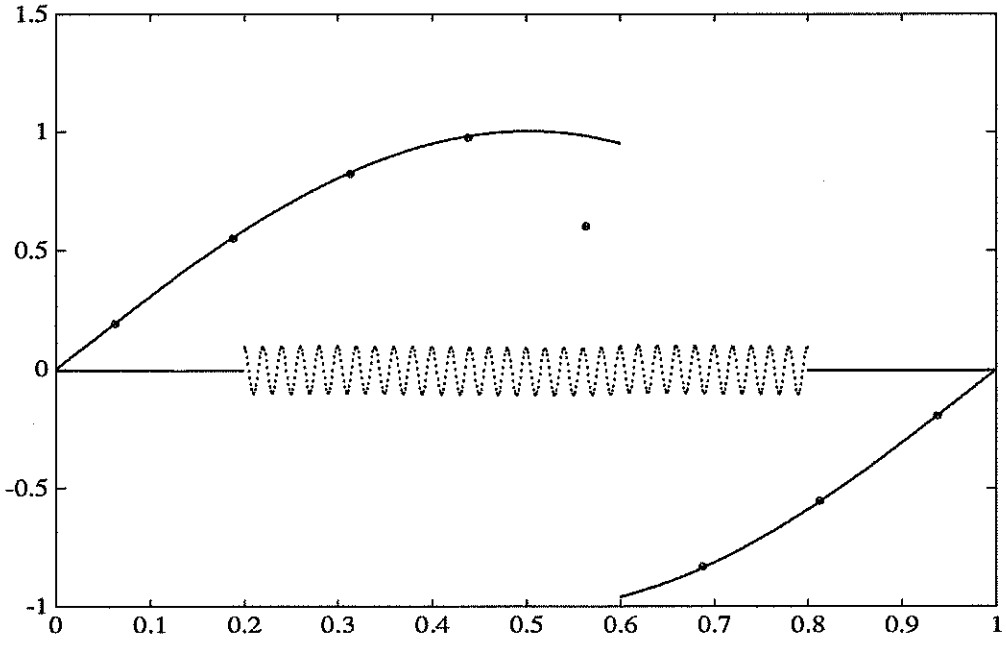


coarse level reconstruction and error

FIG. 8.

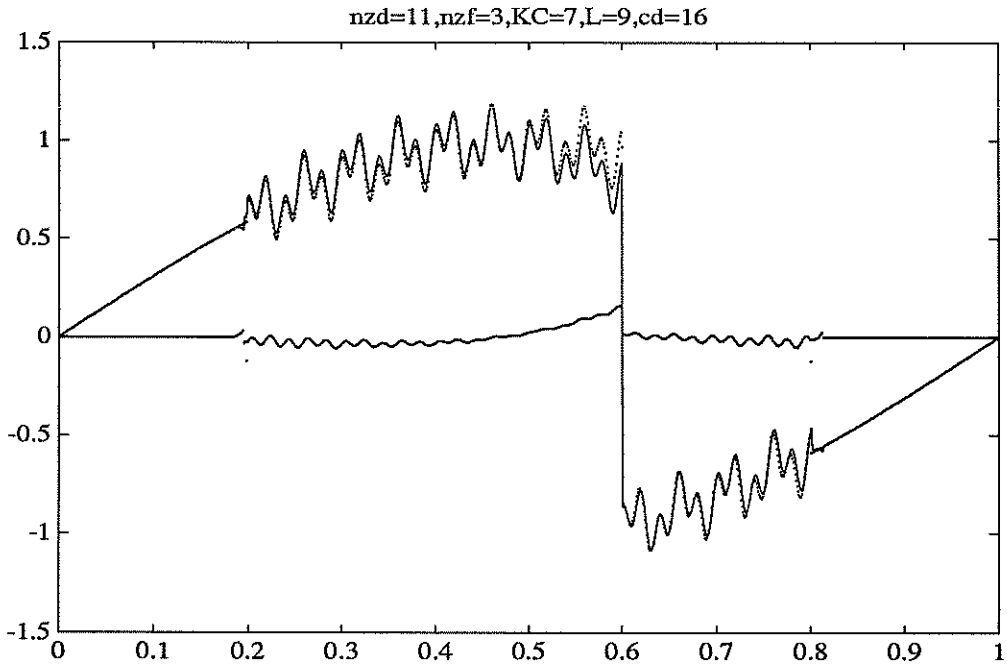


sr-eno3 original, reconstructed and error

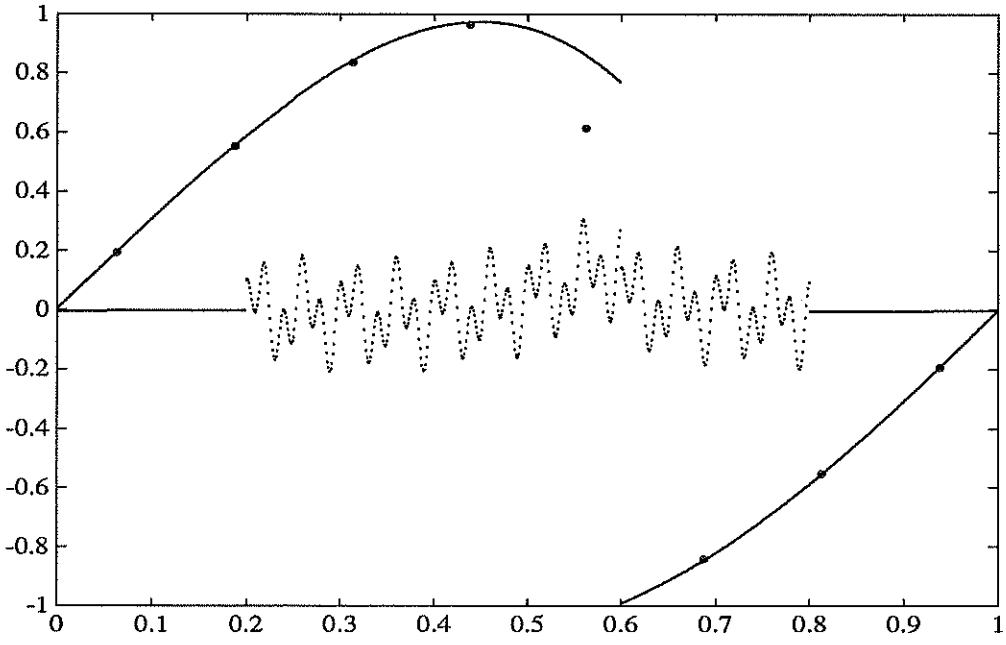


low level reconstruction and error

FIG. 9.



sr-eno3 original, reconstructed and error



sr-eno3 original, reconstructed and error

FIG. 10.