UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws

Ami Harten

March 1993

CAM Report 93-03

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

Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws

Ami Harten

Courant Institute of Mathematical Sciences

New York University

and

School of Mathematical Sciences

Tel-Aviv University

This research was supported by DOE grant DE-FG02-88ER25053 at the Courant Institute of Mathematical Sciences, NYU, and by grants ONR-N00014-92-J-1890 and NSF-DMS91-03104 at UCLA.

Abstract

Given any scheme in conservation form and an appropriate uniform grid for the numerical solution of the initial value problem for one-dimensional hyperbolic conservation laws we describe a multiresolution algorithm that approximates this numerical solution to a prescribed tolerance in a more efficient manner. To do so we consider the grid-averages of the numerical solution for a hierarchy of nested diadic grids in which the given grid is the finest, and introduce an equivalent multiresolution representation. The multiresolution representation of the numerical solution consists of its grid-averages for the coarsest grid and the set of errors in predicting the grid-averages of each level of resolution in this hierarchy from those of the next coarser one. Once the numerical solution is resolved to our satisfaction in a certain locality of some grid, then the prediction errors there are small for this particular grid and all finer ones; this enables us to obtain data compression by setting to zero small components of the representation which fall below a prescribed tolerance. Therefore instead of computing the time-evolution of the numerical solution on the given grid we compute the time-evolution of its compressed multiresolution representation. Algorithmically this amounts to computing the numerical fluxes of the given scheme at the points of the given grid by an hierarchical algorithm which starts with the computation of these numerical fluxes at the points of the coarsest grid and then proceeds through diadic refinements to the given grid. At each step of refinement we add the values of the numerical flux at the center of the coarser cells. The information in the multiresolution representation of the numerical solution is used to determine whether the solution is locally well-resolved. When this is the case we replace the costly exact value of the numerical flux with an accurate enough approximate value which is obtained by an inexpensive interpolation from the coarser grid. The computational efficiency of this multiresolution algorithm is proportional to the rate of data compression (for a prescribed level of tolerance) that can be achieved for the numerical solution of the given scheme.

1. Introduction

In this paper we present a class of multiresolution algorithms for the numerical solution of the initial value problem for hyperbolic conservation laws in one space dimension

$$w_t + f(w)_x = 0, \quad w(x,0) = w_0(x). \tag{1.1}$$

Here $w(x,t)$ is a vector of q components and we assume that the Jacobian $\partial f/\partial w$ has q real eigenvalues $a_1 \leq \cdots \leq a_q$, and that the corresponding system of right-eigenvectors $\{r_1, \ldots, r_q\}$ spans $\mathbb{R}^q$. To simplify our presentation let us assume that $w_0(x)$, and consequently $w(x,t)$ are periodic with a period 1.

Let $x_j = j \cdot h$, $0 \leq j \leq N$, be a uniform partition of $[0,1]$ into N intervals of size h. We consider the numerical solution of (1.1) by the explicit conservative scheme

$$v_j^{n+1} = v_j^n - \lambda(\bar{f}_j - \bar{f}_{j-1}), \ \lambda = \tau/h, \tag{1.2a}$$

where

$$\bar{f}_i = \bar{f}(v_{i-K+1}^n, \ldots, v_{i+K}^n) \tag{1.2b}$$

is the numerical flux of the scheme. Here v_j^n is an approximation to the average of the solution $w(x,t)$ in the cell $[x_{j-1}, x_j]$ at time $t_n = n\tau$, i.e.

$$v_j^n \approx \frac{1}{h}\int_{x_{j-1}}^{x_j} w(x,t_n)dx. \tag{1.3}$$

The numerical flux function $\bar{f}(u_1, \ldots, u_{2K})$ is a function of $2K$ arguments which is consistent with the flux $f(w)$ in (1.1) in the sense that

$$\bar{f}(u, u, \ldots, u) = f(u). \tag{1.4a}$$

$\bar{f}_j$ in (1.2) is an approximation to the time average of the flux of the solution $f(w(x,t))$ in $[t_n, t_{n+1}]$ at x_j

(1.4b)
$$\bar{f}_j \approx \frac{1}{\tau} \int_{t_n}^{t_{n+1}} f(w(x_j,t))dt$$

(see [11] for more details).

We consider now a situation where the solution $w(x,t)$ is highly nonuniform in its behavior as a function of x, i.e. it varies strongly in some subintervals of $[0,1]$ but is a lot smoother in the rest of the interval. If we use a uniform grid with spacing h sufficiently small so that the strongest variation of the solution is resolved to our satisfaction, then the numerical solution is necessarily over-resolved in some parts of the computational domain. In order to improve the efficiency of the numerical solution in this situation it is a common practice to use an adaptive nonuniform grid in which the spacing of the grid is dynamically adjusted to the local variation of the solution. In this paper we present a multiresolution alternative to the adaptive grid methodology in which we perform the uniform fine-grid computation to a prescribed accuracy but reduce the number of arithmetic operations and computer memory requirements to the level of an adaptive grid computation. The main advantage of the multiresolution approach over that of an adaptive grid is its simplicity of programming.

In [4] Daubechies combined Mallat's multiresolution analysis [14] with the construction of a compactly supported mother wavelet function $\psi(x)$ to obtain an orthonormal basis for L_2-functions

(1.5a)
$$u(x) \approx \sum_{j,k} d_j^k(u)\psi_j^k(x), \quad \psi_j^k(x) = 2^{-\frac{k}{2}}\psi(2^k x - j)$$

where the coefficients $d_j^k(u)$

(1.5b)
$$d_j^k(u) = \langle u, \psi_j^k \rangle = \int u(x)\psi_j^k(x)dx,$$

measure the size of the k-th scale component of u in the neighborhood of $x = 2^{-k}j$. This representation leads to a data compression algorithm by discarding coefficients $d_j^k(u)$ that fall below a suitable tolerance. Since the transformation between the function and its wavelet representation is computationally fast, the attractive proposition of performing numerical operations in their multiresolution representation becomes computationally feasible. Beylkin, Coifman and Rokhlin [3] used the wavelet transform to design fast multiresolution algorithms for matrix-vector multiplication and showed that it is an $O(N \log N)$ algorithm in the case of Calderon-Zygmund operators and pseudo-differential operators. Engquist, Osher and Zhong [5] have used [3] to obtain fast algorithms for the time-evolution of solutions to linear hyperbolic and parabolic initial value problems. Liandrat and Tchamitchian [12] and Madday and Ravel [13] have used wavelets to derive a Galerkin-type scheme for the computation of the multiresolution representation of the solution to the viscous Burgers equation. Barcy, Mallat and Papanicolau [2] developed a wavelet based space-time adaptive methods for parabolic and hyperbolic problems, including Burgers equation.

In [8] and [10] we used Mallat's multiresolution analysis to design a general class of data compression algorithms which includes Daubechies' algorithm as a particular case. Two subclasses of such algorithms are described in the present paper: In Section 2 we describe multiresolution representations which are suitable for applications where the solution is discretized by pointvalues and in Section 3 we do the same for discretization by cell-averages. In Section 4 we show how to use the multiresolution representation of cell-averages in order to obtain data compression.

The richness of this class enables us to outperform the wavelet based algorithms in many applications. In [10] we design general fast multiresolution algorithms for matrix-vector multiplication and show that they are more efficient than the fast wavelet transform of [3]. In [1] Arandiga, Candela and Donat use the multiresolution representation of [10] to design fast algorithms for the time evolution of solutions to linear hyperbolic and parabolic initial value problems and show that they compare favorably to the wavelet based algorithm of [5].

In [7] we extended the notion of multiresolution analysis to nested unstructured grids. Given cell-averages of a function on the finest grid we show how to determine the minimal level of resolution which is needed at each locality in order to approximate the finest grid data to a prescribed accuracy. The computation of the numerical solution of hyperbolic conservation laws by a Golunov-type scheme on any given grid amounts to the evaluation of numerical fluxes at the boundaries of the cells. Our goal is to reduce the number of numerical flux evaluations to what is actually needed in order to resolve the solution to our satisfaction. The numerical flux of a Golunov-type scheme is an approximation to the time-average of the *normal* component of the flux of the solution along the boundary of the cell. If the grid is unstructured, then the normal direction at the boundaries of the cells, and consequently the numerical flux, cannot be thought of as discrete values of a piecewise smooth function. Reduction of computational effort in this case can be obtained by using at each time-step a composite grid which consists of patches of cells from the appropriate level of resolution as determined from multiresolution analysis of the solution at time t_n. However, if the grid is rectangular and uniform (or can be obtained by a piecewise-smooth transformation from such a grid) then the numerical fluxes of *each* of the space directions can be thought of as pointvalues of a piecewise-smooth function, and thus can be successfully operated upon by a pointwise multiresolution algorithm for data compression.

This observation suggests the following multiresolution algorithm for the approximation of the numerical fluxes of the finest grid: We consider a sequence of nested diadic grids $\{G^k\}_{k=0}^{L}$ where $k = 0$ is the finest and $k = L$ is the coarsest. We start by computing the values of the finest grid fluxes at the few grid points of the coarsest grid G^L. As we proceed from G^k to the finer grid G^{k-1} we calculate finest-grid fluxes at the gridpoints of G^{k-1} as follows: Values at the common points $G^k \cap G^{k-1}$ are transferred from G^k; at each point of $(G^k)^c \cap G^{k-1}$ we either interpolate the value of the flux from G^k or, if this is not accurate enough, we compute it directly from the finest grid G^0. The local decision whether interpolation from the coarser grid G^k is accurate enough is thus the heart of this algorithm. As pointed out in [7] this decision can be made by examining the multiresolution analysis of v^n, the numerical solution at the beginning of the time-step, which is readily available to us. Roughly speaking, numerical fluxes of the finest grid G^0 can be interpolated from G^k if the same is true for the cell-averages of v^n at the same locality, i.e. if finest-grid cell-averages of v^n can be accurately computed from the averages of v^n over the larger cells of G^k.

In Section 5 of the present paper we follow an idea of Liandrat and Tchamitchian [12] in order to formulate a simple criterion for this decision. In Section 6 we present a sample of numerical experiments which indicate that this multiresolution algorithm is a viable alternative to the adaptive grid approach. In Section 7 we make some observations about the accumulation of error in the multiresolution algorithm and draw some conclusions.

2. Multiresolution representation of pointvalues

In this section we consider a situation where we are given $N_0 = 2^{n_0}$ values

$$u^0 = \{u_j^0\}_{j=1}^{N_0} \tag{2.1}$$

which we interprete as pointvalues of some function $u(x)$ corresponding to a uniform partition of $[0,1]$,

$$G^0 = \{x_j^0\}_{j=0}^{N_0}, \quad x_j^0 = j \cdot h_0, \quad h_0 = 1/N_0, \tag{2.2a}$$

$$u_j^0 = u(x_j^0), \qquad 1 \le j \le N_0. \tag{2.2b}$$

To simplify our presentation we assume that $u(x)$ is periodic with period 1, so that its values outside $(0,1]$ are known by periodic extension, i.e. $u_0^0 = u_{N_0}^0$, etc.

We consider the set of nested diadic grids $G^k,\ 0 \le k \le L$,

(2.3a) $$G^k = \{x_j^k\}_{j=0}^{N_k},\ x_j^k = j \cdot h_k,\ h_k = 2^k h_0,\ N_k = N_0/2^k,$$

where $k = 0$, the original grid (2.2a), is the finest in the hierarchy and $k = L,\ L < n_0$, is the coarsest. Note that G^k is formed from the finer G^{k-1} by removing the gridpoints with odd indeces, i.e.

(2.3b) $$G^{k-1} - G^k = \{x_{2j-1}^{k-1}\}_{j=1}^{N_k}$$

and that $G^{k-1} \cap G^k = G^k$,

(2.3c) $$x_j^k = x_{2j}^{k-1},\quad 0 \le j \le N_k.$$

Interpreting the given sequence u^0 (2.1) as pointvalues of a function $u(x)$ (2.2) we define for the k-th grid a sequence $u^k = \{u_j^k\}_{j=1}^{N_k}$ by

(2.4) $$u_j^k = u(x_j^k) = u(x_{2^k j}^0) = u_{2^k j}^0,\quad 0 \le j \le N_k;$$

thus u^k is formed from u^{k-1} by setting

(2.5a) $$u_j^k = u_{2j}^{k-1},\quad 1 \le j \le N_k,$$

and removing the values of u^{k-1} with odd indeces

(2.5b) $$u^{k-1} - u^k = \{u_{2j-1}^{k-1}\}_{j=1}^{N_k}.$$

Let $I(x; u^k)$ denote any interpolating function of the k-th grid, i.e.

(2.6a) $$I(x_j^k; u^k) = u_j^k, \quad 0 \le j \le N_k,$$

and use it in order to obtain approximate values $\tilde{u}_{2j-1}^{k-1}$, $1 \le j \le N_k$, to the "missing values" in (2.5b),

(2.6b) $$\tilde{u}_{2j-1}^{k-1} = I(x_{2j-1}^{k-1}; u^k).$$

Let us denote by $D^k(u^0) = \{D_j^k\}_{j=1}^{N_k}$ the sequence of interpolation errors

(2.7a) $$D_j^k = u_{2j-1}^{k-1} - \tilde{u}_{2j-1}^{k-1} = u_{2j-1}^{k-1} - I(x_{2j-1}^{k-1}; u^k), \quad 1 \le j \le N_k,$$

and observe that knowledge of (u^k, D^k) is equivalent to knowledge of u^{k-1}

(2.7b) $$u^{k-1} \leftrightarrow (D^k, u^k)$$

in the sense that there is a one-to-one transformation between the two sets (note that both sets have the same number of elements $N_{k-1} = 2N_k$):

$\underline{u^{k-1} \to (D^k, u^k)}$

(i) Define

(2.8a) $$u_j^k = u_{2j}^{k-1}, \quad 1 \le j \le N_k.$$

(ii) Compute

(2.8b) $$D_j^k = u_{2j-1}^{k-1} - I(x_{2j-1}^{k-1}; u^k), \quad 1 \le j \le N_k.$$

$\underline{(D^k, u^k) \to u^{k-1}}$

(i) Define

(2.9a)
$$u_{2j}^{k-1} = u_j^k, \quad 1 \le j \le N_k.$$

(ii) Compute

(2.9b)
$$u_{2j-1}^{k-1} = I(x_{2j-1}^{k-1}; u^k) + D_j^k.$$

Using (2.7b) for $1 \le k \le L$, we get

(2.10)
$$u^0 \leftrightarrow (D^1, u^1) \leftrightarrow (D^1, (D^2, u^2)) = (D^1, D^2, u^2) \leftrightarrow \cdots \leftrightarrow (D^1, D^2, \ldots, D^L, u^L) = (u_M)^T;$$

where u_M,

(2.11)
$$u_M = (D^1, D^2, \ldots, D^L, u^L)^T$$

is the multiresolution representation of u^0. Relation (2.10) shows that there is a one-to-one transformation between u^0 and u_M which we denote by

(2.12)
$$u_M = \mathbf{M}\ u^0, \quad u^0 = \mathbf{M}^{-1}\ u_M.$$

We note that the above relations hold for any interpolation technique in (2.6)-(2.7) and that for purposes of pure data compression it is advisable to use an adaptive (data-dependent) interpolation (see [8]), in which case $\mathbf{M}$ is a nonlinear operator. In the application of this paper (as in [10]) we elect to use the simplest central interpolation where the value at x^{k-1}_{2j-1} is computed from the $(r-1)$-th degree polynomial that interpolates $(u^k_{j-s}, \dots, u^k_{j+s-1})$, $r = 2s$. In this case

(2.13a) $$\tilde{u}^{k-1}_{2j-1} = I(x^{k-1}_{2j-1}; u^k) = \sum_{\ell=1}^{s} \beta_\ell (u^k_{j+\ell-1} + u^k_{j-\ell}), \quad r = 2s,$$

(2.13b) $$\begin{cases} r = 2 \Rightarrow & \beta_1 = 1/2 \\ r = 4 \Rightarrow & \beta_1 = 9/16, \ \beta_2 = -1/16 \\ r = 6 \Rightarrow & \beta_1 = 150/256, \quad \beta_2 = -25/256, \ \beta_3 = 3/256, \end{cases}$$

and $\mathbf{M}$ is a linear operator which can be represented by an $N_0 \times N_0$ matrix. Without writing the explicit form of this matrix, it follows from (2.8) and (2.9) that

$\underline{u_M = \mathbf{M} u^0}$ (Encoding)

can be computed by the algorithm

(2.14) $$\begin{cases} \text{DO for } k = 1, 2, \dots, L \\ u^k_j = u^{k-1}_{2j}, \qquad 1 \le j \le N_k \\ D^k_j = u^{k-1}_{2j-1} - \sum_{\ell=1}^{s} \beta_\ell (u^k_{j+\ell-1} + u^k_{j-\ell}), \quad i \le j \le N_k \end{cases}$$

and that

$\underline{u^0 = \mathbf{M}^{-1} u_M}$ (Decoding)

can be computed by the algorithm

(2.15) $$\begin{cases} \text{DO for } k = L, \ L-1, \dots, 1 \\ u^{k-1}_{2j} = u^k_j, \qquad 1 \le j \le N_k \\ u^{k-1}_{2j-1} = \sum_{\ell=1}^{s} \beta_\ell (u^k_{j+\ell-1} + u^k_{j-\ell}) + D^k_j, \quad 1 \le j \le N_k. \end{cases}$$

Note that the encoding algorithm (2.14) goes from fine to coarse while the decoding algorithm (2.15) goes from coarse to fine, and that both operations are inexpensive $O(N_0)$ algorithms

3. Multiresolution representation of cell-averages

In this section we consider a situation where we are given a sequence of N_0 values

(3.1a) $$\bar{u}^0 = \{\bar{u}_j^0\}_{j=1}^{N_0}$$

which we elect to interpret as cell-averages over the grid G^0 (2.2a) of some function $u(x)$

(3.1b) $$\bar{u}_j^0 = \frac{1}{h_0} \int_{x_{j-1}^0}^{x_j^0} u(x)dx, \quad 1 \le j \le N_0.$$

We consider now the set of nested grids G^k, $1 \le k \le L$ in (2.3) and define the sequence $\bar{u}^k = \{\bar{u}_j^k\}_{j=1}^{N_k}$ by

(3.2a) $$\bar{u}_j^k = \frac{1}{h_k} \int_{x_{j-1}^k}^{x_j^k} u(x)dx.$$

It follows immediately from this definition and (2.3c) that

(3.2b)
$$\bar{u}_j^k = \frac{1}{h_k} \int_{x_{j-1}^k}^{x_j^k} u(x)dx = \frac{1}{2h_{k-1}} \left[\int_{x_{2j-2}^k}^{x_{2j-1}^{k-1}} u(x)dx + \int_{x_{2j-1}^{k-1}}^{x_{2j}^{k-1}} u(x)dx \right] = \frac{1}{2}(\bar{u}_{2j-1}^{k-1} + \bar{u}_{2j}^{k-1}).$$

Therefore $\{\bar{u}_j^k\}_{j=1}^{N_k}$, $1 \le k \le L$, can be computed directly from the given data $\bar{u}^0$, without any explicit knowledge of the function $u(x)$, by the following algorithm

(3.3)
```
DO for $k = 1, 2, \dots, L$
DO for $j = 1, \dots, N_k$
$\bar{u}_j^k = \frac{1}{2}(\bar{u}_{2j-1}^{k-1} + \bar{u}_{2j}^{k-1}).$
```

We turn now to consider the pointvalues $U_j^k = U(x_j^k)$ of the primitive function of u

$$U(x) = \int_0^x u(y)dy \tag{3.4}$$

and observe that knowledge of the cell-averages $\bar{u}^k$ is equivalent to knowledge of the point-values U^k of the primitive function (3.4),

$$U^k = \{U_j^k\}_{j=1}^{N_k} \quad \leftrightarrow \quad \bar{u}^k = \{\bar{u}_j^k\}_{j=1}^{N_k}, \tag{3.5a}$$

as is evident from the following two relations:

$$U_j^k = U(x_j^k) = h_k \sum_{i=1}^{j} \bar{u}_i^k, \tag{3.5b}$$

$$\bar{u}_j^k = [U(x_j^k) - U(x_{j-1}^k)]/h_k = (U_j^k - U_{j-1}^k)/h_k. \tag{3.5c}$$

Hence knowing the values $\bar{u}^k$ we can compute U^k and use (2.6) to approximate the "missing values" U_{2j-1}^{k-1}, $1 \leq j \leq N_k$ by $\tilde{U}_{2j-1}^{k-1}$

$$\tilde{U}_{2j-1}^{k-1} = I(x_{2j-1}^{k-1}; U^k). \tag{3.6a}$$

Recalling that $U_j^k = U_{2j}^{k-1}$ (2.5a) we use (3.5c) to get an approximation $\tilde{u}^{k-1}$ for the cell-averages $\bar{u}^{k-1}$ of the finer grid by

$$\tilde{u}_{2j-1}^{k-1} = (\tilde{U}_{2j-1}^{k-1} - U_{j-1}^k)/h_{k-1}, \; \tilde{u}_{2j}^{k-1} = (U_j^k - \tilde{U}_{2j-1}^{k-1})/h_{k-1}, \; 1 \leq j \leq N_k. \tag{3.6b}$$

We note that

$$\frac{1}{2}(\tilde{u}_{2j-1}^{k-1} + \tilde{u}_{2j}^{k-1}) = \frac{1}{2h_{k-1}}(U_j^k - U_{j-1}^k) = \bar{u}_j^k, \tag{3.6c}$$

therefore $\tilde{u}_{2j}^{k-1}$ can be computed from knowledge of $\bar{u}_j^k$ and $\tilde{u}_{2j-1}^{k-1}$ by

(3.6d) $$\tilde{u}_{2j}^{k-1} = 2\bar{u}_j^k - \tilde{u}_{2j-1}^{k-1}.$$

Next we denote by $d^k(\bar{u}^0) = \{d_j^k\}_{j=1}^{N_k}$ the sequence of approximation errors in predicting $\{\bar{u}_{2j-1}^{k-1}\}_{j=1}^{N_k}$ from the cell-averages $\bar{u}^k$ of the coarser grid

(3.7a) $$d_j^k = \bar{u}_{2j-1}^{k-1} - \tilde{u}_{2j-1}^{k-1} = \bar{u}_{2j-1}^{k-1} - [I(x_{2j-1}^{k-1}; U^k) - U_{j-1}^k]/h_{k-1}, \ 1 \le j \le N_k,$$

and observe that knowledge of $(d^k, \bar{u}^k)$ is equivalent to knowledge of $\bar{u}^{k-1}$

(3.7b) $$\bar{u}^{k-1} \leftrightarrow (d^k, \bar{u}^k)$$

in the sense that there is a one-to-one transformation between the two sets:

$\underline{\bar{u}^{k-1} \rightarrow (d^k, \bar{u}^k)}$

(i) Define

(3.8a) $$\bar{u}_j^k = \frac{1}{2}(\bar{u}_{2j-1}^{k-1} + \bar{u}_{2j}^{k-1}), \ 1 \le j \le N_k$$

(ii) Compute

(3.8b) $$d_j^k = \bar{u}_{2j-1}^{k-1} - [I(x_{2j-1}^{k-1}; U^k) - U_{j-1}^k]/h_{k-1}, \ 1 \le j \le N_k.$$

$\underline{(d^k, \bar{u}^k) \rightarrow \bar{u}^{k-1}}$

Compute

(3.9) $$\begin{aligned} \bar{u}_{2j-1}^{k-1} &= [I(x_{2j-1}^{k-1}; U^k) - U_{j-1}^k]/h_{k-1} + d_j^k, \\ \bar{u}_{2j}^{k-1} &= 2\bar{u}_j^k - \bar{u}_{2j-1}^{k-1}, \qquad 1 \le j \le N_k. \end{aligned}$$

As in (2.10) we conclude that there is a one-to-one transformation between $\bar{u}^0$ and its multiresolution representation $\bar{u}_M$

$$\bar{u}_M = (d^1, d^2, \dots, d^L, \bar{u}^L)^T \tag{3.10}$$

which we denote by

$$\bar{u}_M = \bar{\mathbf{M}}\bar{u}^0, \quad \bar{u}^0 = \bar{\mathbf{M}}^{-1}\bar{u}_M. \tag{3.11}$$

When we use the central interpolation (2.13) in (3.8)-(3.9) we get that $\bar{\mathbf{M}}$ is a linear operator which can be expressed by an $N_0 \times N_0$ matrix, and that the transformations in (3.11) can be performed by the following algorithms:

$\underline{\bar{u}_M = \bar{\mathbf{M}}\bar{u}^0}$ (Encoding)

$$\left\{ \begin{array}{l} \text{DO for } \ k = 1, 2, \dots, L \\ \bar{u}^k_j = \frac{1}{2}(\bar{u}^{k-1}_{2j-1} + \bar{u}^{k-1}_{2j}), \qquad 1 \le j \le N_k, \\ d^k_j = \bar{u}^{k-1}_{2j-1} - \bar{u}^k_j - \sum_{\ell=1}^{s-1} \gamma_\ell(\bar{u}^k_{j+\ell} - \bar{u}^k_{j-\ell}), \quad 1 \le j \le N_k. \end{array} \right. \tag{3.12}$$

$\underline{\bar{u}^0 = \bar{\mathbf{M}}^{-1}\bar{u}_M}$ (Decoding)

$$\left\{ \begin{array}{l} \text{DO for } \ k = L, L-1, \dots, 1 \\ \text{DO for } \ j = 1, \dots, N_k \\ \Delta = \sum_{\ell=1}^{s-1} \gamma_\ell(\bar{u}^k_{j+\ell} - \bar{u}^k_{j-\ell}) + d^k_j \\ \bar{u}^{k-1}_{2j-1} = \bar{u}^k_j + \Delta, \quad \bar{u}^{k-1}_{2j} = \bar{u}^k_j - \Delta. \end{array} \right. \tag{3.13}$$

Here the order of accuracy is $\bar{r} = 2s - 1$, and the corresponding coefficients γ_ℓ are

$$\left\{ \begin{array}{l} \bar{r} = 3 \Longrightarrow \gamma_1 = -1/8 \\ \bar{r} = 5 \Longrightarrow \gamma_1 = -22/128, \ \gamma_2 = 3/128. \end{array} \right. \tag{3.14}$$

We remark that since $\bar{u}^0$ is equivalent to U^0 ((3.5a) with $k = 0$) it follows immediately that $\bar{u}_M$ is equivalent to U_M, the multiresolution representation of the pointvalues of the primitive function

$$\bar{u}_M \leftrightarrow U_M = (D^1, D^2, \dots, D^L, U^L)^T. \tag{3.15a}$$

The transformation between $\bar{u}^L$ and U^L is given by (3.5b)-(3.5c) with $k = L$. The transformation between $d_j^k(\bar{u}^0)$ (3.7a) and $D_j^k(U^0)$ (2.7a) is given by

$$d_j^k(\bar{u}^0) = D_j^k(U^0)/h_{k-1}. \tag{3.15b}$$

This relation follows immediately from (3.7a) by using (3.5c) for $\bar{u}_{2j-1}^{k-1}$ and recalling $U_{j-1}^k = U_{2j-2}^{k-1}$ (2.5a), i.e.

$$\begin{aligned} d_j^k(\bar{u}^0) &= (U_{2j-1}^{k-1} - U_{2j-2}^{k-1})/h_{k-1} - [I(x_{2j-1}^{k-1}; U^k) - U_{2j-2}^{k-1}]/h_{k-1} \\ &= [U_{2j-1}^{k-1} - I(x_{2j-1}^{k-1}; U^k)]/h_{k-1} = D_j^k(U^0)/h_{k-1}. \end{aligned}$$

4. Regularity analysis and data compression

In this section we use the multiresolution representation $\bar{u}_M$ to obtain a data compression algorithm of cell-averages and then study its application to the numerical solution v^n of the conservative scheme (1.2).

Using standard interpolation results and recalling that the primitive function $U(x)$ (3.4) is smoother than $u(x)$ we get from (3.15b) the following qualitative description of the behavior of $d_j^k(\bar{u}^0)$: If $u(x)$ at $x = \bar{x}$ has $p-1$ continuous derivatives and a jump-discontinuity in its p-th derivative, then for x_j^k near $\bar{x}$

$$d_j^k(\bar{u}^0) \sim \begin{cases} (h_k)^p[u^{(p)}] & \text{for } 0 \le p \le \bar{r} \\ (h_k)^{\bar{r}} u^{(\bar{r})} & \text{for } p > \bar{r} \end{cases}; \tag{4.1a}$$

here $\bar{r}$ is the order of accuracy of the approximation ($\bar{r} = r - 1$) and [ ] denotes the jump at the discontinuity. It follows therefore that

$$|d_{2j}^{k-1}| \approx 2^{-\bar{p}} |d_j^k|, \quad \bar{p} = \min(p, \bar{r}), \tag{4.1b}$$

provided that the k-th grid is fine enough for the asymptotics to hold. This shows that away from discontinuities of the function, the coefficients $d_j^k(\bar{u}^0)$ diminish in size as we go to a finer grid with a rate which is determined by the local regularity of the function and the order of accuracy of the approximation. In the neighborhood of a discontinuity of $u(x)$ the coefficients $d_j^k(\bar{u}^0)$ remain of the same size independent of the level of refinement. Conversely, the above relation between rate of decay and regularity also enables us to estimate the local regularity of the function by studying the local rate of decay of its coefficients $d_j^k(\bar{u}^0)$. In this respect the process of finding the multiresolution representation of $\bar{u}^0$ can be viewed as performing analysis of its local regularity.

We turn now to discuss data compression of $\bar{u}^0$. In [10] we showed stability of the decoding algorithm with respect to perturbations in the multiresolution representation. Let $\hat{u}_M$,

$$\hat{u}_M = \mathbf{tr}_\varepsilon(\bar{u}_M) = (\hat{d}^1, \hat{d}^2, \ldots, \hat{d}^L, \bar{u}^L)^T \tag{4.2a}$$

denote the result of the following truncation operation on $\bar{u}_M$:

$$\hat{d}_j^k = \begin{cases} d_j^k & \text{if } |d_j^k| > \varepsilon_k \\ 0 & \text{if } |d_j^k| \le \varepsilon_k \end{cases}, \quad 1 \le j \le N_k, \quad 1 \le k \le L, \tag{4.2b}$$

and let $\hat{u}^0$ denote the result of applying the decoding algorithm to the truncated data $\hat{u}_M$

$$\hat{u}^0 = \bar{\mathbf{M}}^{-1}\hat{u}_M. \tag{4.2c}$$

Using the results of [10] we get that

$$\|\bar{u}^0 - \hat{u}^0\| = \|\bar{\mathbf{M}}^{-1}[\bar{u}_M - \mathbf{tr}_\varepsilon(\bar{u}_M)]\| \tag{4.3}$$

can be bounded by

$$\|\bar{u} - \hat{u}^0\| \le C \sum_{k=1}^{L} \varepsilon_k \tag{4.4a}$$

where C is independent of the number of levels L, for both the L_1 and L_∞ norms (here we use $\|u^0\|_1 = h_0 \sum_{i=1}^{N_0} |u_i^0|$). Taking

$$\varepsilon_k = 2^{-k}\varepsilon/C \tag{4.4b}$$

we get from (4.4a) that

$$\|\bar{u}^0 - \hat{u}^0\| \le \varepsilon. \tag{4.4c}$$

Hence given ε, a prescribed tolerance for error, we can achieve data compression by truncating the multiresolution representation (4.2) with truncation levels ε_k determined by (4.4b).

Let $\mathcal{D}_\varepsilon$ denote the set of indices (j,k) of the significant coefficients d_j^k in the multiresolution representation of $\bar{u}^0$, i.e.

$$\mathcal{D}_\varepsilon = \mathcal{D}_\varepsilon(\bar{u}^0) = \{(j,k) \mid |d_j^k| > \varepsilon_k\}, \tag{4.5}$$

and let $|\mathcal{D}|$ denote the number of elements in $\mathcal{D}$. Taking into account the number of elements in the coarsest level $\bar{u}^L$, the rate of compression is thus

$$\text{(4.6)} \qquad \text{rate of compression } = \frac{N_0}{|\mathcal{D}_\varepsilon| + N_0/2^L}.$$

Next we apply data compression to the numerical solution (1.2) and examine its relation to the regularity of the solution. In the following experiments we take (1.2) to be the first order upwind scheme [6]

$$\text{(4.7a)} \qquad \bar{f}(u_1, u_2) = \frac{1}{2}[f(u_1) + f(u_2) - |\bar{a}(u_1, u_2)|(u_2 - u_1)]$$

where

$$\text{(4.7b)} \qquad \bar{a}(u_1, u_2) = \begin{cases} [f(u_2) - f(u_1)]/(u_2 - u_1) & u_1 \neq u_2 \\ f'(u_1) & u_1 = u_2 \end{cases};$$

this scheme is the scalar version of Roe's scheme [15].

In Figures 1a, b, c we show the numerical solution v^n and its data compression $\mathcal{D}_\varepsilon(v^n)$ for $n = 25, 75, 125$, respectively, for the initial value problem

$$\text{(4.8)} \qquad w_t + (w^2/2)_x = 0, \quad w(x,0) = \chi_{[-\frac{1}{2},\frac{1}{2}]}(x) = \begin{cases} 1 & |x| \leq \frac{1}{2} \\ 0 & 1 \geq |x| > \frac{1}{2} \end{cases}$$

with periodic boundary conditions at $x = \pm 1$; here $N_0 = 256$ and $CFL = 0.8$. Each figure consists of 2 parts: The diagram at the bottom displays $\mathcal{D}_\varepsilon(v^n)$ in the $x - k$ plane by drawing a circle around (x_{2j-1}^{k-1}, k) for each element (j, k) in the set.

The plot at the top displays v^n by asterisks and $\hat{v}^n$, the decoded values (4.2c) by circles. In Table 1 we show the compression rate and the errors $e_p = \|v^n - \hat{v}^n\|_p,\ p = 1, 2, \infty$, for the above calculations

$$(4.9a) \qquad e_\infty = \max_{1 \le i \le N_0} |v_i^n - \hat{v}_i^n|,$$

$$(4.9b) \qquad e_p = \left\{ \frac{1}{N_0} \sum_{i=1}^{N_0} |v_i^n - \hat{v}_i^n|^p \right\}^{\frac{1}{p}}, \quad p = 1, 2.$$

In Figures 2a, b, c and Table 2 we repeat the above for the numerical solution of

$$(4.10) \qquad w_t + (w^2/2)_x = 0, \quad w(x,0) = 2 + \sin \pi x, \quad -1 \le x \le 1,$$

for $n = 25, 150, 400$ respectively.

In both cases 1 and 2 we applied the data compression algorithm (4.2) to the numerical solution with $L = 5$ levels of resolution and $\varepsilon_k = 10^{-3}/2^k$, $1 \le k \le L$, and used (3.12)-(3.13) with $\bar{r} = 3$ $(s = 1)$.

The solution of (4.8) starts with two discontinuities at $x = \pm 0.5$; the one at $x = 0.5$ is a shock and that at $x = -0.5$ is a centered rarefaction wave with discontinuous derivatives at its endpoints (corners). We observe that $\mathcal{D}_\varepsilon(v^n)$ in Figures 1a, b, c has three spikes corresponding to these irregularities of the solution: The spike corresponding to the shock remains of the same shape throughout the evolution of the solution; the one corresponding to the right endpoint of the rarefaction wave is becoming lower due to the expansion of the wave and the numerical rounding of the corner; the spike at the stationary left endpoint $x = -0.5$ retains its shape, which indicates a shock-like behaviour. Indeed a closer examination of the numerical solution shows a small stationary "negative shock" at the foot of the rarefaction wave at $x = -0.5$; this is due to the fact that Roe's scheme (4.7) does not enforce the entropy condition at sonic points.

The solution of (4.10) starts with smooth periodic data with extrema at $x = \pm 0.5$. The segment which has the maximum on its left and the minimum on its right is being compressed into a shock; the segment which is initially between $-0.5 \leq x \leq 0.5$ expands to fill the space between the shocks, thus forming a decaying N-wave. In Figure 2a ($n = 25$) the solution is still smooth; note that $\mathcal{D}_\varepsilon(v^n)$ in this figure does not have points at the finest levels 1 and 2. Comparing Fig. 2b for $n = 150$, which is just before the formation of the shock, with Fig. 2a we see the addition of finer scales at the shock and the elimination of scale 3 at the rarefaction wave due to its expansion. In Figure 2c ($n = 400$) we see an N-wave structure and observe that $\mathcal{D}_\varepsilon(v^n)$ has the typical spike for the shock, while the rarefaction, which is almost linear by now, is described by the coarsest level 5.

In Tables 1 and 2 we see that the error in all norms is well below the prescribed tolerance. The L_1-norm measures the smallest error and indeed it is the most appropriate for discontinuous functions.

REMARK: Figures 1 and 2 demonstrate that the location of irregularities in the numerical solution is shown by spikes in the $\mathcal{D}_\varepsilon(v^n)$ diagram. This information about the local regularity of the solution, which is available to us at the beginning of each time-step, can be used to trigger special procedures at shocks and discontinuous derivatives. For example one can use the inexpensive central stencil for ENO schemes [11] except at the "spikes", and only there to use the more expensive procedure of an adaptive stencil.

5. Multiresolution scheme

In this section we apply data compression to the numerical solution (1.2) and show how to compute its evolution in the compressed multiresolution representation. Our goal is to calculate the numerical solution of the given fine grid

$$v_j^{n+1,0} = v_j^{n,0} - \lambda_0(\bar{f}_j^0 - \bar{f}_{j-1}^0) \equiv (\mathbf{E}_0 \cdot v^{n,0})_j, \quad 1 \le j \le N_0, \tag{5.1a}$$

$$\bar{f}_j^0 = \bar{f}(v_{j-K}^{n,0}, \dots, v_{j+K}^{n,0}), \tag{5.1b}$$

within a prescribed tolerance in an efficient manner; here $\lambda_0 = \tau/h_0$ and $\mathbf{E}_0$ is the numerical evolution operator of the given scheme.

We use the notation $v_j^{n,k}$ for the cell-averages of the numerical solution (5.1) in $I_j^k = [x_{j-1}^k, x_j^k]$, the cells of the k-th grid, and $\bar{f}_j^k$ for the *pointvalue* of the numerical flux (5.1b) at x_j^k,

$$v_j^{n,k} = \frac{1}{2}(v_{2j-1}^{n,k-1} + v_{2j}^{n,k-1}), \tag{5.2a}$$

$$\bar{f}_j^k = \bar{f}_{2^k j}^0. \tag{5.2b}$$

Note that the numerical flux is generally a nonlinear function and that the values in (5.1b)-(5.2b) are computed on the given fine-grid; savings in computation will be achieved by calculating these fine-grid fluxes at fewer points. Let v_M^m denote the multiresolution representation (3.10) of the numerical solution $v^{m,0}$ (5.1a),

$$v_M^m = \bar{\mathbf{M}} v^{m,0} = \{d^1(v^m), \dots, d^L(v^m),\ v^{m,L}\}^T, \quad m = n, n+1. \tag{5.3}$$

Writing (5.1a) in vector form and operating with $\bar{\mathbf{M}}$ on both sides we get $\mathbf{E}_M$, the multiresolution form of the scheme $\mathbf{E}_0$

$$v_M^{n+1} = v_M^n - \lambda_0 \bar{\mathbf{M}} \begin{pmatrix} \bar{f}_1^0 - \bar{f}_0^0 \\ \vdots \\ \bar{f}_{N_0}^0 - \bar{f}_{N_0-1}^0 \end{pmatrix} = \bar{\mathbf{M}}\ \mathbf{E}_0 \cdot (\bar{\mathbf{M}}^{-1} v_M^n) \equiv \mathbf{E}_M \cdot v_M^n. \tag{5.4}$$

Note that $\mathbf{E}_M$ is equivalent to $\mathbf{E}_0$ and that both are nonlinear operators.

In Appendix A we show that the multiresolution scheme (5.4) can be expressed by

$$\text{(5.5)}\qquad \begin{cases} d_j^k(v^{n+1}) = d_j^k(v^n) - \lambda_{k-1} D_j^k(\bar{f}^0), & 1 \le j \le N_k,\ 1 \le k \le L, \\ v_j^{n+1,L} = v_j^{n,L} - \lambda_L(\bar{f}_j^L - \bar{f}_{j-1}^L), & 1 \le j \le N_L; \end{cases}$$

here $D_j^k(\bar{f}^0)$ is the interpolation error (2.7a)

$$\text{(5.6)}\qquad D_j^k(\bar{f}^0) = \bar{f}_{2j-1}^{k-1} - I(x_{2j-1}^{k-1}; \bar{f}^k),$$

and $\lambda_{k-1} = \tau / h_{k-1}$.

We introduce data compression into the multiresolution scheme (5.4)-(5.5) by applying the truncation operator $\mathbf{tr}_\varepsilon$ (4.2) to the numerical solution at the beginning of each time-step

$$\text{(5.7)}\qquad v_M^{n+1} = \bar{\mathbf{M}}\ \mathbf{E}_0 \cdot [\bar{\mathbf{M}}^{-1} \mathbf{tr}_\varepsilon(v_M^n)] \equiv \mathbf{E}_M^\varepsilon \cdot v_M^n;$$

observe that for $\varepsilon = 0$ (no compression) $\mathbf{E}_M^0 = \mathbf{E}_M$. Given a tolerance ε and $\mathbf{E}_0$ which is a monotone scheme, we show now that by chosing appropriate ε_k in (4.2) we get for any v_M

$$\text{(5.8a)}\qquad \|\mathbf{E}_M^\varepsilon \cdot v_M - \mathbf{E}_M \cdot v_M\|_1 \le \varepsilon.$$

Thus we can control the deviation of the compressed solution from the fine-grid solution per a single time-step. To prove (5.8) we recall [9] that if $\mathbf{E}_0$ is monotone, then it is L_1-contractive, i.e. for any $\bar{u}^0$ and $\bar{v}^0$

$$\|\mathbf{E}_0 \cdot \bar{u}^0 - \mathbf{E}_0 \cdot \bar{v}^0\|_1 \le \|\bar{u}^0 - \bar{v}^0\|_1.$$

This implies that

$$\begin{aligned}\|\mathbf{E}_M^\varepsilon \cdot v_M - \mathbf{E}_M \cdot v_M\|_1 &\le \|\bar{\mathbf{M}}\|_1 \|\mathbf{E}_0 \cdot [\bar{\mathbf{M}}^{-1}\mathbf{tr}_\varepsilon(v_M)] - \mathbf{E}_0 \cdot (\bar{\mathbf{M}}^{-1} v_M)\|_1 \\ &\le \|\bar{\mathbf{M}}\|_1 \|\bar{\mathbf{M}}^{-1}[\mathbf{tr}_\varepsilon(v_M) - v_M]\|_1.\end{aligned}$$

Using in (4.4)

$$\varepsilon_k = \frac{1}{\bar{C}\,C}\varepsilon/2^k, \quad \|\bar{\mathbf{M}}\|_1 \le \bar{C} \tag{5.8b}$$

we get that the RHS of the above inequality is smaller than ε which proves (5.8a).

We turn now to consider computational aspects of the multiresolution scheme (5.7). Since v_M^{n+1} is about to be truncated at the beginning of the next time-step, only $d_j^k(v^{n+1})$ which are above tolerance, i.e. for $(j,k) \in \mathcal{D}_\varepsilon(v^{n+1})$ (4.5), need to be calculated. At the beginning of the n-th time-step we only know $\mathcal{D}_\varepsilon(v^n)$ and not $\mathcal{D}_\varepsilon(v^{n+1})$. In Section 6 we show how to use knowledge of $\mathcal{D}_\varepsilon(v^n)$ in order to obtain an estimate $\tilde{\mathcal{D}}^{n+1}$ to $\mathcal{D}_\varepsilon(v^{n+1})$ which satisfies

$$\tilde{\mathcal{D}}^{n+1} \supseteq \mathcal{D}_\varepsilon(v^n) \cup \mathcal{D}_\varepsilon(v^{n+1}). \tag{5.9a}$$

In this case, if (j,k) is in $(\tilde{\mathcal{D}}^{n+1})^c$, the complement of $\tilde{\mathcal{D}}^{n+1}$, then both

$$|d_j^k(v^n)| < \varepsilon_k, \quad |d_j^k(v^{n+1})| < \varepsilon_k, \tag{5.9b}$$

and therefore it follows from (5.5a) and (5.8b) that

$$\lambda_0 |D_j^k(\bar{f})| < \frac{1}{\bar{C}\,C}\,\varepsilon \quad \text{for} \quad (j,k) \in (\tilde{\mathcal{D}}^{n+1})^c. \tag{5.9c}$$

This shows that for $(j,k) \in (\tilde{\mathcal{D}}^{n+1})^c$, the numerical flux $\bar{f}_j^{k-1}$ can be interpolated from $\bar{f}^k$ within a prescribed tolerance.

Next we describe our algorithmic implementation of the multiresolution scheme (5.7). Given v_M^n (5.3)

$$v_M^n = \{d^1(v^n), \dots, d^L(v^n),\ v^{n,L}\}^T$$

we compute v_M^{n+1} by the following steps:

Multiresolution Algorithm

Step (i) Truncate

(5.10a) $\hat{v}_M^n = \mathbf{tr}_\varepsilon(v_M^n)$

and calculate $\tilde{\mathfrak{D}}^{n+1}$ (details are given in (6.1) of the next section).

Step (ii) Prepare fine-grid values for numerical flux calculations

(5.10b) $\hat{v}^n = \bar{\mathbf{M}}^{-1}\hat{v}_M^n.$

Step (iii) Coarsest grid calculations:

Calculate

(5.10c) $\bar{f}_j^L = \bar{f}(\hat{v}_{j_0-K}^n, \dots, \hat{v}_{j_0+K}^n), \quad j_0 = 2^L j,\ 1 \le j \le N_L$

and update

(5.10d) $v_j^{n+1,L} = v_j^{n,L} - \lambda_L(\bar{f}_j^L - \bar{f}_{j-1}^L), \quad 1 \le j \le N_L.$

Step (iv) Computation of $\{d_j^k(v^{n+1})\}_{(j,k)\in\tilde{\mathfrak{D}}^{n+1}}$:

```
            DO for $k = L, L-1, \dots, 1$
            DO for $j = 1, \dots, N_k$
(5.10e)     $\bar{f}_{2j}^{k-1} = \bar{f}_j^k$
(5.10f)     $\tilde{f}_{2j-1}^{k-1} = I(x_{2j-1}^{k-1}; \bar{f}^k) = \sum_{\ell=1}^{s} \beta_\ell(\bar{f}_{j+\ell-1}^k + \bar{f}_{j-\ell}^k)$.
                  IF $((j,k) \in \tilde{\mathfrak{D}}^{n+1})$
            THEN
(5.10g)          $\bar{f}_{2j-1}^{k-1} = \bar{f}(\hat{v}_{j_0-K}^n, \dots, \hat{v}_{j_0+K}^n)$,   $j_0 = (2j-1)2^{k-1}$
(5.10h)          $d_j^k(v^{n+1}) = d_j^k(v^n) - \lambda_{k-1}(\bar{f}_{2j-1}^{k-1} - \tilde{f}_{2j-1}^{k-1})$
            ELSE
(5.10i)          $\bar{f}_{2j-1}^{k-1} = \tilde{f}_{2j-1}^{k-1}$
(5.10j)          $d_j^k(v^{n+1}) = 0.$
```

Our basic assumption in designing this algorithm is that a call for a numerical flux calculation (5.10g) is considerably more expensive than interpolating the flux from a coarser grid (5.10f). Therefore we measure the efficiency of this algorithm by the ratio μ of N_0, the number of flux calculations in the fine-grid scheme (5.1), to the actual number of calls for a numerical flux calculation in (5.10c) and (5.10g), i.e.

(5.11a)
$$\mu = \frac{N_0}{|\tilde{\mathfrak{D}}^{n+1}| + N_0/2^L};$$

since from (5.9a)

(5.11b)
$$|\tilde{\mathfrak{D}}^{n+1}| \geq |\mathfrak{D}_\varepsilon(v^n) \cup \mathfrak{D}_\varepsilon(v^{n+1})|,$$

we get that the efficiency μ is dominated by the rate of data compression for the numerical solution (4.6).

We observe that (5.10c), (5.10e)-(5.10g) is basically a decoding of the fine-grid fluxes from its compressed multiresolution representation via the pointvalue algorithm (2.15), except that the coefficients $D_j^k(\bar{f}^0)$ are computed "on the fly" and only where needed. The decision whether $D_j^k(\bar{f}^0)$ is to be computed is made indirectly from the multiresolution analysis of v^n, which is readily available to us. Therefore the multiresolution algorithm (5.10) can be viewed as a modified version of the given scheme in which the numerical fluxes are computed in an hierarchical way from coarse to fine, and the modification is that direct evaluation of the numerical flux is replaced by interpolation wherever this can be done to a prescribed accuracy. We describe this version of the multiresolution scheme in Appendix B.

6. Numerical Experiments

In solving hyperbolic conservation laws there are two effects that have to be taken into account: Finite speed propagation and compressibility; by the latter we refer to convergence of characteristics which is responsible for creation of shock waves. In order to illustrate the role of these effects in estimating $\tilde{\mathcal{D}}^{n+1}$ we show the relation between $\mathcal{D}_\varepsilon(v^n)$ and $\mathcal{D}_\varepsilon(v^{n+1})$ in two typical situations. In Figures 3a, b we present results of a fine-grid calculation by Roe's scheme (4.7) with $N_0 = 256$ and $CFL = 0.8$. Each figure consists of two parts: In the upper part we plot values of v^n and v^{n+1} for a particular n; values of v^n are marked by circles while those of v^{n+1} are shown by asterisks. In the diagram at the lower part of the figure we display the corresponding $\mathcal{D}_\varepsilon(v^n)$ and $\mathcal{D}_\varepsilon(v^{n+1})$, the first by circles and the latter by asterisks; here we use $\bar{r} = 5, L = 5$ and $\varepsilon_k = 10^{-3}/2^k$ for $1 \leq k \leq L$. In Figure 3a we show the results at $n = 100$ for the initial value problem (4.8). Here we see instances of an asterisk to the right of the circles – this is due to propagation. In Figure 3b we show results at $n = 99$ of the initial value problem (4.10). Here we have an example of an asterisk above the circles, i.e. a creation of a finer scale – this is due to a compression wave which is on its way to turn into a shock.

Taking into account these effects of finite-speed propagation and compressibility we suggest the following algorithm which combines the calculation of $\tilde{\mathfrak{D}}^{n+1}$ with the truncation operation in (5.10a):

(i) Set

$$\hat{\imath}(j,k) = 0, \quad 1 \le j \le N_k, \quad 1 \le k \le L. \tag{6.1a}$$

(ii)

```
DO for  $k = 1, \dots, L$
DO for  $j = 1, \dots, N_k$
        IF  $(|d_j^k(v^n)| \le \varepsilon_k)$
THEN
        $d_j^k(v^n) = 0$
ELSE
        $\hat{\imath}(j-\ell,k) = 1, \quad -\bar{K} \le \ell \le \bar{K}$
                IF  $(|d_j^k(v^n)| \ge 2^{\bar{p}+1}\varepsilon_k$ and $k > 1)$
        THEN
                $\hat{\imath}(2j-1,k-1) = 1,$
                $\hat{\imath}(2j,k-1) = 1$
```

(iii) Define $\tilde{\mathfrak{D}}^{n+1}$ by

$$\tilde{\mathfrak{D}}^{n+1} = \{(j,k) \mid \hat{\imath}(j,k) = 1\}. \tag{6.1e}$$

The matrix $\hat{\imath}(j,k)$ of 0 and 1 serves as a flag; the check whether $(j,k) \in \tilde{\mathfrak{D}}^{n+1}$ in (5.10) is replaced by a check whether $\hat{\imath}(j,k) = 1$. The range of the parameter $\bar{K}$ in (6.1c) is

(6.2a) $$1 \leq \bar{K} \leq K$$

where K is the support of the numerical flux function (1.2b). The choice $\bar{K} = K$ corresponds to the maximal speed of propagation in the numerical scheme. However, since propagation of "real" information is limited by the CFL condition, we expect the choice $\bar{K} = 1$ to be sufficient in most cases. In the numerical experiments of this paper we use the Roe's scheme (4.7) where $K = 1$ and thus $\bar{K} = 1$ from (6.2a). The parameter $\bar{p}$ in (6.1d) corresponds to the regularity analysis (4.1b) and is derived from mesh refinement considerations, i.e. we estimate $|d^k_{2j-1}(v^n)|$, $|d^k_{2j}(v^n)|$ by $2^{-\bar{p}}|d^k_j(v^n)|$, and check the latter w.r. to $\varepsilon_{k-1} = 2\varepsilon_k$. Hence $\bar{p} = \min(p, \bar{r})$ is determined by the regularity of the solution at x^k_j. We observe that at shocks all scales are present initially (see Figures 1 and 3a); new finer scales are created where the solution is smooth during a process of shock formation (see Figures 2 and 3b). Therefore we consider the range of $\bar{p}$ in the algorithm (6.1) to be

(6.2b) $$1 \leq \bar{p} \leq \bar{r} - 1,$$

where the upper limit of $(\bar{r} - 1)$ is set in order to take into account the effective loss of smoothness in the final stages of the shock formation.

In Figures 4, 5 and 6 we present results of the multiresolution scheme (5.10), (6.1) with Roe's numerical flux (4.7) for the periodic initial value problem (4.10). Each figure consists of 3 snapshots at $n = 25, 150, 400$ which are denoted respectively by a, b, c. In these calculation we used $N_0 = 256$ and time increment τ which corresponds to a $CFL = 0.8$ for the finest grid, i.e.

(6.3) $$\frac{\tau}{h_0} \max_{-1 \leq x \leq 1} |f'(w_0)| = 0.8.$$

In the upper part of each snapshot we compare the solution of the multiresolution scheme (circles) with the solution of the uniform fine-grid scheme (5.1) (asterisks) which is computed independently. In the lower part of each snapshot we display $\mathcal{D}_\varepsilon(v^n)$ for the solution of the multiresolution scheme (circles) and its corresponding estimate by (6.1) $\tilde{\mathcal{D}}^n$ (asterisks); here we use $L = 5$ levels of resolution and $\varepsilon_k = 10^{-3}/2^k$, $1 \leq k \leq L$. In Tables[1] 4, 5 and 6 we show the corresponding factor of efficiency μ (5.11a) and the L_p difference (4.9) between the numerical solution of the multiresolution scheme (5.10), (6.1) and the fine-grid solution (5.1).

In Figures 4 we show the results of the multiresolution scheme (5.10) with $\bar{r} = 3$ where $\tilde{\mathcal{D}}^{n+1}$ is calculated in the same way as in the wavelet algorithm of [2], i.e. if $(j,k) \in \mathcal{D}_\varepsilon(v^n)$ then we include the points $(2j-1, k-1)$ and $(2j, k-1)$ in $\tilde{\mathcal{D}}^{n+1}$; this is done by using the algorithm (6.1) with $\bar{p} = -1$ in (6.1d).

In Figure 5 we repeat the calculation of Figure 4 except that now we take $\bar{p} = \bar{r} - 1 = 2$ in (6.1d). Comparing the corresponding Tables 4 and 5 we see that the error in all norms is about the same, but the efficiency factor for $\bar{p} = \bar{r} - 1$ is considerably larger than that of $\bar{p} = -1$. Comparing $\tilde{\mathcal{D}}^n$ in Figures 4 and 5 with $\mathcal{D}_\varepsilon(v^n)$ for the fine-grid solution in Figure 2, we see that $\tilde{\mathcal{D}}^n$ in both cases is an overestimate, and that the one for $\bar{p} = \bar{r} - 1$ is more accurate than that for $\bar{p} = -1$. Comparing the efficiency factors of Tables 4 and 5 with the compression ratio of Table 2 we can quantify the extent of this overestimate.

In Figure 6 we repeat the calculation of Figure 5, except that now we take $\bar{r} = 5$; as before we take $\bar{p} = \bar{r} - 1 = 4$ in (6.1d). Comparing Table 6 with Table 5 we see that the errors are about the same; the factors of efficiency for $\bar{r} = 5$ is significantly larger than those of $\bar{r} = 3$ whenever the numerical solution has large segments of smooth variation; on

[1]In this paper we use the convention that the tables are numbered as the figures they correspond to; there is no Table 3

the other hand the spike at shocks for $\bar{r} = 5$ is wider by 2 points than that of $\bar{r} = 3$ due to the larger support of the stencil in (3.13) and (5.10f).

Next we show the results of the multiresolution scheme (5.10), (6.1) for a Riemann initial value problem (shock tube) for the Euler equations of gasdynamics

$$w_t + f(w)_x = 0, \;\; w(x,0) = \begin{cases} w_L & x < 0 \\ w_R & x > 0 \end{cases}, \tag{6.4a}$$

$$w = (\rho, m, E)^T, \tag{6.4b}$$

$$f(w) = uw + (0, P, Pu)^T; \tag{6.4c}$$

here ρ, m, E are the density, momentum, energy, respectively; $u = m/\rho$ is the velocity and P is the pressure which is related to w by an equation of state of polytropic gas with $\gamma = 1.4$

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho u^2\right); \tag{6.4d}$$

the states w_L and w_R in our numerical experiment are those of Sod's shock-tube problem

$$w_L = (1, 0, 2.5)^T, \quad w_R = (0.125, 0, 0.25)^T. \tag{6.4e}$$

The first order upwind scheme of Roe for the Euler equations of gasdynamics can be expressed by its numerical flux (4.7a) where $\bar{a}(u_1, u_2)$ is Roe's average Jacobian (see [15] for more details).

In Figure 7 and Table 7 we present results of the multiresolution scheme (5.10), (6.1) with Roe's numerical flux. In these calculations we solved the initial value problem (6.4) in the interval $[-1,1]$ with $N_0 = 512$ and a variable time-increment which corresponds to a $CFL = 0.8$ for the finest grid at each time-step. We stopped the calculation at $n = 250$ in order to avoid the question of boundary conditions. In Figure 7a we display $\mathcal{D}_\varepsilon(v^n)$ for the solution of the multiresolution scheme at $n = 250$ by circles and its corresponding estimate $\tilde{\mathcal{D}}^n$ (6.1) by asterisks; here we used $L = 6$ levels of resolution and $\varepsilon_k = 10^{-3}/2^k$, $1 \le k \le L$. The algorithm (6.1) was used with the following definition of $|d_j^k(v^n)|$,

$$|d_j^k(v^n)| = \max(|d_j^k(\rho^n)|,\ |d_j^k(m^n)|,\ |d_j^k(E^n)|) \tag{6.5}$$

where the quantities on the RHS are absolute values of the scalar quantities in (3.12) applied to the density, momentum and energy with $\bar{r} = 3$. In (6.1d) we used $\bar{p} = \bar{r} - 1 = 2$. In Figures 7b, c, d we compare the solution of the multiresolution scheme (circles) with the independently computed solution of the uniform fine-grid (5.1) (asterisks) for the density, velocity and pressure, respectively. In Table 7 we show the factor of efficiency (5.11a) and the difference $e_p, p = 1, \infty$ between the multiresolution solution and the fine-grid solution at several times during the calculation. These differences are calculated by

$$e_\infty = \max_{1\le i\le N_0} \max[e_i(\rho^n),\ e_i(m^n),\ e_i(E^n)], \tag{6.6a}$$

$$e_1 = \frac{1}{N_0}\sum_{i=1}^{N_0}[e_i(\rho^n) + e_i(m^n) + e_i(E^n)]/3, \tag{6.6b}$$

where $e_i(\cdot)$ denotes absolute value of the difference at x_i^0 between the two solutions. It is interesting to note that the errors in Table 7 for the Euler equations of gasdynamics are of the same order as those for the scalar case in Tables 4-6. Comparing $\tilde{\mathcal{D}}^n$ to $\mathcal{D}_\varepsilon(v^n)$ in Figure 7a we see that the algorithm (6.1) allowed for refinement on level 2 at the rarefaction wave and at the contact discontinuity; this is clearly unnecessary since both become smoother in time as a result of numerical dissipation and expansion of the rarefaction wave. To demonstrate this point we repeat the above calculation with $\bar{p} = 4$ in (6.1d), which effectively eliminates most of this superfulous refinement, and show the corresponding results in Table 8. Comparing to Table 7 we see an increased factor of efficiency with no significant change in the errors.

7. Accumulation of error and concluding remarks

Let v_ε^n denote the numerical solution of the compressed multiresolution scheme (5.7)

$$v_\varepsilon^n = \bar{\mathbf{M}}^{-1} v_M^n, \quad v_M^n = \bar{\mathbf{M}} v_\varepsilon^n, \tag{7.1a}$$

and w^n the cell-averages of the exact solution $w(x,t)$ at time t_n. Using (5.7) we express the evolution of the error e^n

$$e^n = v_\varepsilon^n - w^n \tag{7.1b}$$

by

$$\begin{aligned} e^{n+1} = v_\varepsilon^{n+1} - w^{n+1} &= [\mathbf{E}_0 \cdot \bar{\mathbf{M}}^{-1}\mathbf{tr}_\varepsilon(\bar{\mathbf{M}} v_\varepsilon^n) - \mathbf{E}_0 \cdot v_\varepsilon^n] + [\mathbf{E}_0 v_\varepsilon^n - \mathbf{E}_0 w^n] \\ &+ [\mathbf{E}_0 w^n - w^{n+1}] \end{aligned} \tag{7.2}$$

and study its accumulation in the time-strip $0 \le t \le T$ as $h_0 \to 0$ and $\lambda_0 = \tau / h_0 =$ fixed; in this process we keep a fixed coarsest grid and let $L \to \infty$ as $h_0 \to 0$.

First let us consider the linear case and assume that $\mathbf{E}_0$ is a linear oeprator. In this case we get from (7.2) and (4.4) that

(7.3a)

$$\begin{aligned} \|e^{n+1}\| &\le \|\mathbf{E}_0\|\|e^n\| + \|\mathbf{E}_0\|\|\bar{\mathbf{M}}^{-1}\mathbf{tr}_\varepsilon(\bar{\mathbf{M}} v_\varepsilon^n) - v_\varepsilon^n\| + \|w^{n+1} - \mathbf{E}_0 w^n\| \\ &\le \|\mathbf{E}_0\|\|e^n\| + [\|\mathbf{E}_0\|\varepsilon + \delta] \end{aligned}$$

where δ is a bound on the local truncation error

$$\delta = \max_n \|w^{n+1} - \mathbf{E}_0 w^n\|. \tag{7.3b}$$

If we assume that

$$\|\mathbf{E}_0\| \leq 1 + \tilde{C}\ h_0 \tag{7.3c}$$

then we get from (7.3a) the standard estimate

$$\|e^N\| \leq e^{\tilde{C}T/\lambda_0}\|e^0\| + (e^{\tilde{C}T/\lambda_0} - 1)[\delta + (1 + \tilde{C}\ h_0)\varepsilon]/(\tilde{C}\ h_0) \tag{7.3d}$$

where $0 \leq N\tau \leq T$.

Next we consider the nonlinear case and assume that $\mathbf{E}_0$ is a monotone scheme. In this case we get from (7.2), (4.4) and the L_1-contraction of $\mathbf{E}_0$ that

$$\|e^{n+1}\|_1 \leq \|e^n\|_1 + \varepsilon + \delta \leq \|e^0\|_1 + (n+1)(\varepsilon + \delta) \leq \|e^0\|_1 + \frac{T}{\lambda_0} \cdot \frac{\varepsilon + \delta}{h_0}. \tag{7.4}$$

In both cases (7.3) and (7.4) it is clear that chosing

$$\varepsilon = 0(\delta) \tag{7.5}$$

preserves the order of accuracy of $\mathbf{E}_0$ and the quality of the numerical results. We observe that the data-compression algorithm (4.2) in this refinement process $h_0 \to 0$ is used with a number of levels L which tends to infinity. Therefore stability of the data compression algorithm in this context means that for any v

$$\|\bar{\mathbf{M}}^{-1}\mathbf{tr}_\varepsilon(\bar{\mathbf{M}}v) - v\| \leq C \cdot \varepsilon \tag{7.6}$$

where the constant C is independent of L.

We remark that the numerical solution of the multiresolution scheme can be stored in its compressed form and thus reduce the storage requirements for the problem. If memory requirements pose a limitation on a satisfactory solution of the problem, we can replace the global decoding (5.10b) by a more compact one which takes into consideration the actual fine-grid values that are needed for the exact numerical flux evaluations in $\tilde{\mathfrak{D}}^{n+1}$.

The efficiency of the compressed multiresolution scheme (5.4) is very much problem dependent and is determined primarily by the rate of data compression that can be achieved for the solution of the given initial value problem. The algorithm in Appendix B is essentially an hierarchical rearrangement of the fine-grid scheme where at the extra cost of calculating $\tilde{\mathfrak{D}}^{n+1}$ (B.2a) and a check in (B.2b) we save the cost of numerical flux calculations wherever they can be accurately replaced by their interpolated value. We remark that the cost of calculating $\tilde{\mathfrak{D}}^{n+1}$ can be reduced by keeping track of Γ^n, the boundary of $\mathfrak{D}_\varepsilon(v^n)$ and restricting the operation of algorithm (6.1) to elements of the boundary Γ^n. A sharper estimate for $\tilde{\mathfrak{D}}^{n+1}$ can be obtained by limiting the refinement in (6.1d) to regions where the solution undergoes compression.

The above analysis shows that the compressed multiresolution scheme (5.4), or (B.2), is generally more efficient than the original fine-grid calculation. Since it is simple to program, it can be implemented in existing computer codes with relatively little effort.

Appendix A

In this appendix we show that the multiresolution scheme (5.4) can be expressed by (5.5)

$$\text{(A.1)}\qquad \begin{cases} d_j^k(v^{n+1}) = d_j^k(v^n) - \lambda_{k-1} D_j^k(\bar{f}^0), & 1 \le j \le N_k,\ 1 \le k \le L, \\ v_j^{n+1,L} = v_j^{n,L} - \lambda_L(\bar{f}_j^L - \bar{f}_{j-1}^L), & 1 \le j \le N_L. \end{cases}$$

To do so we express the fine-grid solution (5.1) in terms of the pointvalues $V_j^{m,0}$ of its primitive (3.5)

$$\text{(A.2)}\qquad V_j^{m,0} = h_0 \sum_{i=1}^{j} v_i^{m,0} \quad 1 \le j \le N_0,\ V_0^{m,0} = 0.$$

Summing (5.1) as above we get the pointvalue relation

$$\text{(A.3a)}\qquad V_j^{n+1,0} = V_j^{n,0} - \tau(\bar{f}_j^0 - \bar{f}_0^0),\ \ 1 \le j \le N_0.$$

Recalling that $V_j^{m,k} = V_{j\cdot 2^k}^{m,0}$, $m = n, n+1$, we get for the k-th grid that

$$\text{(A.3b)}\qquad V_j^{n+1,k} = V_j^{n,k} - \tau(\bar{f}_j^k - \bar{f}_0^0),\ \ 1 \le j \le N_k,$$

where $\bar{f}_j^k = \bar{f}_{j\cdot 2^k}^0$. Using the linearity of the interpolation $I(x;\cdot)$ in (2.13) as a functional of the data, we get from (A.3) that

$$\text{(A.4)}\qquad I(x; V^{n+1,k}) = I(x; V^{n,k}) - \tau[I(x;\bar{f}^k) - \bar{f}_0^0].$$

Subtracting (A.4) at x_{2j-1}^{k-1} from (A.3b) at the same grid-point we get the following relation between the various interpolation errors:

$$\text{(A.5)}\qquad D_j^k(V^{n+1}) = D_j^k(V^n) - \tau\ D_j^k(\bar{f}^0).$$

Dividing the above equality by h_{k-1} and using (3.15b) and $\lambda_{k-1} = \tau/h_{k-1}$ we get (A.1a). Relation (A.1b) follows immediately from (3.5c) and (A.3b) for $k = L$, i.e.

(A.6)
$$\begin{aligned} v_j^{n+1,L} &= \frac{1}{h_L}(V_j^{n+1,L} - V_{j-1}^{n+1,L}) = \frac{1}{h_L}(V_j^{n,L} - V_{j-1}^{n,L}) - \frac{\tau}{h_L}(\bar{f}_j^L - \bar{f}_{j-1}^L) \\ &= v_j^{n,L} - \lambda_L(\bar{f}_j^L - \bar{f}_{j-1}^L). \end{aligned}$$

Appendix B

In this appendix we present the multiresolution scheme (5.10) as an operation on the given finest grid, in which the numerical fluxes of the given scheme are computed efficiently within a prescribed tolerance by a combination of direct evaluation and interpolation. Given $v^{n,0}$ we calculate $v^{n+1,0}$ by the following algorithm

Step (i) Apply data compression and define $\tilde{\mathfrak{D}}^{n+1}$:

Compute

$$\hat{v}^{n,0} = \bar{\mathbf{M}}^{-1}\mathbf{tr}_\varepsilon(\bar{\mathbf{M}}v^{n,0}),$$

use algorithm (6.1) to define

$$\tilde{\mathfrak{D}}^{n+1} = \{(j,k) \mid \hat{i}(j,k) = 1\}. \tag{B.1}$$

Step (ii) Compute approximate $\bar{f}^0$:

Evaluate

$$\bar{f}_j^L = \bar{f}(\hat{v}_{i-K}^{n,0}, \dots, \hat{v}_{i+K}^{n,0}), \quad i = j \cdot 2^L, \ 1 \le j \le N_L \tag{B.2a}$$

(B.2b)

```
DO for $k = L, L-1, \dots, 1$
DO for $j = 1, \dots, N_k$
$\bar{f}_{2j}^{k-1} = \bar{f}_j^k$
    IF $(\hat{i}(j,k) = 1)$
THEN
    $\bar{f}_{2j-1}^{k-1} = \bar{f}(\hat{v}_{i-K}^{n,0}, \dots, \hat{v}_{i+K}^{n,0}), \quad i = (2j-1) \cdot 2^{k-1}$
ELSE
    $\bar{f}_{2j-1}^{k-1} = I(x_{2j-1}^{k-1}; \bar{f}^k) = \sum_{\ell=1}^{s} \beta_\ell(\bar{f}_{j+\ell-1}^k + \bar{f}_{j-\ell}^k)$
```

Step (iii) Update $v^{n+1,0}$:

$$v_j^{n+1,0} = \hat{v}_j^{n,0} - \lambda_0(\bar{f}_j^0 - \bar{f}_{j-1}^0), \quad 1 \le j \le N_0. \tag{B.3}$$

Observe that if we skip part (i) of estimating $\tilde{\mathfrak{D}}^{n+1}$ and instead set

$$\hat{i}(j,k) = 1, \quad i \le j \le N_k, \ 1 \le k \le L,$$

then the resulting algorithm is just a rearrangement of the given scheme in which the same numerical fluxes

$$\bar{f}_j^0 = \bar{f}(v_{j-K}^{n,0}, \dots, v_{j+K}^{n,0}), \quad 1 \le j \le N_0,$$

are computed in a different order, except for the extra (now redundant) IF statement. Therefore we can view the algorithm (B.1)-(B.3) as a modified version of the given scheme in which the numerical fluxes are computed in an hierarchical way from coarse to fine, and the modification is that direct evaluation of the numerical flux is replaced by interpolation wherever this can be done to a prescribed accuracy. The primary role of multiresolution in this algorithm is to supply the analysis whether the solution is indeed well resolved at a certain level at a particular locality, so that the numerical flux can be interpolated from the coarser grid to the specified accuracy.

Algorithm (B.1)-(B.3) is conceptually simpler than (6.1); in order to generalize it to other situations one has only to generalize the estimate in $\tilde{\mathfrak{D}}^{n+1}$.

References

[1] F. Arandiga, V. F. Candela and R. Donat, "Fast Multiresolution Algorithms for Solving Linear Equations: A Comparative Study", UCLA CAM Report 92-52, December 1992.

[2] E. Bacry, S. Mallat and G. Papanicolau, "A Wavelet Based Space-Time Adaptive Numerical Method for Partial Differential Equations", Mathematical Modeling and Numerical Analysis, Vol 26, pp. 793-834, 1992.

[3] G. Beylkin, R. Coifman and V. Rokhlin, "Fast Wavelet Transform and Numerical Algorithms. I", Comm. Pure Appl. Math., Vol. 44, pp. 141-183, 1991.

[4] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets", Comm. Pure Appl. Math., Vol. 41, pp. 909-996, 1988.

[5] B. Engquist, S. Osher and S. Zhang, "Fast Wavelet Algorithms for Linear Evolution Equations", ICASE Report 92-14, 1992.

[6] A. Harten, "The Artificial Compression Method for Computation of Shocks and Contact Discontinuities: I. Single Conservation Laws", Comm. Pure Appl. Math., Vol. 30, pp. 611-638, 1977.

[7] A. Harten, "Multiresolution Analysis for ENO Schemes", ICASE Report 91-77, September 1991.

[8] A. Harten, "Discrete Multiresolution Analysis and Generalized Wavelets", Journal of Applied Numerical Mathematics, Vol. 12, pp. 153-193, 1993; also UCLA CAM Report 92-08, February 1992.

[9] A. Harten, J. M. Hyman and P. D. Lax, "On Finite-Difference Approximations and Entropy Condition for Shocks", Comm. Pure Appl. Math., Vol. 29, pp. 297-322, 1976.

[10] A. Harten and I. Yad-Shalom, "Fast Multiresolution Algorithms for Matrix-Vector Multiplication", ICASE Report 92-55, October 1992.

[11] A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy, "Uniform High-Order Accurate ENO Schemes, III", J. Comput. Phys., Vol. 71, pp. 231-303, 1987.

[12] J. Liandrat and Ph. Tchamitchian, "Resolution of the 1D Regularized Burgers Equation Using a Spatial Wavelet Approximation", ICASE Report 90-83, December 1990.

[13] Y. Madday and J. C. Ravel, "Adaptive par Ondelettes: Conditions aux limites et dimensions superieures", Tech. Report, Universite Pierre et Marie Curie, Lab. d'Analyse Numerique, January 1992.

[14] S. Mallat, "Multiresolution Approximation and Wavelet Orthonormal Bases of L_2", Trans. Amer. Math. Soc., Vol. 315, pp. 69-87, 1989.

[15] P. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", J. Comput. Phys., Vol. 43, pp. 357-372, 1981.

Table 1. Data compression of numerical solution (4.8)

$N_0 = 256,\ \bar{r} = 3,\ L = 5;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \le k \le L.$

n	Compression	e_∞	e_1	e_2
25	5.56	4.65×10^{-4}	1.40×10^{-5}	6.55×10^{-5}
75	5.22	2.38×10^{-4}	1.54×10^{-5}	4.05×10^{-5}
125	5.45	2.31×10^{-4}	2.08×10^{-5}	4.37×10^{-5}

Table 2. Data compression of numerical solution (4.10)

$N_0 = 256,\ \bar{r} = 3,\ L = 5;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \le k \le L.$

n	Compression	e_∞	e_1	e_2
25	4.27	1.52×10^{-4}	3.78×10^{-5}	4.66×10^{-5}
150	4.00	4.66×10^{-4}	5.80×10^{-5}	9.07×10^{-5}
400	7.76	3.39×10^{-4}	3.00×10^{-5}	5.22×10^{-5}

Table 4. Compressed multiresolution scheme

$\bar{r} = 3,\ \bar{p} = -1,\ N_0 = 256,\ L = 5;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \leq k \leq L.$

n	Efficiency μ	e_∞	e_1	e_2
25	2.44	3.68×10^{-4}	7.22×10^{-5}	1.08×10^{-4}
150	2.41	3.94×10^{-3}	3.05×10^{-4}	6.89×10^{-4}
400	4.74	2.33×10^{-3}	8.90×10^{-5}	2.18×10^{-4}

Table 5. Compressed multiresolution scheme

$\bar{r} = 3,\ \bar{p} = \bar{r} - 1 = 2,\ N_0 = 256,\ L = 5;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \leq k \leq L.$

n	Efficiency μ	e_∞	e_1	e_2
25	4.06	3.57×10^{-4}	8.06×10^{-5}	1.11×10^{-4}
150	3.56	4.28×10^{-3}	3.09×10^{-4}	7.43×10^{-4}
400	5.56	2.92×10^{-3}	9.62×10^{-5}	2.60×10^{-4}

Table 6. Compressed multiresolution scheme

$\bar{r} = 5,\ \bar{p} = \bar{r} - 1 = 4,\ N_0 = 256,\ L = 5;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \leq k \leq L.$

n	Efficiency μ	e_∞	e_1	e_2
25	9.85	3.40×10^{-4}	7.71×10^{-5}	1.17×10^{-4}
150	4.57	3.26×10^{-3}	2.82×10^{-4}	4.93×10^{-4}
400	4.65	2.74×10^{-3}	1.15×10^{-4}	2.53×10^{-4}

Table 7. Compressed multiresolution scheme for gasdynamics

$\bar{r} = 3,\ \bar{p} = 2,\ N_0 = 512,\ L = 6;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \leq k \leq L.$

n	Efficiency μ	e_∞	e_1
50	5.75	1.44×10^{-3}	3.52×10^{-5}
100	4.92	2.23×10^{-3}	9.26×10^{-5}
150	4.61	3.25×10^{-3}	1.67×10^{-4}
200	4.53	3.25×10^{-3}	2.09×10^{-4}
250	4.65	3.48×10^{-3}	2.54×10^{-4}

Table 8. Compressed multiresolution scheme for gasdynamics

$\bar{r} = 3,\ \bar{p} = 4,\ N_0 = 512,\ L = 6;\ \varepsilon_k = 10^{-3}/2^k,\ 1 \leq k \leq L.$

n	Efficiency μ	e_∞	e_1
50	6.40	1.90×10^{-3}	4.18×10^{-5}
100	5.28	2.54×10^{-3}	1.02×10^{-4}
150	5.50	3.71×10^{-3}	1.92×10^{-4}
200	5.22	3.71×10^{-3}	2.42×10^{-4}
250	5.22	3.96×10^{-3}	2.95×10^{-4}

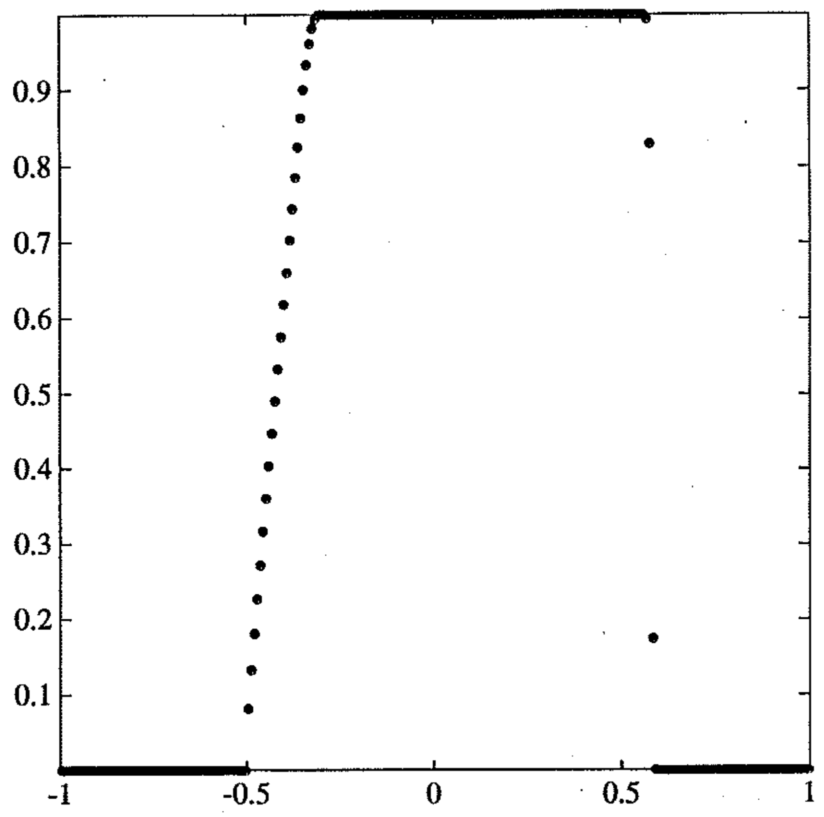
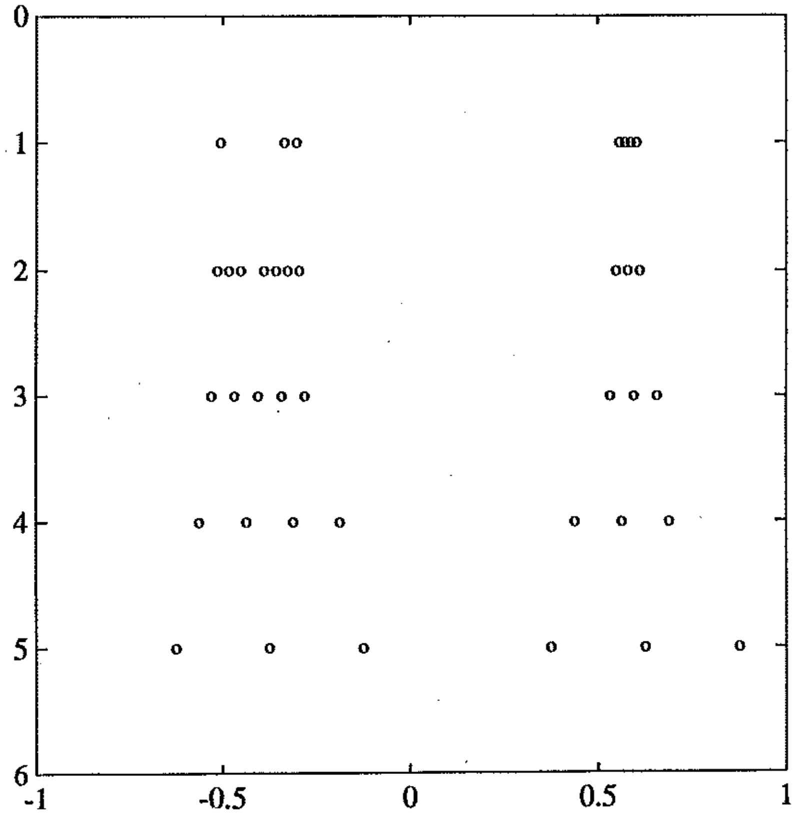


Figure 1a. Data compression of (4.8), $n = 25$.



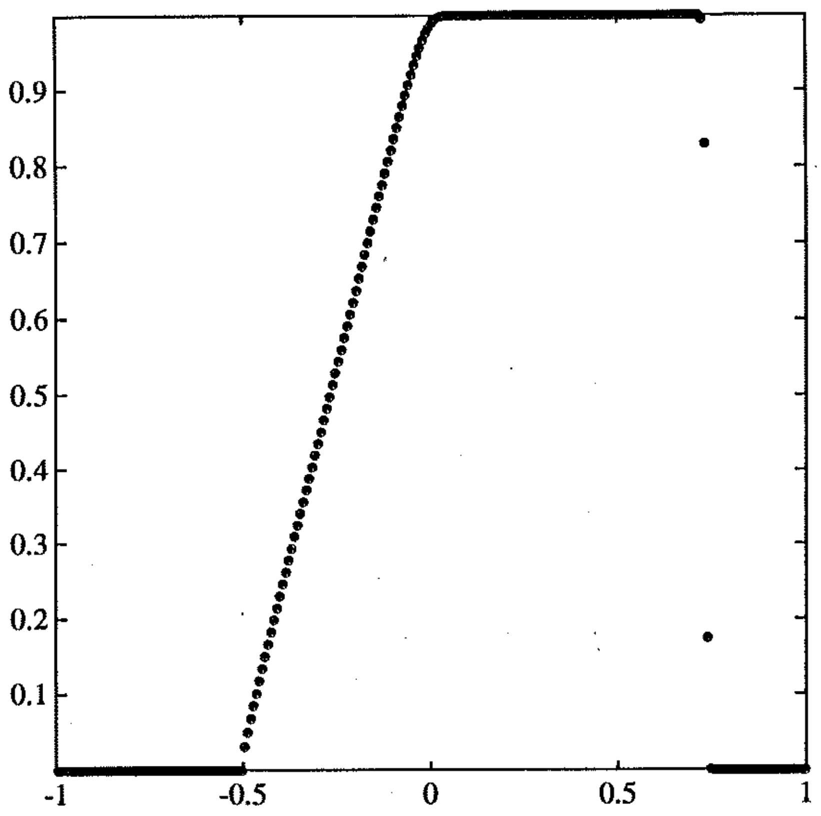
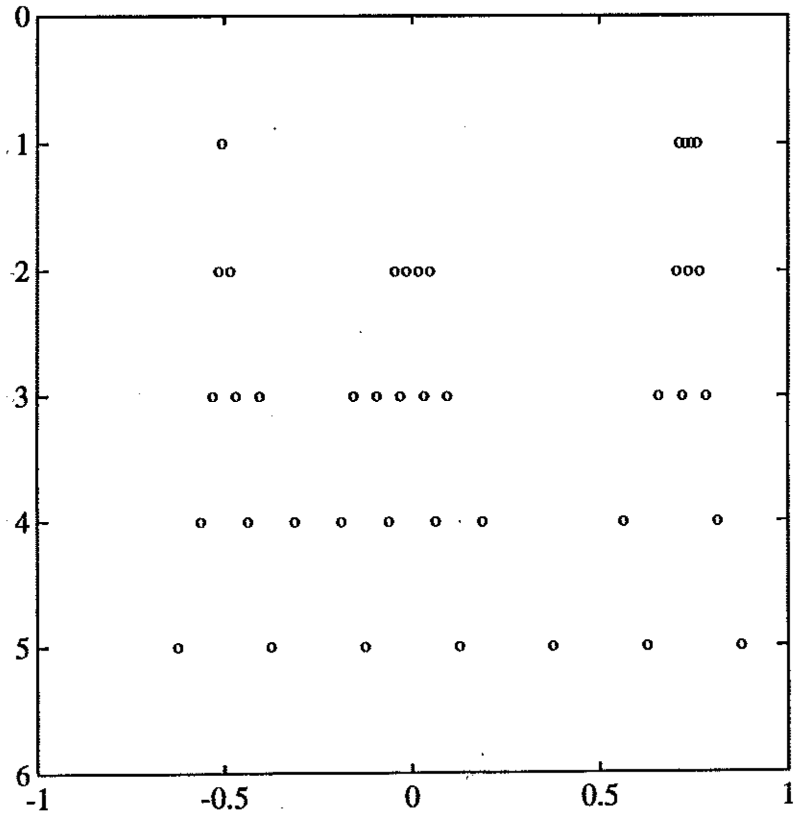


Figure 1b. Data compression of (4.8), $n = 75$.



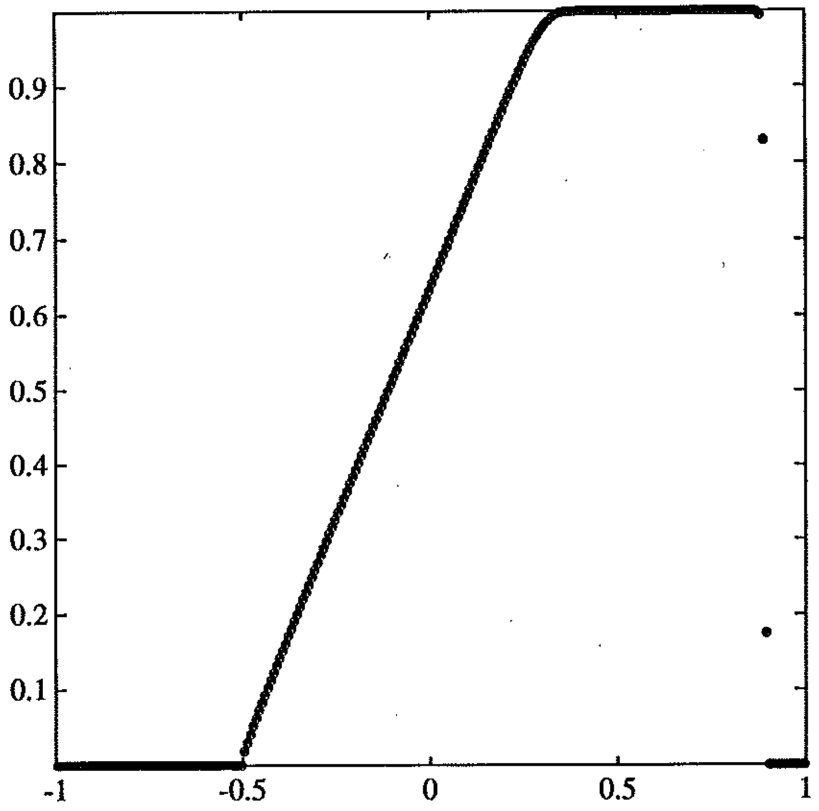
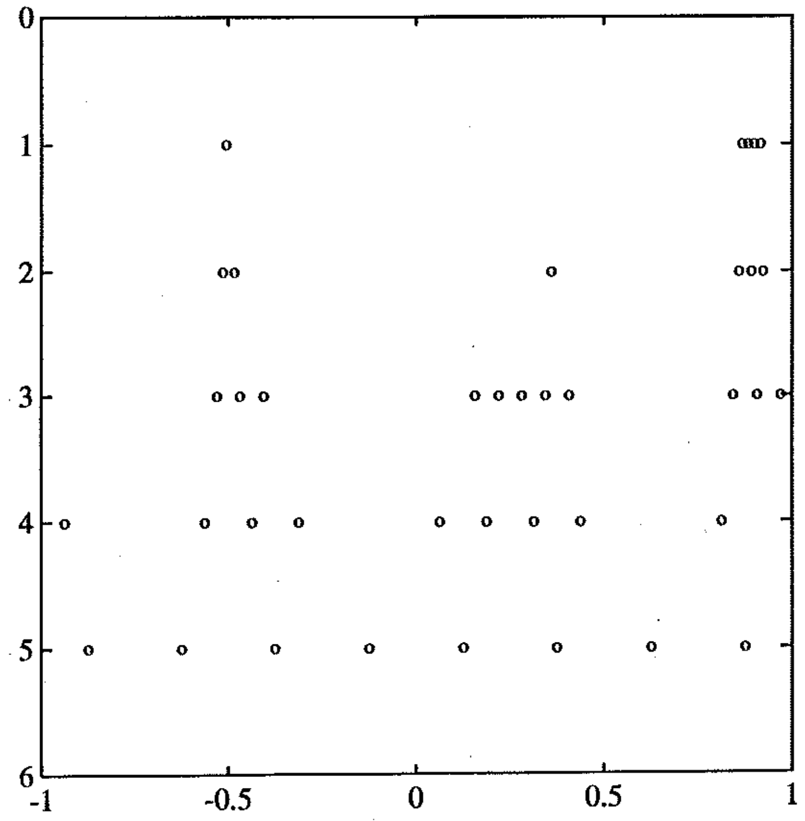


Figure 1c. Data compression of (4.8), $n = 125$.



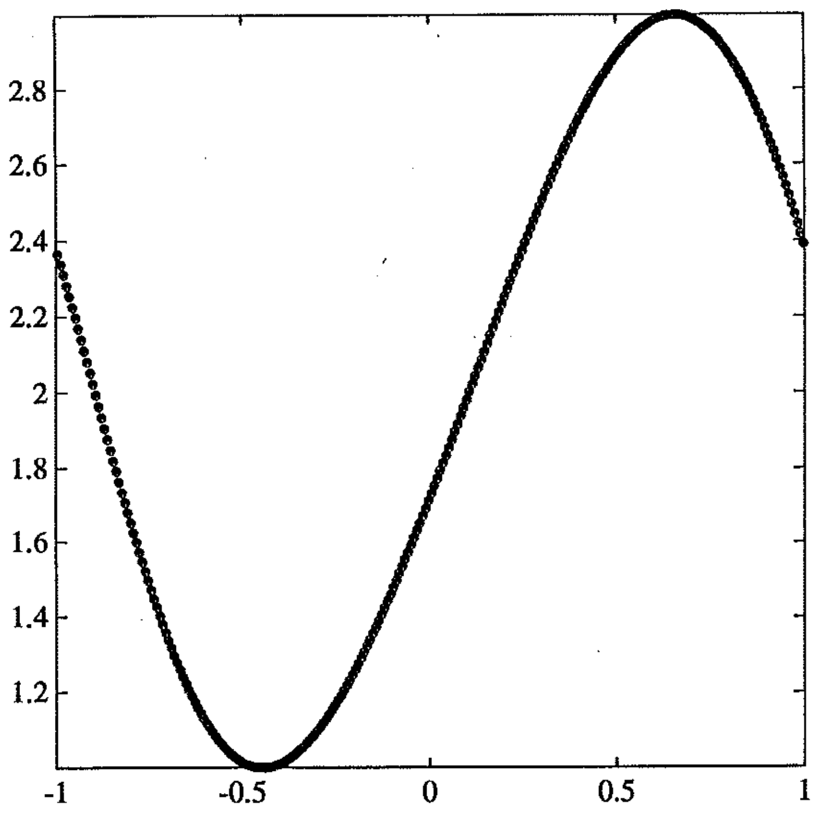
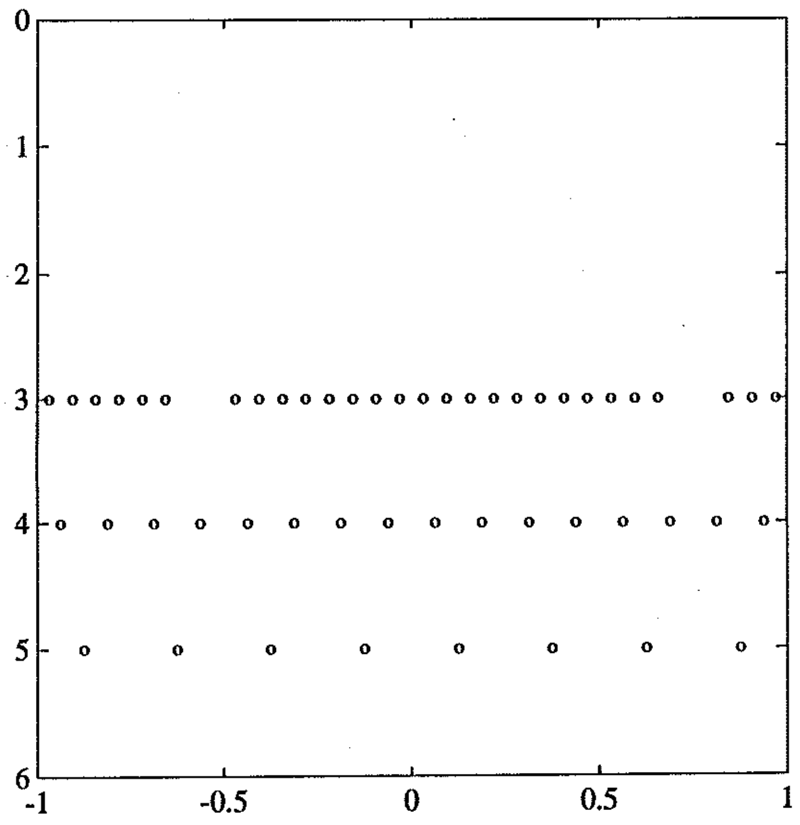


Figure 2a. Data compression of (4.10), $n = 25$.



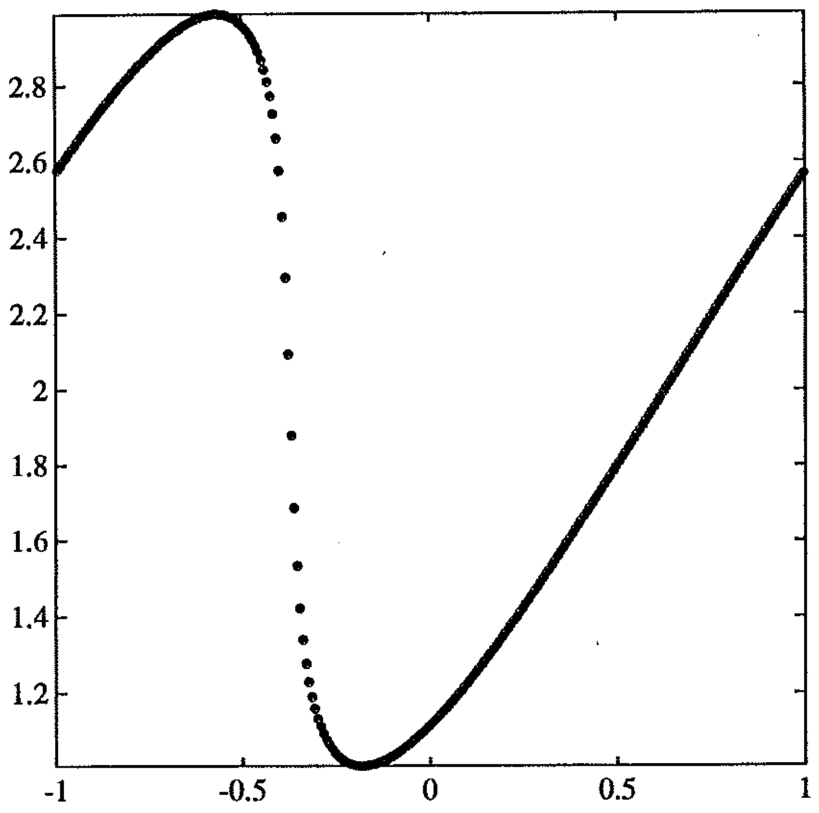
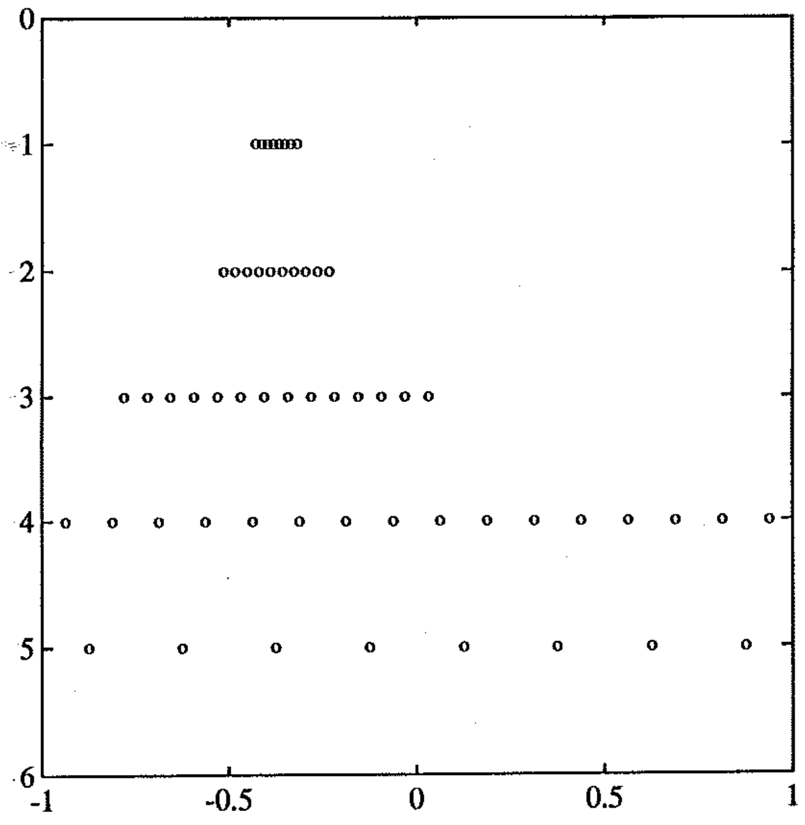


Figure 2b. Data compression of (4.10), $n = 150$.



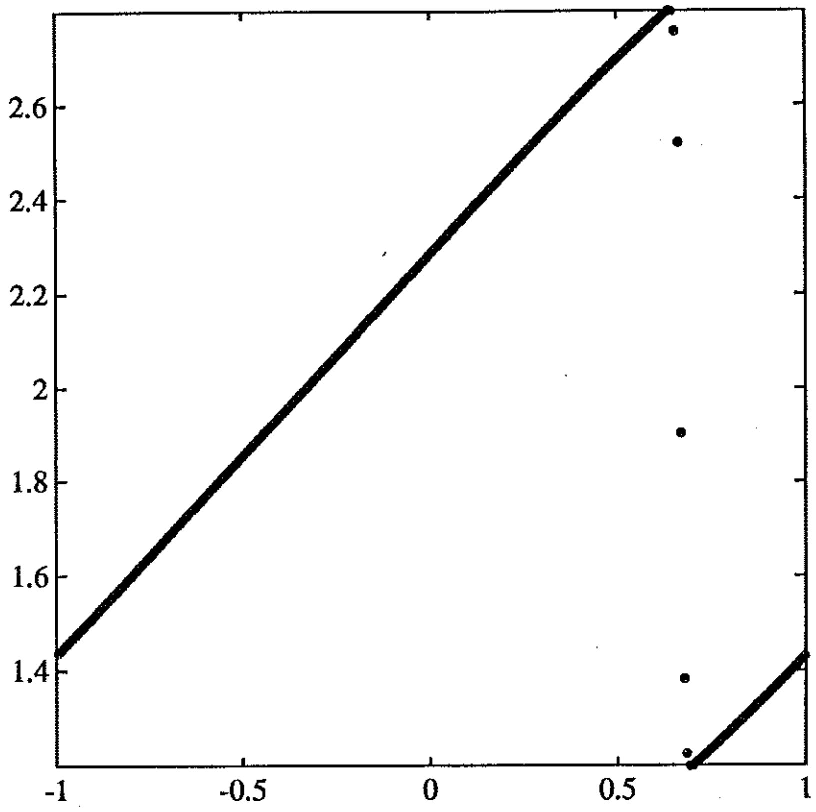
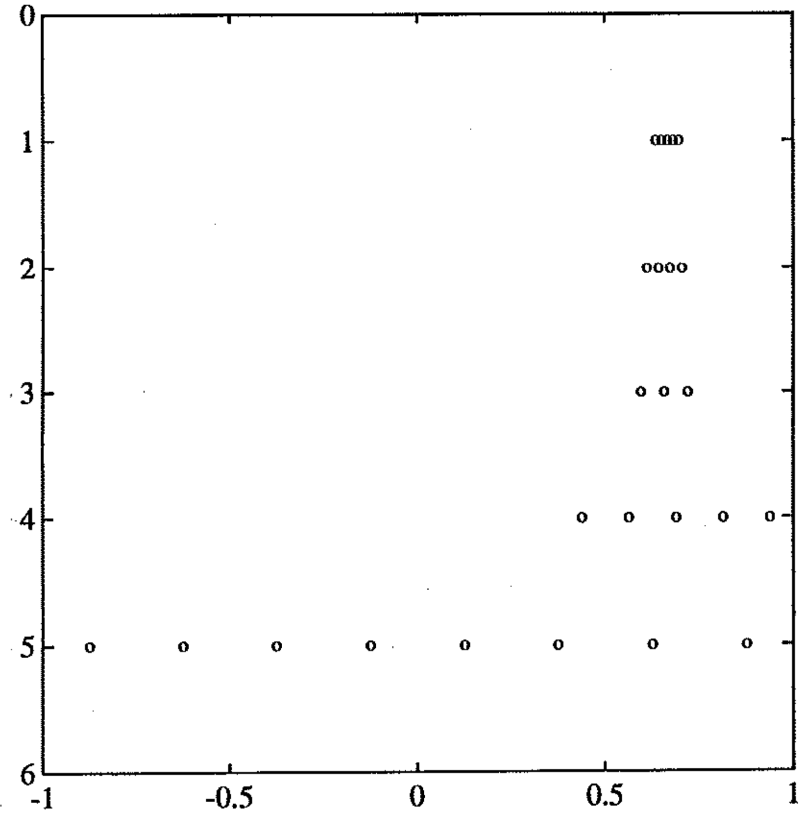


Figure 2c. Data compression of (4.10), $n = 400$.



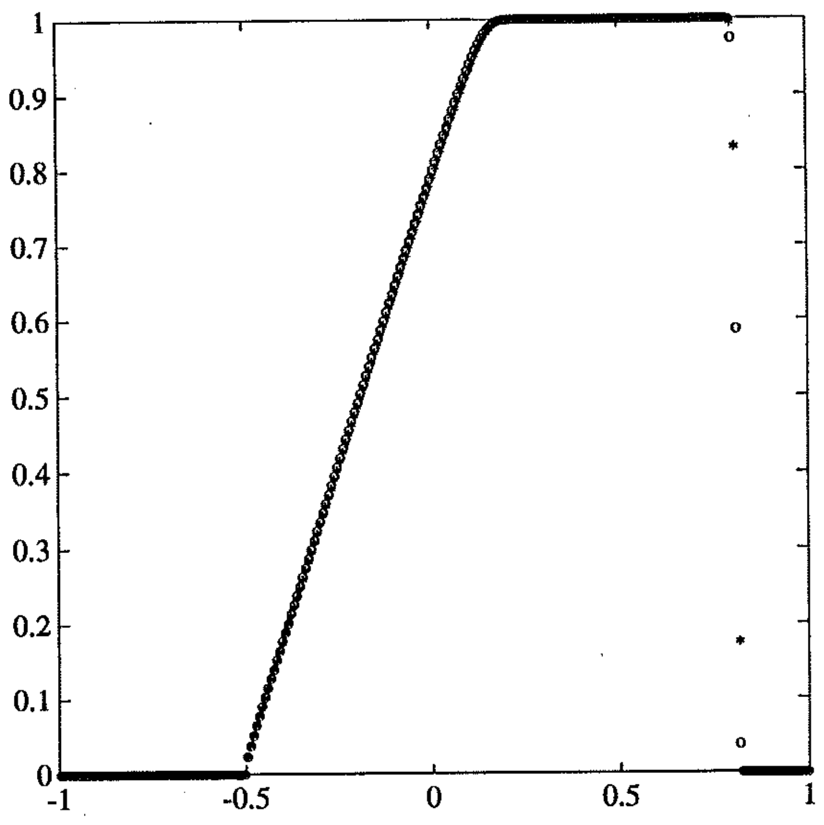
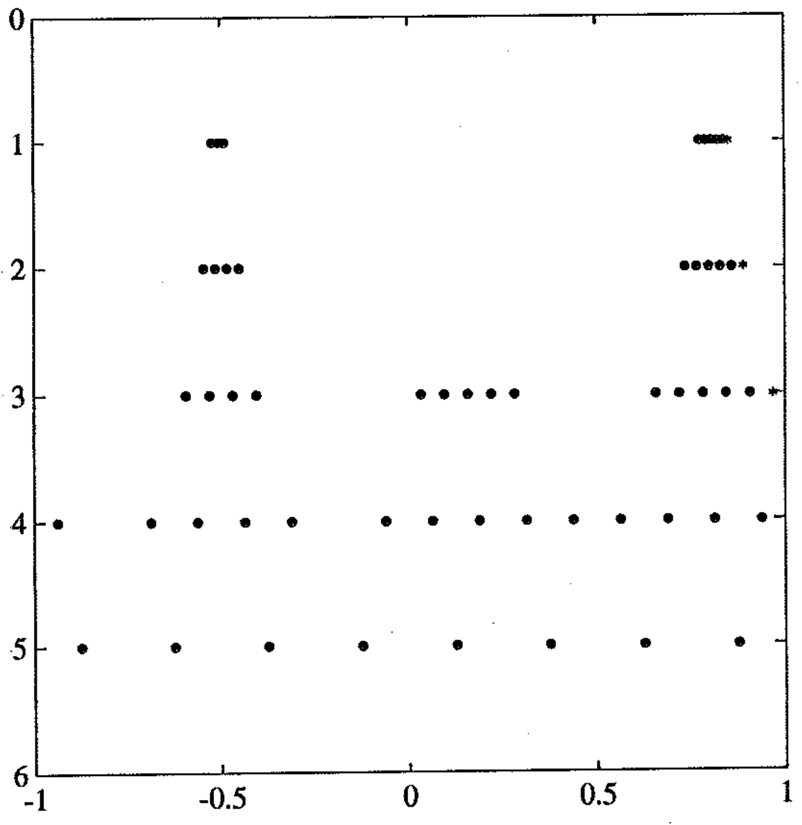


Figure 3a. $\mathcal{D}_\epsilon(v^n)$ and $\mathcal{D}_\epsilon(v^{n+1})$ for (4.8), $n = 100$.



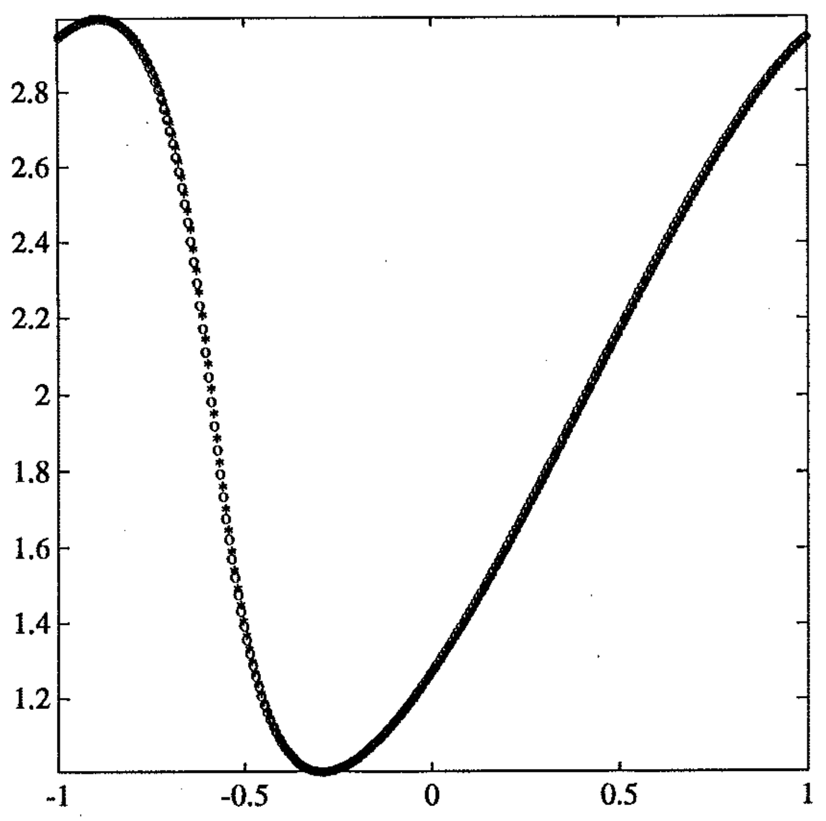
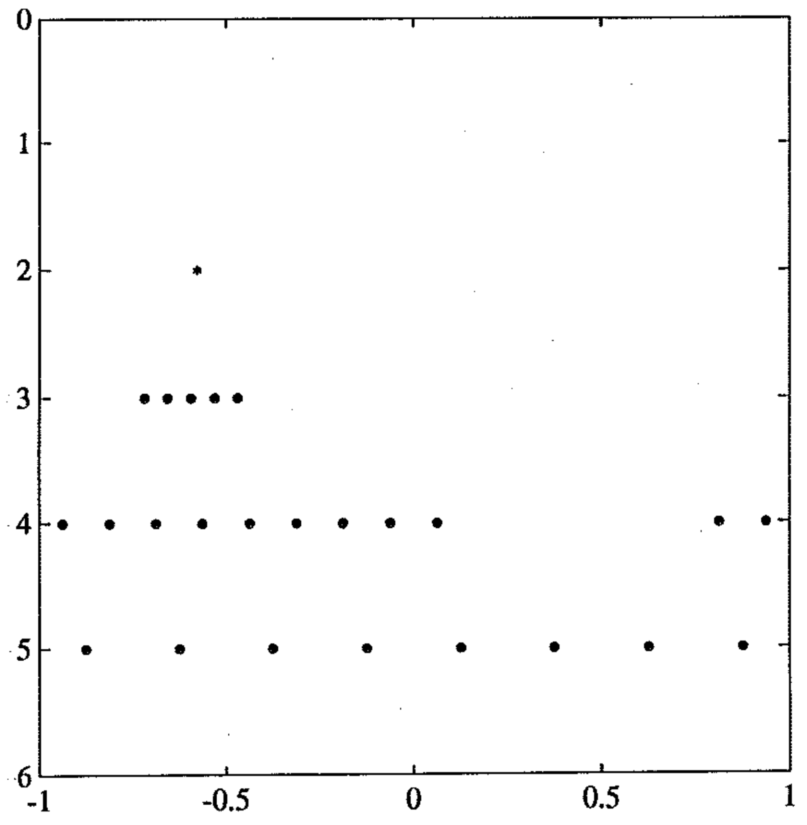


Figure 3b. $\mathcal{D}_\varepsilon(v^n)$ and $\mathcal{D}_\varepsilon(v^{n+1})$ for (4.10), $n = 99$.



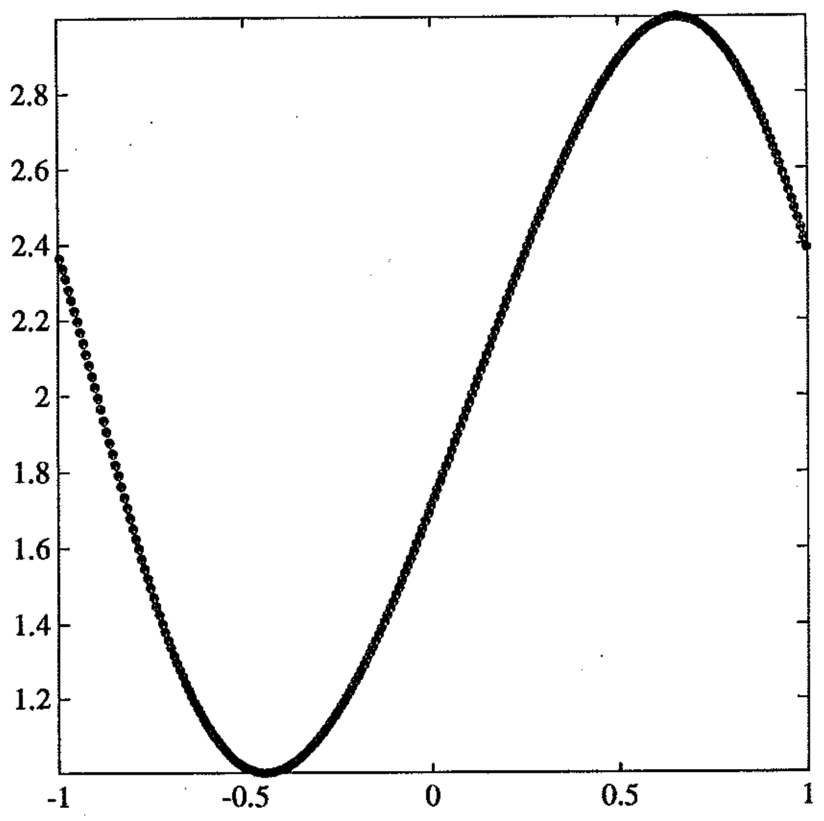
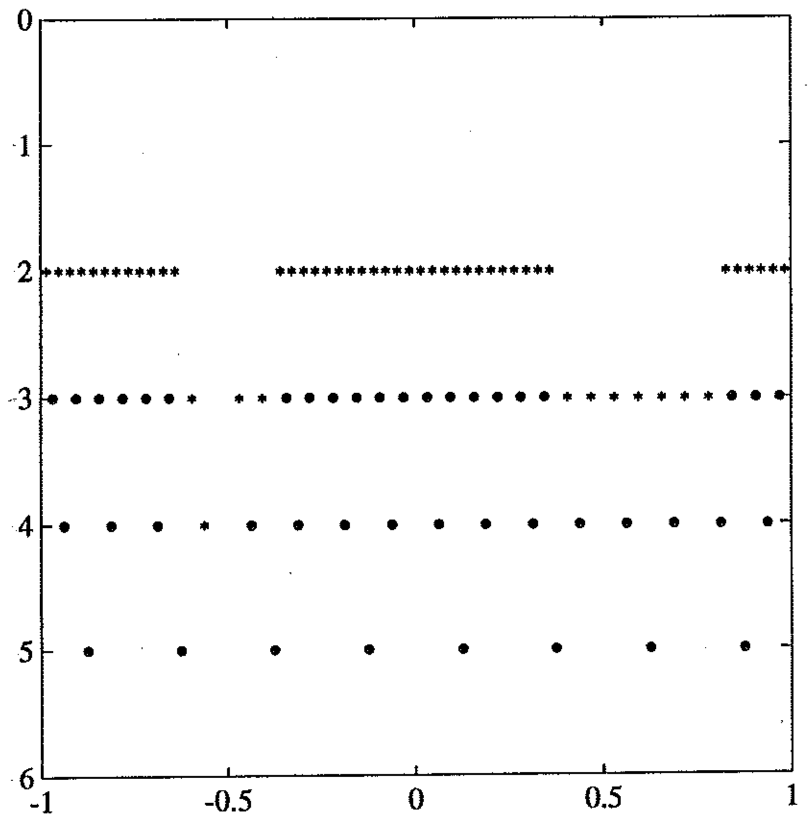


Figure 4a. Multiresolution algorithm, $\bar{r} = 3,\ \bar{p} = -1,\ n = 25.$



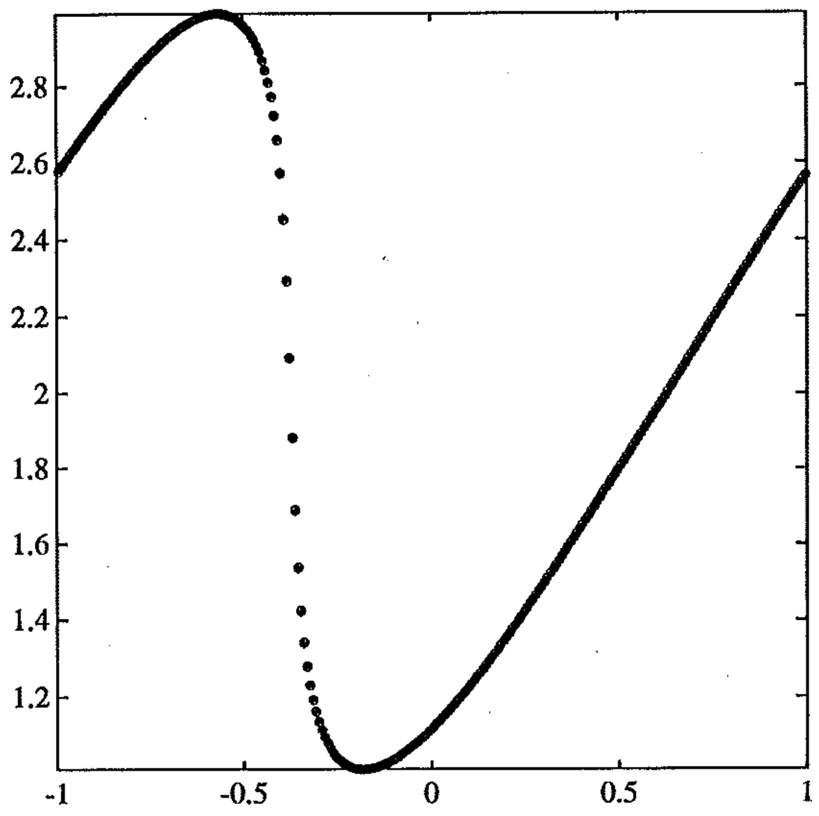
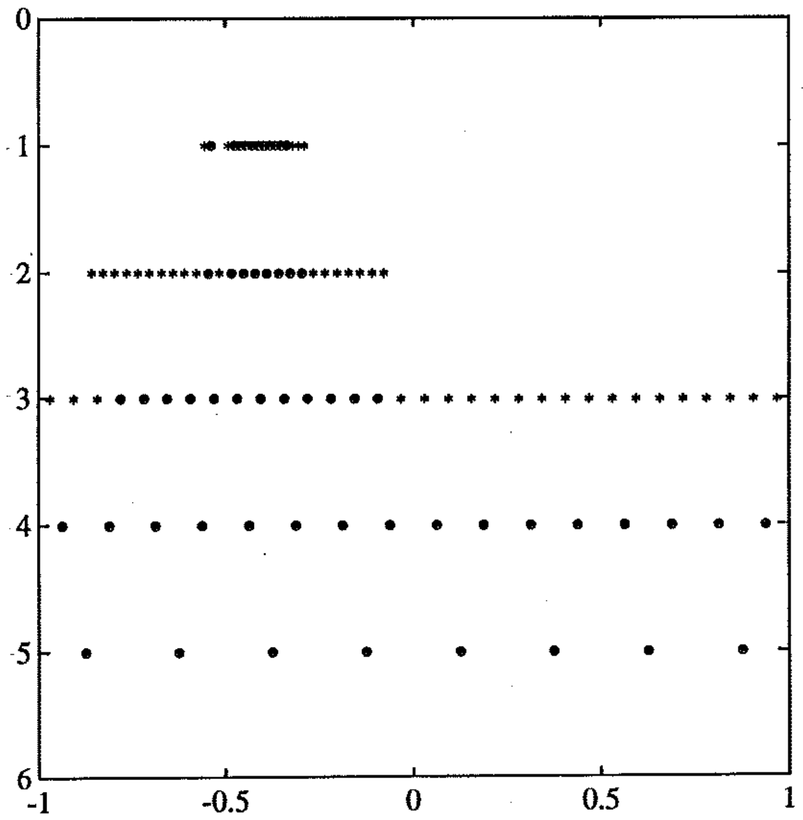


Figure 4b. Multiresolution algorithm, $\bar{r} = 3$, $\bar{p} = -1$, $n = 150$.



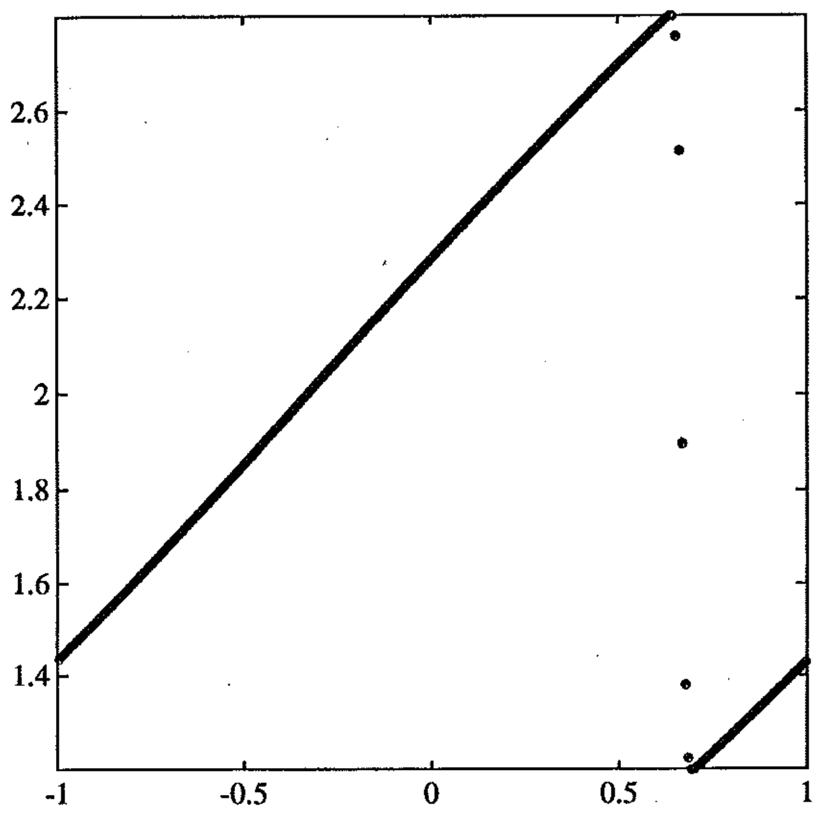
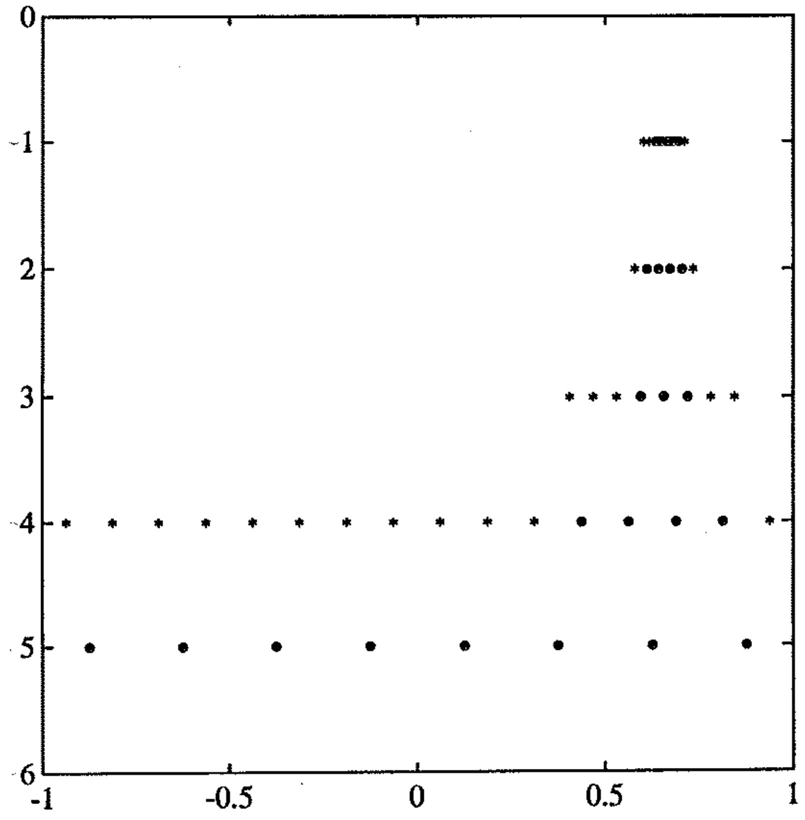


Figure 4c. Multiresolution algorithm, $\bar{r} = 3,\ \bar{p} = -1,\ n = 400.$



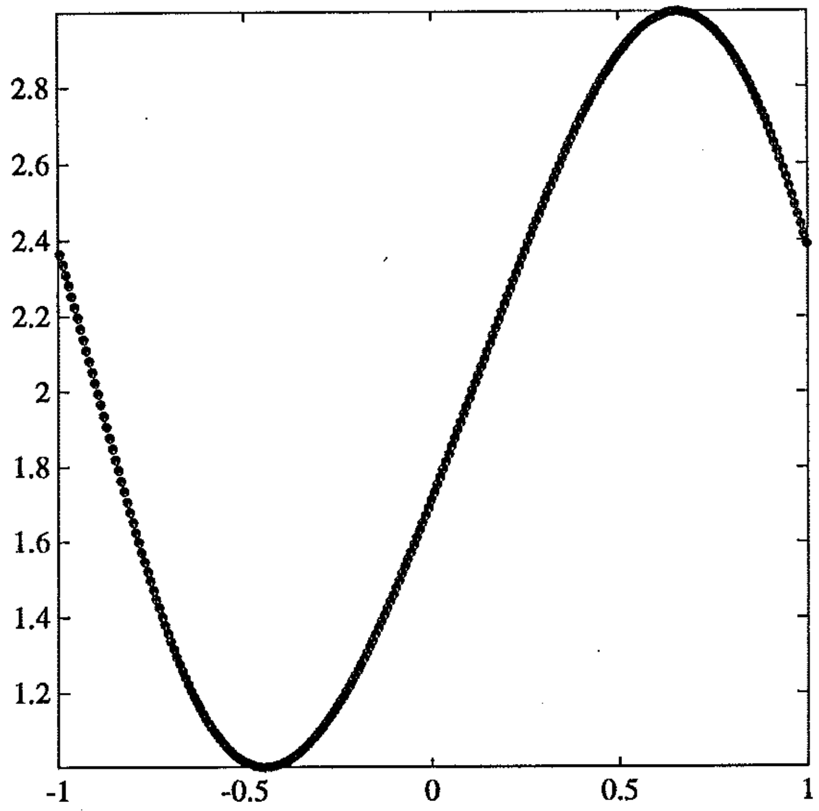
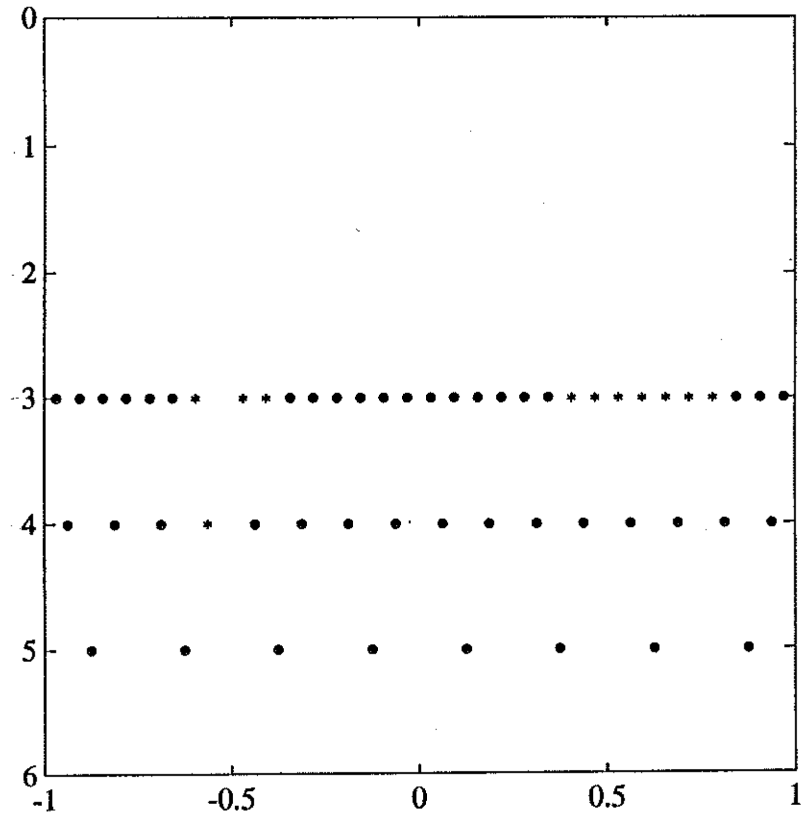


Figure 5a. Multiresolution algorithm, $\bar{r} = 3,\ \bar{p} = 2,\ n = 25.$



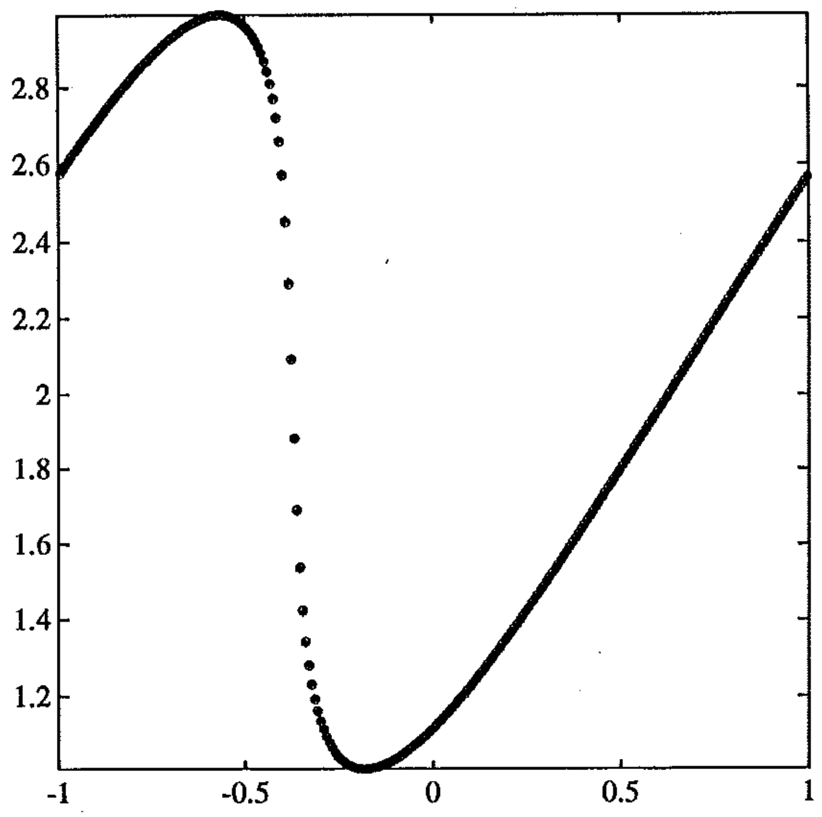
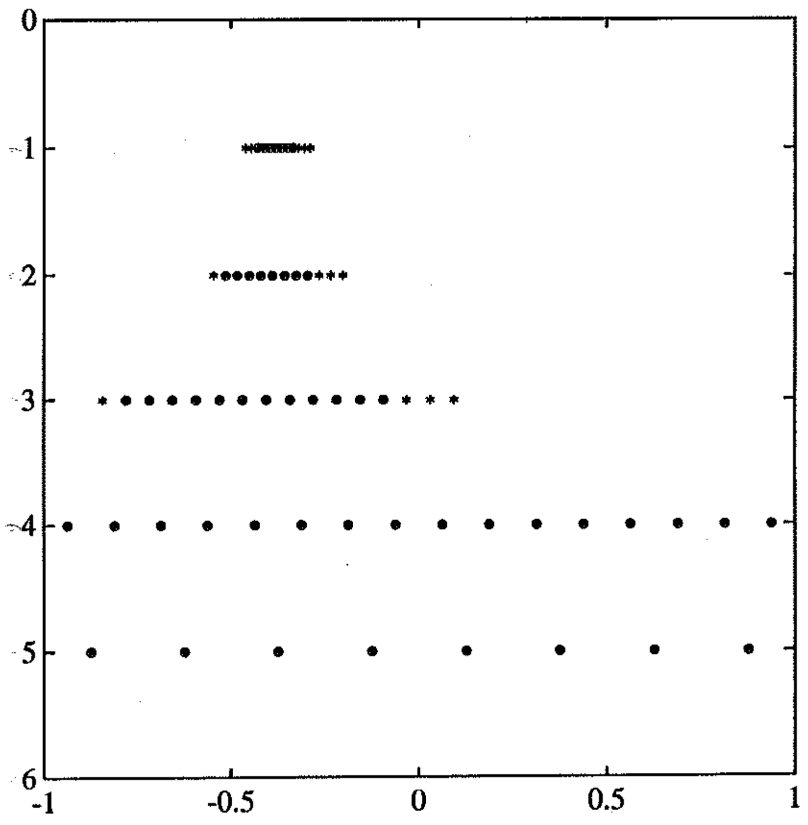


Figure 5b. Multiresolution algorithm, $\bar{r} = 3,\ \bar{p} = 2,\ n = 150.$



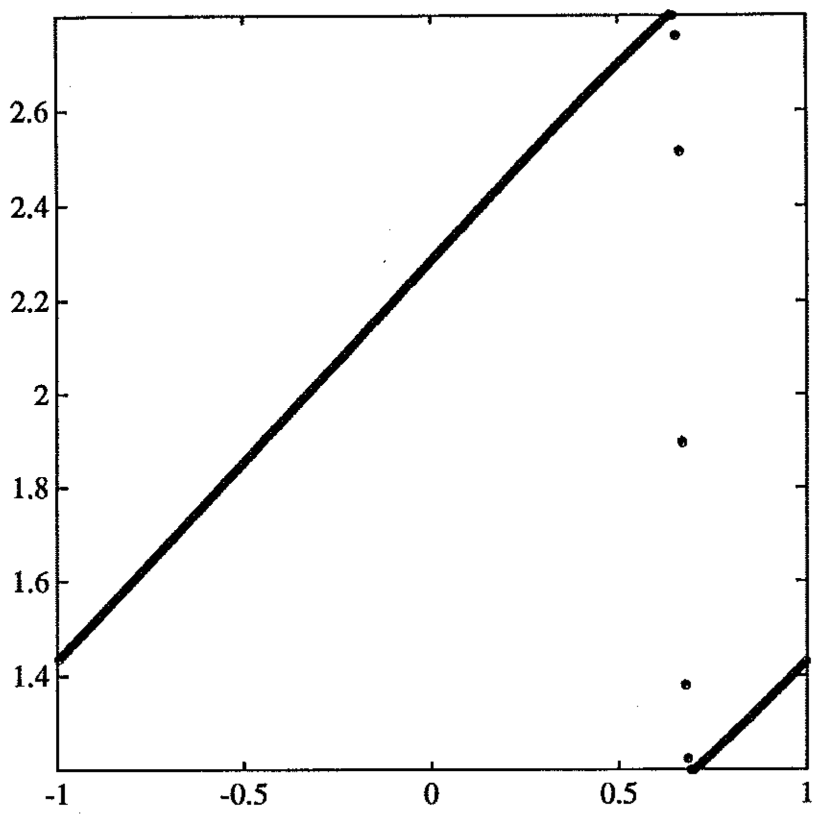
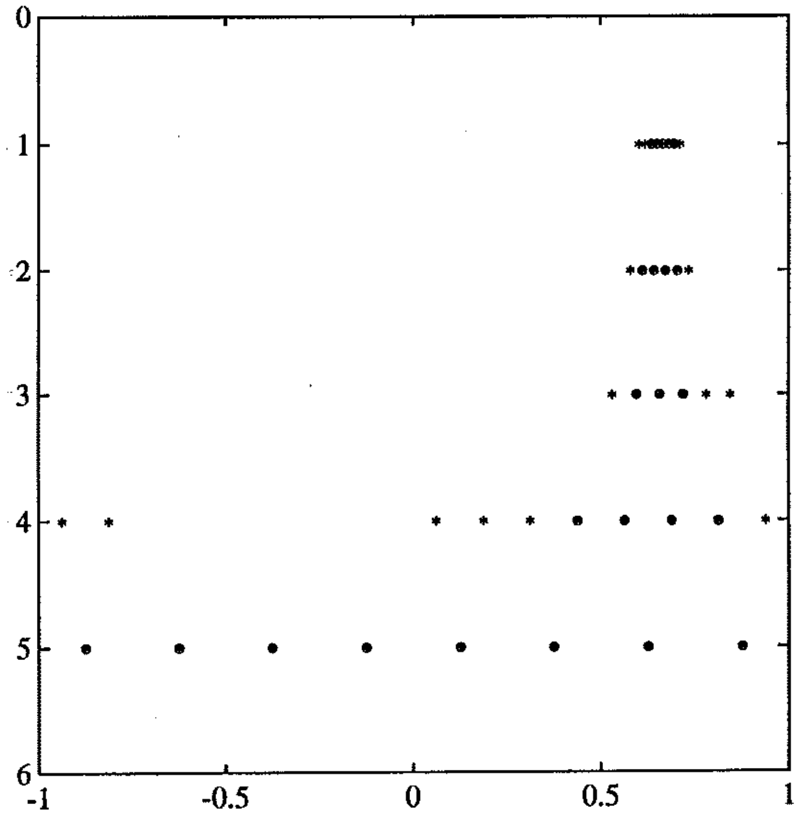


Figure 5c. Multiresolution algorithm, $\bar{r} = 3,\ \bar{p} = 2,\ n = 400$.



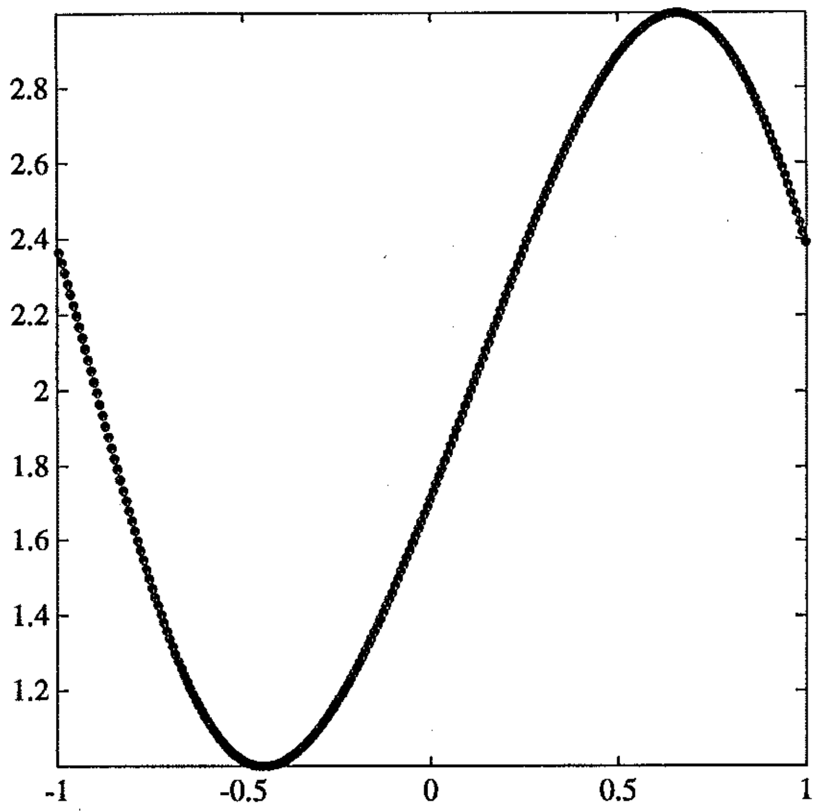
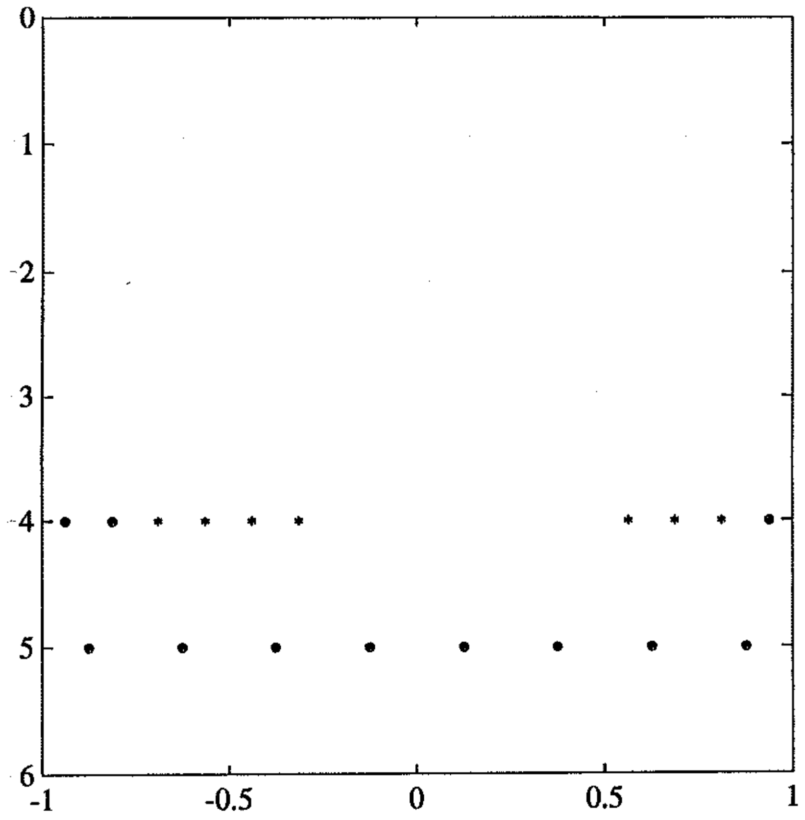


Figure 6a. Multiresolution algorithm, $\bar{r} = 5,\ \bar{p} = 4,\ n = 25$.



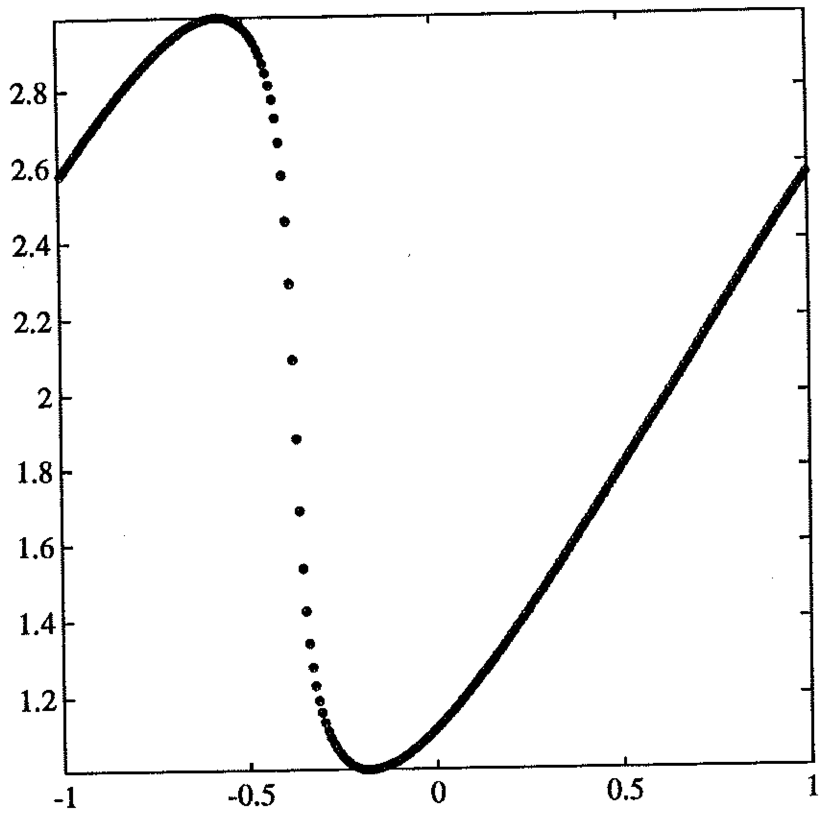
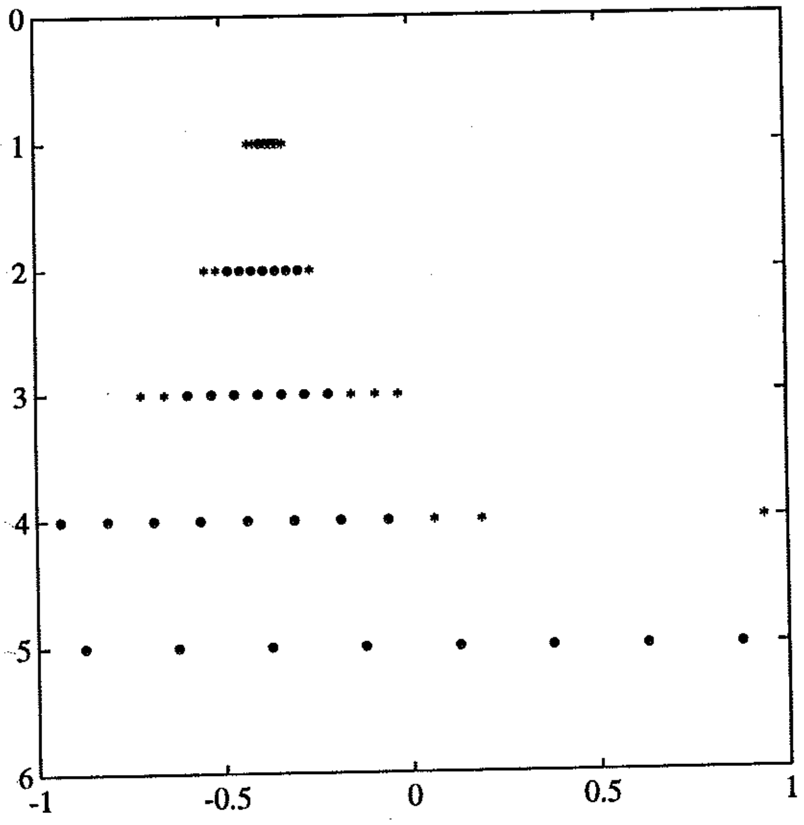


Figure 6b. Multiresolution algorithm, $\bar{r} = 5,\ \bar{p} = 4,\ n = 150.$



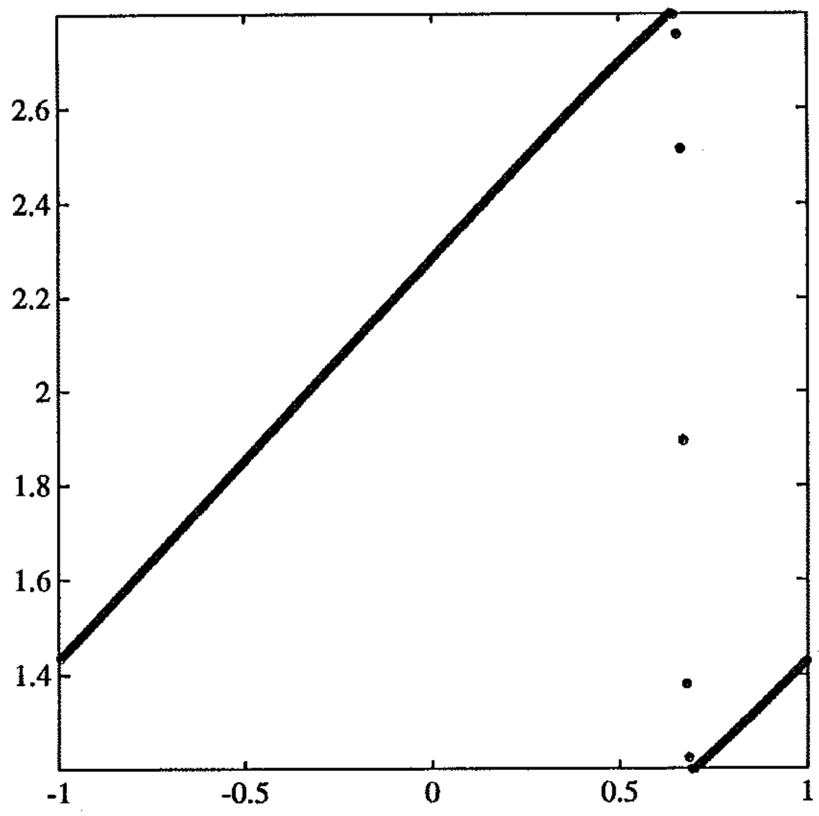
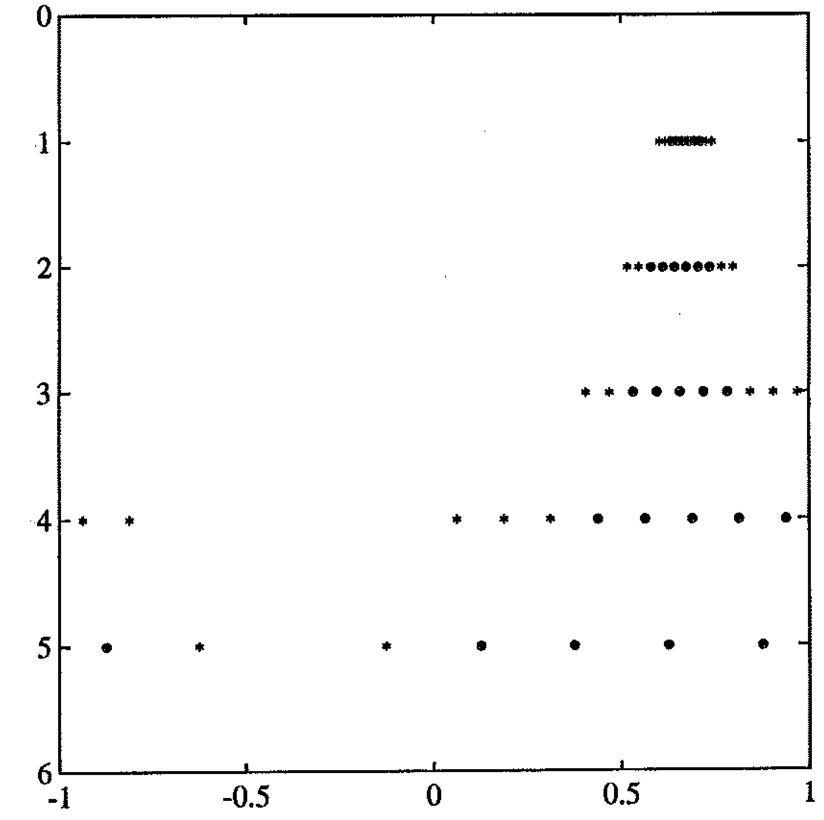


Figure 6c. Multiresolution algorithm, $\bar{r} = 5$, $\bar{p} = 4$, $n = 400$.



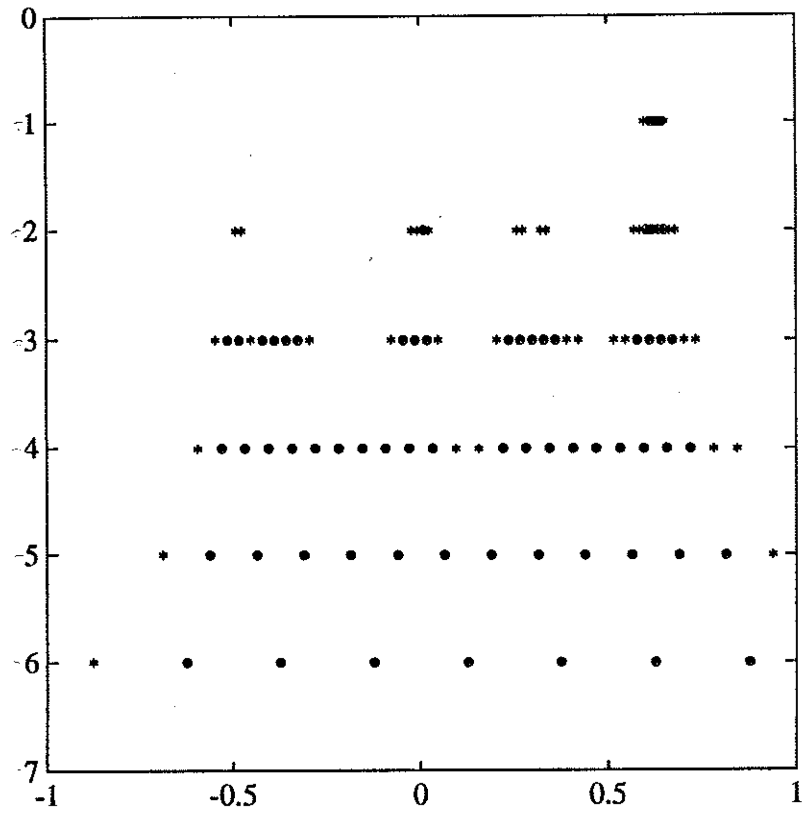


Figure 7a. $\mathcal{D}_\varepsilon(v^n)$ and $\tilde{\mathcal{D}}^n$, $n = 250$.

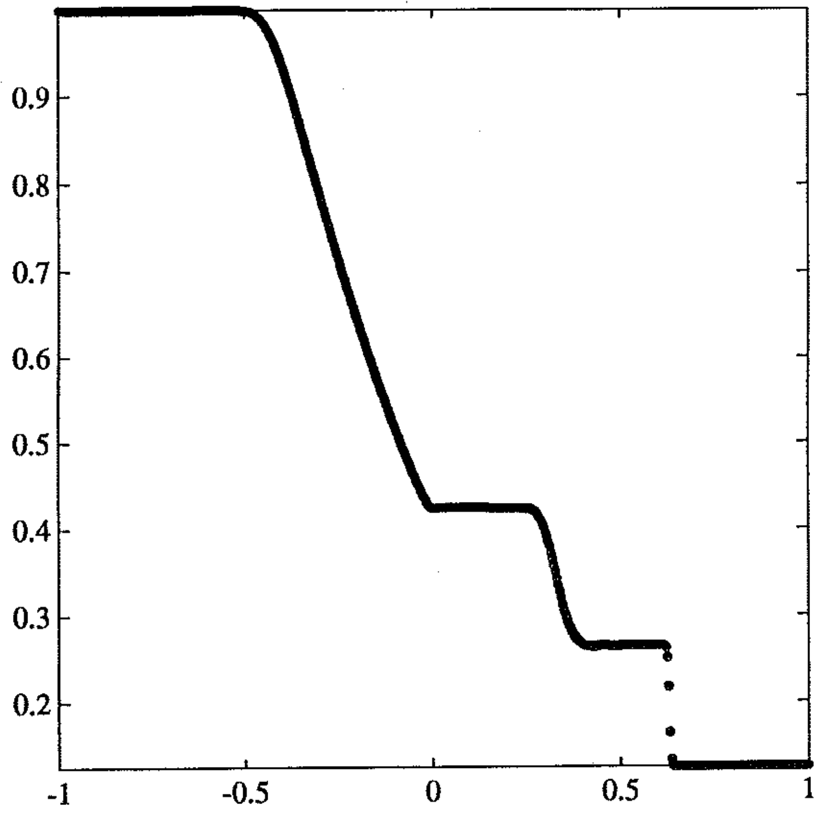


Figure 7b. Density.

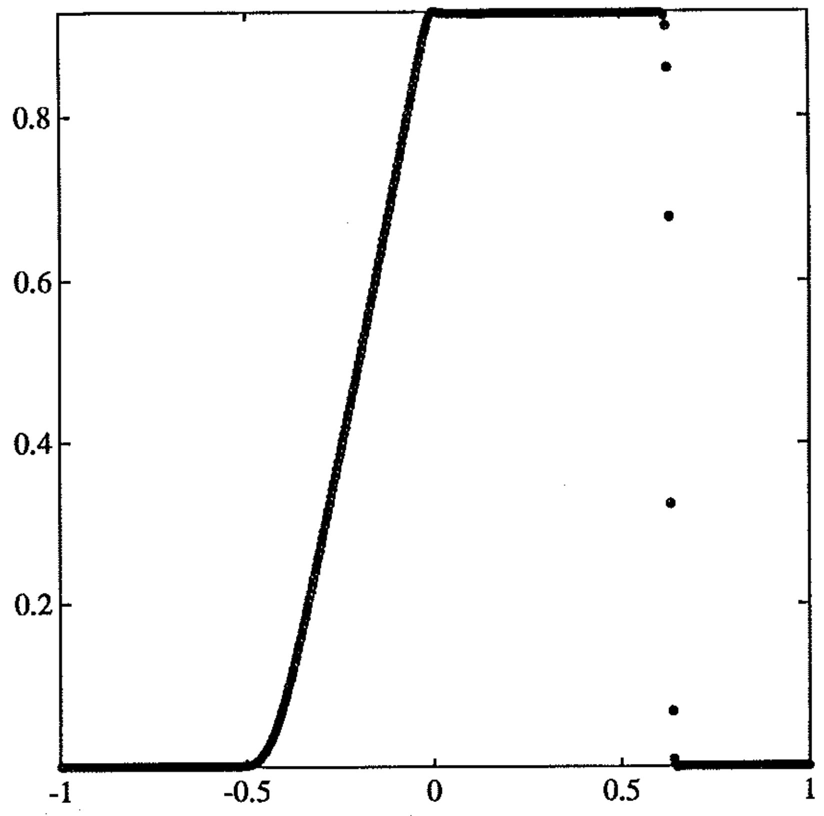


Figure 7c. Velocity.

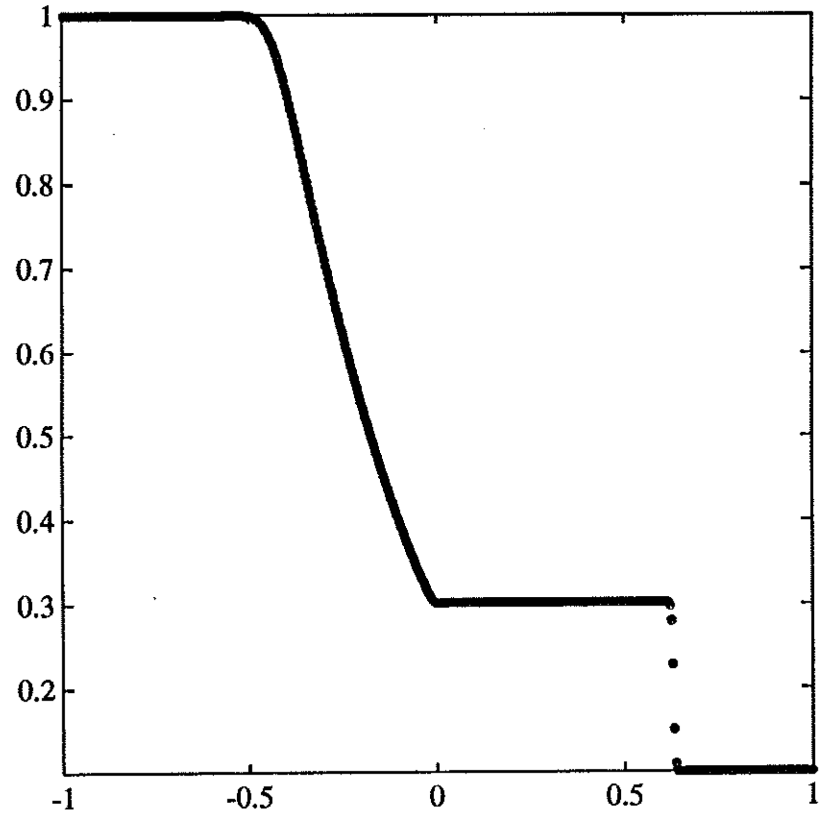


Figure 7d. Pressure.