UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes

Chi-Wang Shu
Stanley Osher

February 1987

CAM Report 87-11

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

I. Introduction. In this paper we are interested in solving the system of hyperbolic conservation laws

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} = 0 \qquad (\text{or} = g(u,x,t), \text{ a forcing term}) \tag{1.1a}$$

$$u(x,0) = u_0(x) \tag{1.1b}$$

Here $u = (u_1, \ldots, u_m)^T$, $x = (x_1, x_2, \ldots, x_d)$, and any real combination of the Jacobian matrices $\sum_{i=1}^{d} \xi_i \frac{\partial f_i}{\partial u}$ has m real eigenvalues and a complete set of eigenvectors.

On a computational grid $x_j = j \cdot \Delta x$, $t_n = n\Delta t$, we use u_j^n to denote the computed approximation to the exact solution $u(x_j, t_n)$ of (1.1).

We also use the abstract form

$$u_t = \mathcal{L}(u) \tag{1.2}$$

in place of (1.1a). Here $\mathcal{L}$ is a spatial operator.

As is well known, the solution to (1.1) may develop discontinuities (shocks, contact discontinuities, etc.) even if the initial condition $u_0(x)$ in (1.1b) is a smooth function. Traditional finite difference methods, even if linearly stable, often give poor results in the presence of shocks and other discontinuities. Recently there has been a lot of activity geared towards constructing efficient finite difference approximations to (1.1). These include TVD (total-variation-diminishing), TVB (total-variation-bounded) and ENO (essentially non-oscillatory) methods. See, e.g. [2], [3], [4], [5], [6], [9], [10], [12], [13], [14], and the references listed therein. Many of the ideas can be traced back to Van Leer's work in [15], [16].

Usually, rigorous analysis (e.g. total-variation stability, convergence) is only done for the scalar, one-dimensional nonlinear case (i.e. $d = m = 1$ in (1.1)). Some

partial theory (e.g. convergence for first order monotone schemes, and maximum norm stability for higher order TVD schemes) exists for scalar multi-dimensional problems ($d > 1$ in (1.1)), but a full convergence theory for multidimensional nonlinear systems appears to be extremely difficult. However, numerical experiments for multi-dimensional problems and/or for systems of equations, using direct generalizations of TVD, TVB and ENO schemes give very good results. Again, see, e.g., [2], [4], [6], [10], [11]. We now shall confine our discussion at first to this one space dimension, scalar case. Systems and multi-dimensional problems are discussed at the end of Section 3.

We shall always use conservative schemes of the form

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}), \quad \lambda = \frac{\Delta t}{\Delta x} \tag{1.3}$$

with a consistent numerical flux

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j-\ell}, \ldots, u_{j+k}); \qquad \hat{f}(u, \ldots, u) = f(u) \tag{1.4}$$

in order to guarantee that any convergent bounded a.e. subsequence has as its limit a weak solution of (1.1), (Lax-Wendroff Theorem [8]), i.e. we construct so-called "shock capturing methods".

The total variation of a discrete scalar solution is usually defined by

$$TV(u) = \sum_j |u_{j+1} - u_j| \tag{1.5}$$

We say the scheme is TVD if

$$TV(u^{n+1}) \leq TV(u^n) \tag{1.6}$$

and TVB in $0 \leq t \leq T$ if

$$TV(u^n) \leq B \tag{1.7}$$

for some fixed B depending only on $TV(u^0)$, and for all n and Δt such that $0 \leq n\Delta t \leq T$.

A nice theoretical advantage of all TVD or TVB schemes is that they have convergent subsequences as $\Delta x \to 0$, and, if a further "entropy condition" is satisfied, then they are convergent. See, e.g. [3].

The formal "order of accuracy" in this paper is in the sense of local truncation errors, i.e. if local truncation error is $0(\Delta x^{r+1})$ in smooth regions, we say the scheme is (formally) r-th order accurate. See, e.g. [2].

There are many TVD schemes constructed in the literature (e.g. [2], [3], [9], [10], [14]). In [10], TVD schemes of very high spatial order (up to 15th order) were constructed. These schemes can be used for steady state calculations (e.g. implemented with the TVD Runge-Kutta type time discretizations with large CFL numbers in [13]) or for time dependent problems, equipped with a multi-level TVD high order time discretization in [13] or with a Runge-Kutta type TVD high order time discretization in Section 2 of this paper. These are perhaps the highest order TVD schemes existing at present. However the definition of total variation (1.5) implies that these methods must degenerate to first order accuracy at extremacy. A TVB modification of such schemes which recovers global high order accuracy even at critical points is obtained in [12].

The above mentioned TVD and TVB schemes use a *fixed*, wide stencil (for the 15th order scheme, the stencil is 17 points wide), thus restricting the advantage of

going to higher order through smearing of discontinuities and resulting degradation of the accuracy. Numerically we observed that third order schemes work quite well [12], but we lost accuracy in a fairly large region near discontinuities by using a fifth order method. Recently Harten, Osher, Engquist and Chakravarthy constructed ENO schemes which are of globally high order accuracy in smooth regions and which use adaptive stencils, thus obtaining information from regions of smoothness if discontinuities are present. These methods achieve high order accuracy right up to discontinuities. Analysis and numerical experiments are found in [5], [6], [4]. At present, a convergence theory (e.g. TV boundedness) for ENO schemes is still unavailable.

There are two natural directions in which to simplify the ENO or TVD, TVB schemes, especially for multi-dimensional problems or problems with forcing terms:

(1) *Time discretization.* Usually, semi-discrete (method of lines) versions of ENO or TVD, TVB schemes are much simpler than the fully discrete ones. There are then mainly two ways to discretize in time. One is of Lax-Wendroff type, i.e., by using $u_t = -f_x$, $u_{tt} = (f'f_x)_x, \ldots, u_j^{n+1} = u_j^n + \Delta t(u_t)_j^n + \frac{\Delta t^2}{2}(u_{tt})_j^n + \cdots$, and then by discretizing the spatial derivatives. Many second order TVD schemes (e.g. Harten's in [2]), and the ENO schemes in [5], [6], [4], used this type of time discretization. The main disadvantages to the procedure is that it is complicated to program, especially for multi-dimensional problems with forcing terms. One can see this by writing out a third order approximation to the equation $u_t + f_1(u)_{x_1} + f_2(u)_{x_2} = g(u, x_1, x_2, t)$. Moreover it is not easy to prove that this results in a TVD or TVB method, even if the original method-of-lines ODE and its Euler forward version are both TVD or TVB. The numerical results have proven satisfactory, but, speaking theoretically, only second order in time TVD or

TVB schemes exist of this type. Another way to discretize in time is to use a multi-level or Runge-Kutta type ODE solver. This is much simpler to program than the Lax-Wendroff type of discretization for multi-dimensional problems or for problems with forcing terms, so it is widely used for numerically implementing a method of lines approximation. However, usually only *linear* stability analysis is available in the literature, which is certainly not enough for our purpose since linear stability *does not* imply convergence if shocks or other discontinuities are present. This is particularly true for ENO schemes which use moving stencils. Linear stability analysis is based on the fact that the stencil is fixed and the error accumulates in a predictable pattern, hence it does not apply to ENO schemes at all. For these reasons we consider TVD time discretizations . In [13], a class of multi-level TVD time discretizations were constructed and analysed, (numerical results can be found in [12]). However, for easy starting and for storage considerations, one step Runge-Kutta type schemes are preferable to multi-level methods. In Section II of this paper we present a class of high order *TVD* Runge-Kutta type time discretizations.

(2). Avoiding the using of cell-averages. The ENO schemes constructed in [5], [6], [4] are for cell-averages but involve point values as well. Hence a reconstruction procedure is needed to recover point values from cell averages to the correct order, which can be rather complicated, especially in multi-dimensional problems. It is desirable to use the moving-stencil idea directly on fluxes to get ENO schemes without using cell-averages. In Section III of this paper a class of such ENO schemes is constructed.

Some encouraging numerical results obtained by using schemes constructed in this paper are included in Section IV.

We conclude these introductory remarks by noting that R. Sanders [17] has recently devised third order accurate TVD methods which degenerate to second order at extrema. He defines the variation of the numerical solution as the variation of an appropriately chosen piecewise parabolic interpolant. The numerical results are very good. However this technique has no method of lines analogue, so we omit it from our present discussion.

II. High Order Runge-Kutta Type TVD Time Discretizations. Define

$$w = T(u) = (I + \Delta t L)(u) \tag{2.1}$$

where T and L are nonlinear discrete operators, L is a r-th order discrete approximate to the spatial operator $\mathcal{L}$ in (1.2):

$$L(u) = \mathcal{L}(u) + O(\Delta x^r) \tag{2.2}$$

if u is smooth.

Our goal is to get a fully r-th order approximation to the differential equation (1.2) of the form

$$u^{n+1} = S(u^n) \tag{2.3}$$

(The operator S depends on T). This means that if $u(x,t)$ is an exact smooth solution of (1.2), then

$$u(x_j, t^{n+1}) - S(u^n)_j = O(\Delta x^{r+1}). \tag{2.4}$$

We also want the scheme to be TVD:

$$TV(S(u)) \leq TV(T(u)) \tag{2.5}$$

under suitable restrictions on Δt (or, equivalently, on the CFL number λ).

We call a time discretization (2.3) r-th order TVD if it satisfies (2.4) and (2.5). If the spatial operator T in (2.1) is TVD or TVB:

(2.6a) $$TV(T(u)) \leq TV(u)$$

or

(2.6b) $$TV(T(u)) \leq TV(u) + M\Delta t$$

for $0 \leq M$ uniformly bounded as $\Delta t \to 0$, then the fully discrete high order scheme (2.3) is TVD or TVB, owing to (2.5).

In [13], a class of multi-level type high order TVD time discretizations was constructed. Numerical experiments in [12] were very promising . But there are two disadvantages of multi-level type methods: (i) for an m-th level method the first $m-1$ levels have to be calculated by other methods to the same order of accuracy (e.g. by using Taylor series expansions); (ii) we have to store all m level datas, creating a rather large storage requirement, stretching up to and beyond the limits of present day computers for physical problems arising e.g. in computational aeronautics. At present, Runge-Kutta type methods are more often used in discretizing the method-of-lines than are multi-level methods. Since the former consists of one-level methods, they are self-starting and reduce storage requirements significantly. In the following we will analyze the *nonlinear* stability (TVD) of a class of such methods.

Assume (2.1) is TVD (or TVB) under a suitable CFL restriction

(2.7) $$\lambda \leq \lambda_0$$

We may also need an approximation to $-\tilde{L}$ to the spatial operator $-\mathcal{L}$ which we take to satisfy

(2.8) $$\tilde{w} = \tilde{T}(u) = (I - \Delta t\tilde{L})(u)$$

(2.9) $$\tilde{L}(u) = \mathcal{L}(u) + O(\Delta x^r)$$

where (2.8) is TVD (or TVB) under the same CFL restriction (2.7).

As an example, a very simple first order ($r = 1$) TVD approximation to

$$u_t = u_x = \mathcal{L}(u)$$

is obtained via simple upwind differencing:

$$L(u) = \frac{u(x + \Delta x) - u(x)}{\Delta x}$$

This scheme (2.1) is TVD if $\lambda = \frac{\Delta t}{\Delta x} \leq \lambda_0 \leq 1$.

The approximation $\tilde{L}(u)$ is defined as

$$\tilde{L}(u) = \frac{u(x) - u(x - \Delta x)}{\Delta x}$$

and (2.8) is TVD again for $\lambda = \frac{\Delta t}{\Delta x} \leq \lambda_0 \leq 1$.

This procedure (2.8), (2.9) easily generalizes to any conventional TVD, TVB, or ENO-approximation (2.1) satisfying (2.2).

The general explicit Runge-Kutta method (we use explicit methods to avoid solving nonlinear equations) for (2.1) is

$$u^{(i)} = u^{(0)} + \Delta t \sum_{k=0}^{i-1} c_{ik} L(u^{(k)}), \qquad i = 1, 2, \ldots, m \tag{2.10a}$$

$$u^{(0)} = u^n, \qquad u^{(m)} = u^{n+1} \tag{2.10b}$$

If the operator L also depends explicitly on t, as is the case when the forcing term g in (1.1a) depends explicitly on t, or when we have time-dependent boundary conditions, the general explicit Runge-Kutta method takes a more complicated form

$$u^{(i)} = u^{(0)} + \Delta t \sum_{k=0}^{i-1} c_{ik} L(u^{(k)},\ t^{(0)} + d_k \Delta t) \tag{2.11a}$$

where

(2.11b) $$d_k = \sum_{\ell=0}^{k-1} c_{k\ell}$$

For details, see any numerical ODE text, e.g. [1], [7] (any such method is usually written in a slightly different form, using $k_1, k_2, \ldots$, but that form is clearly equivalent to (2.10) or (2.11)).

We shall restrict our attention to (2.10); the generalization to (2.11) is clearly straightforward, using (2.11b).

In order to get conditions for TVD, we rewrite (2.10) as follows: For $\alpha_{ik} \geq 0$, $\sum_{k=0}^{i-1} \alpha_{ik} = 1$, we have

$$\begin{aligned} u^{(i)} &= \sum_{k=0}^{i-1} \alpha_{ik} u^{(0)} + \Delta t \sum_{k=0}^{i-1} c_{ik} L(u^{(k)}) \\ &= \alpha_{i0} u^{(0)} + \sum_{k=1}^{i-1} \alpha_{ik} (u^{(\ell)} - \Delta t \sum_{\ell=0}^{k-1} c_{k\ell} L(u^{(\ell)})) + \\ &\qquad + \Delta t \sum_{k=0}^{i-1} c_{ik} L(u^{(k)}) \\ &= \sum_{k=0}^{i-1} \left[\alpha_{ik} u^{(k)} + \left(c_{ik} - \sum_{\ell=k+1}^{i-1} c_{\ell k} \alpha_{i\ell}\right) \Delta t L(u^{(k)})\right] \end{aligned}$$

So if we let $\beta_{ik} = c_{ik} - \sum_{\ell=k+1}^{i-1} c_{\ell k} \alpha_{i\ell}$, then (2.10) may be written in the following equivalent form

(2.12) $$u^{(i)} = \sum_{k=0}^{i-1} [\alpha_{ik} u^{(k)} + \beta_{ik} \Delta t L(u^{(k)})]$$

It is well-known that we can get $(m+1)$-th order accurate methods in the form (2.10) or (2.11) for $m \leq 3$; m-th order methods for $m = 4, 5, 6$; or $(m-1)$-th order methods for $m = 7, 8$ (see, e.g. [1]).

For the classical 4-th order Runge-Kutta methods, the constants c_{ik} in (2.10) are all non-negative. However, β_{ik} in (2.12) may well be negative. In order to obtain TVD we apply a trick used in [13], i.e. we replace L in (2.12) by $\tilde{L}$ in (2.8)-(2.9) whenever β_{ik} is negative. Now (2.12) becomes a convex combination of TVD (or TVB) operators under the CFL restriction

$$\lambda \leq \lambda_0 \min_{i,k} \frac{\alpha_{ik}}{|\beta_{ik}|} \tag{2.13}$$

and we easily get the following

PROPOSITION 2.1. *Scheme (2.12) is TVD under the CFL restriction (2.13), if L is replaced by $\tilde{L}$ when β_{ik} is negative.*

REMARK 2.1. The previous proposition may be put into a more general framework as follows. The TV in (2.5) and (2.6)(a,b) may be replaced by $G(u)$, any convex mapping into the non-negative real line, where u, $S(u)$, $T(u)$ belong to a Banach space of functions $\mathbf{B}^{\Delta x}$. Also if u is a smooth solution of 1.2, then $u(x,t^{n+1}) - S(u(x,t^n)) = 0((\Delta x)^{r+1})$. The statement in the proposition can then be replaced by: $G(u^m) \leq G(u^0)$; moreover the formal order of accuracy is still r.

Now our goal is to choose the α_{ik} and β_{ik} such that (2.12) is of the highest possible order and such that the CFL restriction (2.13) is optimal. We would also like to minimize the number of negative β_{ik}'s in order to reduce the computational work involving $\tilde{L}$.

One easy way to do this is to use a standard Runge-Kutta method and then rewrite it in the form (2.12) to get α_{ik} and β_{ik}. Unfortunately, most classical Runge-Kutta methods lead to small CFL numbers in (2.13) as well as negative β_{ik}'s. Hence the best way is to consider (2.12) directly. Straightforward but tedious

Taylor expansions and an analysis of possible parameters (which we omit) leads us to the following results:

i) Second order case, $m = 1$.

For accuracy

$$\begin{cases} \alpha_{20} = 1 - \alpha_{21} \\ \beta_{20} = 1 - \frac{1}{2\beta_{10}} - \alpha_{21}\beta_{10} \\ \beta_{21} = \frac{1}{2\beta_{10}} \end{cases} \tag{2.14}$$

β_{10}, α_{21} are free parameters.

It can be verified that the "optimal" scheme (considering CFL restriction (2.13) and whether $\tilde{L}$ appears) is

$$\begin{cases} u^{(1)} = u^{(0)} + \Delta t L(u^{(0)}) \\ u^{(2)} = \frac{1}{2}u^{(0)} + \frac{1}{2}\Delta t L(u^{(1)}) \\ CFL\# = 1 \end{cases} . \tag{2.15}$$

Notice that $\tilde{L}$ does not appear in (2.15). This is equivalent to

$$\begin{cases} u^{(1)} = u^{(0)} + \Delta t L(u^{(0)}) \\ u^{(2)} = u^{(0)} + \frac{1}{2}\Delta t L(u^{(0)}) + \frac{1}{2}\Delta t L(u^{(1)}) \end{cases} \tag{2.16}$$

which is the classical Heun's method or modified Euler method [1].

ii) Third order case, $m = 2$.

For accuracy

$$\begin{cases} \beta_{32} = \frac{3\beta_{10}-2}{6P(\beta_{10}-P)} \\ \beta_{21} = \frac{1}{6\beta_{10}\beta_{32}} \\ \beta_{31} = \frac{\frac{1}{2}-\alpha_{32}\ \beta_{10}\ \beta_{21}-P\ \beta_{32}}{\beta_{10}} \\ \beta_{30} = 1 - \alpha_{31}\ \beta_{10} - \alpha_{32}\ P - \beta_{31} - \beta_{32} \\ \beta_{20} = P - \alpha_{21}\ \beta_{10} - \beta_{21} \end{cases} \tag{2.17}$$

α_{21}, α_{30}, α_{31}, β_{10} and $P = \beta_{20} + \alpha_{21}\ \beta_{10} + \beta_{21}$ are free parameters. The solution (2.17) is written in convenient inductive form.

Extensive searching leads to the following preferred scheme:

$$
(2.18)\qquad \begin{cases} u^{(1)} = u^{(0)} + \Delta t L(u^{(0)}) \\ u^{(2)} = \frac{3}{4}u^{(0)} + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}) \\ u^{(3)} = \frac{1}{3}u^{(0)} + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}) \\ CFL\# = 1 \end{cases}
$$

Notice that $\tilde{L}$ does not appear in (2.18). Also in computing $u^{(i)}$ we only need $L(u^{(i-1)})$, so there is no need to store the previous $L(u^{(k)})$, lowering the storage requirement significantly.

(2.18) is equivalent to

$$
(2.19)\qquad \begin{cases} u^{(1)} = u^{(0)} + \Delta t L(u^{(0)}) \\ u^{(2)} = u^{(0)} + \frac{1}{4}\Delta t L(u^{(0)}) + \frac{1}{4}\Delta t L(u^{(1)}) \\ u^{(3)} = u^{(0)} + \frac{1}{6}\Delta t L(u^{(0)}) + \frac{1}{6}\Delta t L(u^{(1)}) + \frac{2}{3}\Delta t L(u^{(2)}) \end{cases}
$$

We have been unable to identify (2.19) with any of the "classical" third order Runge-Kutta methods. On the other hand, the classical third order Runge-Kutta methods in [7], when written in equivalent forms (2.12), lead to negative β_{ik} and small CFL numbers (2.13), and are hence inferior to (2.19).

(iii) Fourth order case, $m = 3$

For accuracy we get a system of 7 equations with 16 unknowns, so there are 9 free parameters. Unfortunately this time the solution is not easily obtained in a convenient form. In [1] a general solution with two parameters is given for the form (2.10). We can certainly rewrite it in the form (2.12). Extensive searching seems to indicate that we cannot avoid negative constants β_{ik} this time. The classical fourth

order Runge-Kutta method can be written in the form (2.12) as

$$
(2.20)\quad \begin{cases} u^{(1)} = u^{(0)} + \frac{1}{2}\Delta t L(u^{(0)}) \\ u^{(2)} = \frac{1}{2}u^{(0)} - \frac{1}{4}\Delta t \tilde{L}(u^{(0)}) + \frac{1}{2}u^{(1)} + \frac{1}{2}\Delta t L(u^{(1)}) \\ u^{(3)} = \frac{1}{9}u^{(0)} - \frac{1}{9}\Delta t \tilde{L}(u^{(0)}) + \frac{2}{9}u^{(1)} - \frac{1}{3}\Delta t \tilde{L}(u^{(1)}) + \frac{2}{3}u^{(2)} + \Delta t L(u^{(2)}) \\ u^{(4)} = \frac{1}{3}u^{(1)} + \frac{1}{6}\Delta t L(u^{(1)}) + \frac{1}{3}u^{(2)} + \frac{1}{3}u^{(3)} + \frac{1}{6}\Delta t L(u^{(3)}) \\ \text{CFL\#} = \frac{2}{3} \end{cases}
$$

Notice that we have to compute $\tilde{L}(u^{(0)})$ and $\tilde{L}(u^{(1)})$. If we use the more awkward definition of $u^{(3)}$

(2.21)

$$
u^{(3)} = \frac{6431}{80000}\,u^{(0)} - \frac{18769}{160000}\Delta t\,\tilde{L}(u^{(0)}) + \frac{18769}{80000}u^{(1)} - \frac{137}{400}\Delta t\,\tilde{L}(u^{(1)}) + \frac{137}{200}u^{(2)} + \Delta t L(u^{(3)})
$$

then the CFL# can be raised slightly to $\frac{137}{200}$.

Another fourth order scheme with a slightly larger CFL # is:

$$
(2.22)\quad \begin{cases} u^{(1)} = u^{(0)} + \frac{1}{2}\Delta t L(u^{(0)}) \\ u^{(2)} = \frac{2}{5}u^{(0)} - \frac{2}{5}\Delta t \tilde{L}(u^{(0)}) + \frac{3}{5}u^{(1)} + \frac{3}{5}\Delta t L(u^{(1)}) \\ u^{(3)} = \frac{831}{20000}u^{(0)} - \frac{1769}{40000}\Delta t \tilde{L}(u^{(0)}) + \frac{4669}{20000}u^{(1)} - \frac{161}{600}\Delta t \tilde{L}(u^{(1)}) + \\ \qquad + \frac{29}{40}u^{(2)} + \frac{5}{6}\Delta t L(u^{(2)}) \\ u^{(4)} = \frac{1}{3}u^{(0)} + \frac{1}{3}u^{(1)} + \frac{1}{3}\Delta t L(u^{(1)}) + \frac{1}{3}u^{(3)} + \frac{1}{6}\Delta t L(u^{(3)}) \\ CFL\# = 0.87 \end{cases}
$$

We still need to compute $\tilde{L}(u^{(0)})$ and $\tilde{L}(u^{(1)})$.

(iv) Fifth order case, $m = 5$.

We simply write out the form (2.12) corresponding to a fifth order method

given on page 143 of [7]:

(2.23)

$$
\begin{cases}
u^{(1)} = u^{(0)} + \frac{1}{2}\Delta t L(u^{(0)}) \\
u^{(2)} = \frac{3}{4}u^{(0)} + \frac{1}{4}u^{(1)} + \frac{1}{8}\Delta t L(u^{(1)}) \\
u^{(3)} = \frac{3}{8}u^{(0)} - \frac{1}{8}\Delta t \tilde{L}(u^{(0)}) + \frac{1}{8}u^{(1)} - \frac{1}{16}\Delta t \tilde{L}(u^{(1)}) + \frac{1}{2}u^{(2)} + \frac{1}{2}\Delta t L(u^{(2)}) \\
u^{(4)} = \frac{1}{4}u^{(0)} - \frac{5}{64}\Delta t \tilde{L}(u^{(0)}) + \frac{1}{8}u^{(1)} - \frac{13}{64}\Delta t \tilde{L}(u^{(1)}) + \frac{1}{8}u^{(2)} + \frac{1}{8}\Delta t L(u^{(2)}) + \\
\qquad \frac{1}{2}u^{(3)} + \frac{9}{16}\Delta t L(u^{(3)}) \\
u^{(5)} = \frac{89537}{2880000}u^{(0)} + \frac{2276219}{40320000}\Delta t L(u^{(0)}) + \frac{407023}{2880000}u^{(1)} + \frac{407023}{672000}\Delta t L(u^{(1)}) + \\
\qquad \frac{1511}{12000}u^{(2)} + \frac{1511}{2800}\Delta t L(u^{(2)}) + \frac{87}{200}u^{(3)} - \frac{261}{140}\Delta t \tilde{L}(u^{(3)}) + \frac{4}{15}u^{(4)} + \frac{8}{7}\Delta t L(u^{(4)}) \\
u^{(6)} = \frac{4}{9}u^{(0)} + \frac{1}{15}u^{(1)} - \frac{8}{45}\Delta t \tilde{L}(u^{(1)}) + \frac{8}{45}u^{(3)} + \frac{2}{3}\Delta t L(u^{(3)}) + \frac{14}{45}u^{(5)} + \frac{7}{90}\Delta t L(u^{(5)}) \\
CFL\# = \frac{7}{30}
\end{cases}
$$

Notice that we need to compute $\tilde{L}(u^{(0)})$, $\tilde{L}(u^{(1)})$, $\tilde{L}(u^{(3)})$.

III. A Simplified Version of ENO Schemes. To use the Runge-Kutta type TVD time discretizations in Section II, we must have a spatial discrete operator (2.1) to start with. Theoretically one would like to use a TVD or TVB operator T satisfying (2.6), because then the full scheme (2.3) would be TVD or TVB. But as indicated in section I, the existing high order TVD or TVB schemes may smear discontinuities and pollute the solution (i.e. we may not get high order accuracy in a fairly large region near discontinuities), due to the fixed, wide stencil. The ENO schemes constructed in [5], [6], [4] are very promising experimentally and appealing conceptually, but the fact that they use cell-averages as well as point values via a reconstruction procedure, and that they were implemented using a Lax-Wendroff type time discretization, makes them rather complicated to program, especially in multi-dimensional problems, or problems with forcing terms. The Runge-Kutta type TVD time discretizations in Section II equipped with semi-discrete ENO schemes will simplify them in many cases (although there is no rigorous theory concerning TVB of semi-discrete ENO schemes or their Euler forward version, analysis in many cases and numerical experiments strongly support that the total variation increase at each

step is $O(\Delta x^r)$ for a r-th order ENO scheme, see [6]. Hence the full scheme (2.3) in this case should also be TVB). In this section we further simplify non-oscillatory methods by deriving a version of ENO schemes using only fluxes, not cell-averages.

We start with a simple first order monotone Lax-Friedrichs type of scheme. If we define

$$f^+(u) = \frac{1}{2}(f(u) + \alpha u), \quad f^-(u) = \frac{1}{2}(f(u) - \alpha u) \tag{3.1}$$

where $\alpha \geq \max|f'(u)|$ is a constant, then clearly

$$f^{+\prime}(u) \geq 0, \quad f^{-\prime}(u) \leq 0 \tag{3.2}$$

$$f^+(u) + f^-(u) = f(u) \tag{3.3}$$

The Lax-Friedrichs scheme is simply (1.3) with the numerical flux defined by

$$\hat{f}_{j+\frac{1}{2}} = f_j^+ + f_{j+1}^- \tag{3.4}$$

Taylor expansion reveals the existence of constants $a_2,\ a_4, \ldots, a_{2m-2}, \ldots$, such that if

$$\hat{f}_{j+\frac{1}{2}} = f_{j+\frac{1}{2}} + \sum_{k=1}^{m-1} a_{2k} \Delta x^{2k} \left(\frac{\partial^{2k}}{\partial x^{2k}} f\right)_{j+\frac{1}{2}} + O(\Delta x^{2m+1}) \tag{3.5}$$

then the scheme (1.3) will be $2m$-th order accurate in space in the sense of (2.2).

For example, $a_2 = -\frac{1}{24}, \quad a_4 = \frac{7}{5760}, \ldots$.

In light of (3.4), it is natural to require

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}^+_{j+\frac{1}{2}} + \hat{f}^-_{j+\frac{1}{2}} \tag{3.6}$$

and to define the positive flux $\hat{f}^+_{j+\frac{1}{2}}$ and the negative flux $\hat{f}^-_{j+\frac{1}{2}}$ (in the meaning of (3.2)) separately.

For accuracy, we require $\hat{f}^{+}_{j+\frac{1}{2}}$ and $\hat{f}^{-}_{j+\frac{1}{2}}$ to satisfy (3.5) separately:

$$\hat{f}^{\pm}_{j+\frac{1}{2}} = f^{\pm}_{j+\frac{1}{2}} + \sum_{k=1}^{m-1} a_{2k}\Delta x^{2k}\left(\frac{\partial^{2k}}{\partial x^{2k}} f^{\pm}\right)_{j+\frac{1}{2}} + O(\Delta x^{2m+1}) \tag{3.7}$$

We achieve (3.7) by using polynomial interpolants $p^{\pm}_{j}$ of $f^{\pm}$ to the correct order:

$$p^{\pm}_{j}(x) = f^{\pm}(u(x)) + O(\Delta x^{2m+1}) \tag{3.8}$$

near $x = x_{j+\frac{1}{2}}$, then define

$$\hat{f}^{\pm}_{j+\frac{1}{2}} = p^{\pm}_{j}(x_{j+\frac{1}{2}}) + \sum_{k=1}^{m-1} a_{2k}(\Delta x)^{2k}\left(\frac{\partial^{2k}}{\partial x^{2k}} p^{\pm}_{j}\right)_{x=x_{j+\frac{1}{2}}} \tag{3.9}$$

Clearly if (3.8) is true, then the fluxes $\hat{f}^{\pm}_{j+\frac{1}{2}}$ defined by (3.9) will satisfy (3.7).

It is in constructing the interpolating polynomials $p^{\pm}_{j}(x)$ that we use the ENO moving stencil idea: in order to achieve (3.8) $p^{\pm}_{j}(x)$ can be polynomials of degree $2m$ interpolating $f^{\pm}(u(x))$ at *any* $2m+1$ points near $x_{j+\frac{1}{2}}$. We use the ENO ideas in [5], [6] and [4] to choose the $2m+1$ points automatically from the smoothest possible region, but *start* with the correct one according to (3.4).

The algorithm can be written as follows: For constructing $p^{+}_{j}(x)$:

(3.10) (1) $k^{(0)}_{\min} = k^{(0)}_{\max} = j, \quad Q^{(0)}_{+}(x) = f^{+}(u_j)$

(2) Inductively, assume we have $k^{(n-1)}_{\min}$, $k^{(n-1)}_{\max}$ and $Q^{(n-1)}_{+}(x)$, then we compute the n-th divided differences of $f^{+}(u(x))$:

$$a^{(n)} = f^{+}[u(x_{k^{(n-1)}_{\min}}), \ldots, u(x_{k^{(n-1)}_{\max}+1})] \tag{3.11a}$$

$$b^{(n)} = f^{+}[u(x_{k^{(n-1)}_{\min}-1}), \ldots, u(x_{k^{(n-1)}_{\max}})]. \tag{3.11b}$$

We proceed to add a point to the stencil according to the smaller n-th divided difference:

(i) If $|a^{(n)}| \geq |b^{(n)}|$, then

$$c^{(n)} = b^{(n)} \tag{3.12a}$$

$$k_{\min}^{(n)} = k_{\min}^{(n-1)} - 1, \quad k_{\max}^{(n)} = k_{\max}^{(n-1)} \tag{3.12b}$$

(ii) If $|a^{(n)}| < |b^{(n)}|$, then

$$c^{(n)} = a^{(n)} \tag{3.13a}$$

$$k_{\min}^{(n)} = k_{\min}^{(n-1)}, \quad k_{\max}^{(n)} = k_{\max}^{(n-1)} + 1 \tag{3.13b}$$

and finally

$$Q_+^{(n)}(x) = Q_+^{(n-1)}(x) + c^{(n)} \Pi_{k=k_{\min}^{(n-1)}}^{k_{\max}^{(n-1)}} (x - x_k) \tag{3.14}$$

(3) $p_j^+(x) = Q_+^{(2m)}(x)$

For constructing $p_j^-(x)$:

(1) $k_{\min}^{(0)} = k_{\max}^{(0)} = j + 1, \quad Q_-^{(0)}(x) = f^-(u_{j+1})$

(3.15) (2) same as (2) above with f^+ replaced by f^- and Q_+ replaced by Q_-;

(3) $p_j^-(x) = Q_-^{(2m)}(x)$

REMARK (3.1):

(a) For the first order scheme we just get back the Lax-Friedrichs flux (3.4);

(b) The second order scheme here is similar to the usual minmod second order TVD scheme, [9] except that the minmod function is replaced by choosing the value closer to zero (we omit the details of the derivation here); this is still a TVD method.

(c) For the linear equation $u_t + au_x = 0$ (the scheme is still nonlinear!), the schemes here are equivalent to the ENO schemes in [6] using the primitive function reconstruction, except for a possible difference in the choice of stencil. The ENO schemes in [6] choose the stencil according to the divided difference table of cell averages of u, rather than that of $f^{\pm}(u)$ here. We again omit the details of derivation here. Since the ENO schemes in [6] worked so well numerically, and our simplified schemes here are equivalent to those in [6] for linear equations, we expect ours to work as well. For preliminary numerical results see section IV.

As mentioned before there is at present no rigorous theory about TVB of this type of ENO schemes. Here we make two observations for our schemes along these lines:

(1) For smooth solutions all the divided differences (3.11) should be bounded (by the maximum norm of the n-th derivative of $f^{\pm}$, times some constant). So if we use

$$\tilde{c}^{(n)} = \min(|c^{(n)}|,\ M^{(n)})\ \text{sign}\ (c^{(n)}) \tag{3.16a}$$

or

$$\tilde{c}^{(n)} = \min(|c^{(n)}|,\ M^{(n)}\Delta x^{n-2})\ \text{sign}\ (c^{(n)}) \tag{3.16b}$$

in the place of $c^{(n)}$ in (3.14) for $n \geq 2$, where $M^{(n)}$ are constants which are related to the maximum norm of the n-th derivative of $f^{\pm}$ in initial smooth regions, it does not affect the accuracy in regions of smoothness. We then get a TVB scheme. We

can easily see that our flux with (3.16) satisfies

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}^{\mathrm{TVD2}}_{j+\frac{1}{2}} + \hat{c}_{j+\frac{1}{2}} \tag{3.17a}$$

where

$$|\hat{c}_{j+\frac{1}{2}}| \leq M\Delta x^2 \tag{3.17b}$$

with the constant M depending on the $M^{(n)}$, and $\hat{f}^{\mathrm{TVD2}}_{j+\frac{1}{2}}$ is the second order TVD flux mentioned in Remark (3.1b) above. Now (3.17) clearly implies TVB of the scheme. Numerically we do not see any essential difference by using or not using (3.16), hence we strongly believe that ENO schemes without (3.16) are also TVB, at least for most practical problems.

(2) The approach in (3.1) - (3.15) is not the only possible one which can be used to construct an ENO scheme based on interpolating fluxes. At an early stage of our current work we used another approach: starting from f^+ and f^- in (3.1), then for constructing $p^+_j(x)$:

(1) $k^{(1)}_{\min} = j-1,\ k^{(1)}_{\max} = j,\ Q^{(1)}_+(x) = f^+(u_j) + f^+[u(x_{j-1}), u(x_j)](x - x_j)$;

(2) & (3), same as the procedures (2) and (3) above in (3.10),

Similarly, for constructing $p^-_j(x)$:

(1) $k^{(1)}_{\min} = j,\ k^{(1)}_{\max} = j+1,\ Q^{(1)}_-(x) = f^-(u_j) + f^-[u(x_j), (x_{j+1})](x - x_j)$;

(2) and (3), same as the procedures (2) and (3) above in (3.15).

Then we take $p_j(x) = p^+_j(x) + p^-_j(x)$, and write our scheme as

$$u^{n+1}_j = u^n_j - \Delta t(\frac{d}{dx}p_j(x))_{x=x_j} \tag{3.18}$$

Notice that the scheme (3.18) is simpler than (3.9) - (1.3). The only trouble is that it is *not* in conservation form (1.3). However, if we use (3.16), then it is easily seen that (3.18) can be written as

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}^{LF}_{j+\frac{1}{2}} - \hat{f}^{LF}_{j-\frac{1}{2}}) + \Delta x^2 \hat{c}_j \tag{3.19}$$

where $|\hat{c}_j| \leq M$, and $\hat{f}^{LF}_{j+\frac{1}{2}}$ is the first order Lax-Friedrichs flux (3.4). We can call such schemes "essentially conservative" because the most important property of a conservative scheme–the conclusion of the Lax-Wendroff theorem in [8] – is still valid. Since the scheme deviates from a first order monotone scheme (not only a TVD scheme) by $M\Delta x^2$, we have even a stronger theory than before – we have the entropy condition, hence full convergence (not just of a subsequence), and also convergence in multi-dimensional scalar problems i.e. we have every convergence property first order monotone schemes have. Unfortunately numerical experiments indicate that in some cases, (3.18) is inferior to the fully conservative (3.9) - (1.3). An illustrative example is to compute the Riemann problem for Burgers' equation $u_t + uu_x = 0$ with a moving shock (e.g. $u_{\text{left}} = \frac{1}{2}$, $u_{\text{right}} = -1.$) using the fifth order versions of (3.18) and (3.9) - (3.15), (1.3), equipped with a fifth order multi-level TVD time discretization in [13]. The procedure (3.9) - (3.15), (1.3) gives good results, with or without (3.16); while without (3.16), the non conservative (3.18) gives the wrong shock location. With (3.16) the shock location becomes correct, but the mechanism that enforces this causes a rather severe smearing of the shock. For these reasons we abandoned the simple and theoretically pleasing version (3.18).

Finally let us point out that the Lax-Friedrichs building block is only a convenient one; we may also use other monotone or E-fluxes (see, e.g. [10]) as our building blocks. Of course it is not always possible to assoicate f^+ and f^- as in (3.1) with each E-flux such that (3.2), (3.3), (3.4) is valid, but careful inspection

reveals that we do not need to use the values of f^+ and f^- – only their divided differences. For each E-flux $h_{j+\frac{1}{2}}$ we can define

$$df^+_{j+\frac{1}{2}} = f_{j+1} - h_{j+\frac{1}{2}}, \quad df^-_{j+\frac{1}{2}} = h_{j+\frac{1}{2}} - f_j \tag{3.20}$$

where $df^+_{j+\frac{1}{2}}$ and $df^-_{j+\frac{1}{2}}$ replace the first (undivided) differences $f^+_{j+1} - f^+_j$ and $f^-_{j+1} - f^-_j$. Hence we can just use the divided difference tables of df^+ and df^- in place of the divided difference tables of f^+ and f^- in constructing p^+_j and p^-_j, and define f^+_j, f^-_j in any consistent way such that $f^+_j + f^-_j = f_j$, e.g. $f^+_j = f_j$, $f^-_j = 0$. By (3.20)

$$df^+_{j+\frac{1}{2}} + df^-_{j+\frac{1}{2}} = f_{j+1} - f_j \tag{3.21}$$

hence if p^+_j and p^-_j use the *same* stencil then $p^+_j(x) + p^-_j(x)$ is a polynomial interpolating $f(u(x))$, thus accuracy is guaranteed with (3.9) - (3.6); if the stencils are different, say $\tilde{p}^+_j$ has the same stencil as p^-_j but p^+_j does not, then it is easy to show that $\tilde{p}^+_j - p^+_j$ is a sum of r-th order undivided differences of $df^+_{j+\frac{1}{2}}$ (r is the order of the polynomials p^+_j, p^-_j) hence as long as these are $O(\Delta x^{r+1})$ (valid if the E-flux $h_{j+\frac{1}{2}}$ is smooth up to order r) we still have the correct accuracy.

The reason one might consider general E-fluxes as building blocks is that the Lax-Friedrichs flux is considered to be too dissipative. While the first order Lax-Friedrichs scheme is much inferior to upwind schemes (e.g. to Godunov's or the Engquist-Osher schemes), our numerical experiments show that higher order ENO schemes using Lax-Friedrichs building blocks work quite well (although they are still slightly inferior to the same order ENO schemes based on upwind building blocks, the difference is much smaller than that in the first order case). The advantage of the Lax-Friedrichs flux is that it is C^∞, hence the ENO schemes based on it have full high order accuracy. On the other hand, most other E-fluxes – Godunov's,

Engquist-Osher's, Entropy condition satisfying version of Roe's, etc. – are *not* smooth at sonic points (points at which $f'(u) = 0$), hence ENO schemes based on them using the methodology of this paper will lose accuracy at sonic points. Although we may overcome this by smoothing those E-fluxes at sonic points, in most cases the simple Lax-Friedrichs building block should be good enough.

Problems in multi-dimensions are approximated by applying the procedure described in (3.1) - (3.15) or its generalization (3.20), (3.21) to each of the terms $\frac{\partial f_i}{\partial x_i}$ in (1.1a). The Range-Kutta methods devised in section 2 are then used, with CFL numbers shrunk by a factor $(d)^{-1}$.

Systems of equations are approximated using by now familiar field-by-field decompositions ideas. In (3.1) we replace the scalar constant α by a constant matrix αI, $\alpha = \{\alpha_{ij}\}_{ij=1}^{m}$, where the eigenvalues of $\frac{\partial f^+}{\partial u}$ are non-negative and of $\frac{\partial f^-}{\partial u}$ are non-positive.

Obviously α might be taken to be a sufficiently large positive scalar multiple of the identity, but this might also lead to some smearing of discontinuities associated with slower waves. Other more practical choices might involve freezing $\frac{\partial f}{\partial u}$ at some constant state $\bar{u}$, diagonalizing:

$$\frac{\partial f}{\partial \bar{u}} = T(\bar{u}) \begin{pmatrix} \lambda_1(\bar{u}) & & \\ & \ddots & \\ & & \lambda_d(\bar{u}) \end{pmatrix} T^{-1}(\bar{u})$$

and letting

$$\alpha = T(\bar{u}) \begin{pmatrix} \bar{\lambda}_1 & & \\ & \ddots & \\ & & \bar{\lambda}_d \end{pmatrix} T^{-1}(\bar{u})$$

where each $\bar{\lambda}_i \geq |\lambda_i(u)|$ throughout the region. To be safe, the margin of difference between each $\bar{\lambda}_i$ and the maximum value of $|\lambda_i(u)|$ has to be sufficiently large.

In any case, the corresponding $p_j^{\pm}(x)$ are obtained with the help of the left and right eigenvectors of $\frac{\partial f}{\partial u}(u_j)$, which we denote by $\ell_j^{(i)}$ and $r_j^{(i)}$, $i = 1, ...d$. We interpolate $\ell_j^{(i)} \cdot f^{\pm}(u)$ obtaining $\ell_j^{(i)} \cdot p_j^{\pm}(x)$ exactly as in (3.10) - (3.15). We then define

$$p^{\pm}_{j_{(s)}} = \sum_{i-1}^{d} (\ell_j^{(i)} \cdot p_j^{\pm}) r_j^{(i)}$$

The fluxes $\hat{f}^{\pm}_{j+\frac{1}{2}}$ are defined through (3.9).

Generalizations of the type described in (3.20), (3.21) using approximate Riemann solvers for $h_{j+\frac{1}{2}}$, appropriately smoothed at sonic points, may also be obtained.

Work is currently under way with various colleagues applying these methods to Euler's equations of compressible gas dynamics in multi-space dimensions.

IV. Preliminary Numerical Results. The numbers in this section are often written in exponential form, e.g. 4.2-3 means 4.2×10^{-3}.

EXAMPLE 1. The ENO schemes (3.1) - (3.15) in section III combined with the Runge-Kutta type TVD time discretizations (2.19) - (2.20) in section II are used to solve the nonlinear Burgers equation with periodic initial conditions:

$$\text{(4.1)} \qquad \begin{cases} u_t + \left(\frac{u^2}{2}\right)_x = 0 \\ u(x,0) = \frac{1}{4} + \frac{1}{2}\sin \pi x \end{cases} \qquad -1 \le x \le 1$$

The exact solution is smooth up to $t = \frac{2}{\pi}$, then it develops a moving shock which interacts with the rarefaction waves. We get the exact solution by using a Newton iteration. For details, see [6].

Since there is a sonic point, we use the smooth LF (Lax-Friedrichs) building block in our ENO schemes. Both 3-3-LF-ENO (third order in time *and* space ENO

schemes with Lax-Friedrichs building blocks) and 4-4-LF-ENO are used.

We use a CFL number of 0.8 for 3-3-LF-ENO and 0.6 for 4-4-LF-ENO.

The errors of the numerical solutions at t = 0.3 are listed in Table 1. Since the exact solution is still smooth, we get the full order of accuracy in both L_1 and L_∞ norms.

At $t = \frac{2}{\pi}$, the shock begins to form. We use $\Delta x = \frac{1}{40}$ and print out the errors at 10 points near the shock:

3-3-LF-ENO: -8.7-4, -2.6-3, -7.1-3, -1.3-2, -1.2-1, *

8.2-2, 8.0-3, 1.1-3, 2.6-4, 1.9-4

4-4-LF-ENO: -1.5-4, -3.2-4, -1.6-3, -1.5-2, -1.2-1, *

7.8-2, 7.9-3, 9.2-4, 1.8-4, 9.5-5

where the * is the position of the shock.

We see that there is a very good shock transition. (No oscillations are observed). Figures 1-6 show the shock transitions.

In smooth regions the numerical solutions are very accurate. We compute the L_1 and L_∞ norms in the region a distance of 0.1 from the shock (i.e. $|x-$ shock location$| \geq$ 0.1) and list them in Table 2. From the table we can see that the errors are of the same magnitude as in the smooth case when $t = 0.3$.

At $t = 1.1$, the reaction between the shock and the rarefaction waves is over. The solution becomes monotone between the shocks. We again print out the errors at 10 points near the shock for $\Delta x = \frac{1}{40}$:

3-3-LF-ENO: -1.0-4, 4.6-4, 4.2-4, -3.3-2, -8.3-3, *

3.9-2, 1.7-3, -1.6-4, -2,5-5, -7.0-6.

4-4-LF-ENO: 1.1-5, 6.6-4, -1.6-3, -5.7-2, -1.3-3, *

5.7-2, 2.7-3, -2.4-4, -5.8-5, -1.2-6

Figures 7-12 show the shock transitions.

The errors where the solution is smooth are again listed in Table 2.

We can see the excellent behavior of ENO schemes in this example.

EXAMPLE 2. A two-dimensional version of Example 1

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \\ u(x,y,0) = \frac{1}{4} + \frac{1}{2}\sin\pi\left(\frac{x+y}{2}\right) \end{cases} \qquad -2 \le x,\ y \le 2 \tag{4.2}$$

is tested using the same schemes as in Example 1 in a dimension by dimension fashion (i.e. $T(u) = (I + \Delta t L_x + \Delta t L_y)(u)$ in (2.1), together with the $R-K$ time discretization: (2.19)-(2.20)). The exact solution is one-dimensional depending only on $\xi = x + y$, however our grid points are rectangular in (x, y) coordinates, and thus this example is a truly 2-dimensional test problem.

The CFL number is always taken to be half of the one dimensional analog, i.e. 0.4 for the 3-3-LF-ENO and 0.3 for the 4-4-LF-ENO.

As in Example 1, we collect the L_1 and L_∞ errors at $t = 0.3$ (smooth solution) in Table 3 and the L_1 and L_∞ errors in regions at a distance of 0.1 from the shock at times $t = \frac{2}{\pi}$ and $t = 1.1$ in Table 4. We also print out 10 points near the shock when $x = 0$, $t = \frac{2}{\pi}$ and $t = 1.1$ for $\Delta x = \Delta y = \frac{1}{20}$.

$t = \frac{2}{\pi}$, $x = 0$:

3-3-LF-ENO: -9.7-4, -2.3-3, -7.6-3, -4.5-3, -1.2-1, *

8.2-2, 7.9-3, 1.1-3, 2.6-4, 1.9-4

4-4-LF-ENO: -1.5-4, -3.2-4, -1.6-3, -1.5-2, -1.2-1, *

7.8-2, 7.9-3, 9.2-4, 1.8-4, 9.5-5

$t = 1.1,\ x = 0$:

3-3-LF-ENO: -1.0-4, 4.6-4, 4.2-4, -3.3-2, -8.3-3, *

3.9-2, 1.7-3, -1.6-4, -2.5-5, -7.0-6

4-4-LF-ENO: 1.1-5, 6.6-4, -1.6-3, -5.7-2, -1.3-3, *

5.7-2, 2.7-3, -2.4-4, -5.8-5, -1.2-6

The shock transition graphs are very similar to Figures 1-12, hence we omit them.

We observe essentially the same results as in the 1-dim Example 1. This indicates that our ENO schemes work well in multi-dimensional problems.

EXAMPLE 3. We use the same schemes as in Example 2 above to solve a linear problem

$$\begin{cases} u_t + u_x + u_y = 0, & -1 \le x, y \le 1 \\ u(x, y, 0) = \begin{matrix} 1, \text{ if } (x,y)\in S \\ 0, \text{ if } (x,y)\notin S \end{matrix} \end{cases} \tag{4.3}$$

where $S = \{(x, y) : |x - y| < \frac{1}{\sqrt{2}},\ |x + y| < \frac{1}{\sqrt{2}}\}$ is a unit square centered at the origin and rotated by an angle of $\frac{\pi}{4}$ (see [4]). We use $\Delta x = \frac{1}{10}$ and run the scheme up to $t = 16$ (8 periods in time), in order to study the stability and the amount of smearing of discontinuities of these methods.

The numerical solutions at $t = 2$ (after 1 period in time) and at $t = 16$; $y = 0$ and $y = -0.4$, are displayed in Figures 13-20.

Observations: (1) the 2 dimensional schemes are stable under CFL numbers one half of those used for 1 dimension; (further experiments using 1-dimensional CFL numbers led to instability–overflow),

(2) The 4-th order scheme resolves the discontinuities better than the 3rd order method.

(3) Overshoots and undershoots, if any, are negligible.

EXAMPLE 4. We did not prove for these ENO methods that limit solutions satisfy the entropy condition. However, numerical experiments in [6], including some tests using nonconvex fluxes, indicated the convergence of ENO schemes to the correct entropy solution. We test our schemes 3-3-LF-ENO and 4-4-LF-ENO for Riemann problems for two such fluxes. One is

$$f(u) = \frac{1}{4}(u^2 - 1)(u^2 - 4) \tag{(4.4)}$$

with $u_L = 2$, $u_R = -2$ (the exact solution is a shock followed by a rarefaction wave followed by another shock) and with $u_L = -3$, $u_R = 3$ (a stationary shock at $x = 0$). See [6] for details. Our schemes converge to the correct solutions in both cases with good resolution. The results are displayed in Figures 21-32.

Another nonconvex flux we test is the well known Buckley-Leverett example:

$$f(u) = \frac{4u^2}{4u^2 + (1-u)^2} \tag{4.5}$$

with initial data $u = 1$ in $[-\frac{1}{2}, 0]$, and $u = 0$ elsewhere.

The exact solution is a shock-rarefaction-contact discontinuity mixture. Our schemes resolve the correct solution well. However, the 3-3-LF-ENO (using the Lax-Friedrichs building block) smears more than the 3-3-EO-ENO (using the Engquist-Osher building block) (Figures 33-38). Although in this example $f'(u) \geq 0$ so

there was no need to smooth the flux, the observed improvement using the upwind building block indicate that sometimes it is worthwhile spending the effort to smooth the EO or other upwind flux rather than to use the simple LF building block.

Table 1. (Example 1).

$t = 0.3$; E : type of error; r : numerical order

$\Delta x \backslash^{E}$	L_∞				L_1			
	3-3-ENO	r	4-4-ENO	r	3-3-ENO	r	4-4-ENO	r
$\frac{1}{10}$	2.6-3		2.1-3		7.2-4		5.3-4	
$\frac{1}{20}$	2.3-4	3.50	1.1-4	4.25	5.7-5	3.66	2.5-5	4.41
$\frac{1}{40}$	3.1-5	2.89	5.4-5	4.35	6.5-6	3.13	9.2-7	4.76

Table 2. (Example 1).

Errors in smooth region $|x-$ shock$| \geq 0.1$; $\Delta x = \frac{1}{40}$;

L_∞				L_1			
$t = 2/\pi$		$t = 1.1$		$t = 2/\pi$		$t = 1.1$	
3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO
8.7-4	1.5-4	1.6-4	1.6-5	3.5-5	8.6-6	6.7-6	1.4-6

Table 3. (Example 2).

$t = 0.3$; E : type of error; r : numerical order

$\Delta x = \Delta y \backslash^{E}$	L_∞				L_1			
	3-3-ENO	r	4-4-ENO	r	3-3-ENO	r	4-4-ENO	r
$\frac{1}{5}$	2.7-3		2.1-3		1.4-3		2.7-4	
$\frac{1}{10}$	2.3-4	3.55	1.1-4	4.21	1.1-4	3.67	1.3-5	4.40
$\frac{1}{20}$	3.2-5	2.85	5.6-6	4.36	1.3-5	3.08	4.5-7	4.80

Table 4. (Example 2).

Errors in smooth region $|(x, y) -$ shock$| \geq 0.1$; $\Delta x = \Delta y = \frac{1}{20}$;

L_∞				L_1			
$t = 2/\pi$		$t = 1.1$		$t = 2/\pi$		$t = 1.1$	
3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO	3-3-ENO	4-4-ENO
9.9-4	1.5-4	1.6-4	1.7-5	7.9-5	4.8-6	1.5-5	7.7-7

In the following figures, the solid lines are for exact solutions, and the circles are for numerical solutions.

Figures 1-12 are for Burgers' equation (4.1);

Figures 13-20 are for the linear two dimensional equation with discontinuous initial data (4.3), with $\Delta x = \frac{1}{10}$;

Figures 21-32 are for Riemann problems for the nonconvex flux (4.4). Figures 21-26 correspond to $u_{\text{left}} = 2$, $u_{\text{right}} = -2$; Figures 27-32 correspond to $u_{\text{left}} = -3$, $u_{\text{right}} = 3$;

Figures 33-38 are for Riemann problems for the nonconvex flux (4.5).

References

[1] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations; Prentice-Hall, 1971.

[2] A. Harten, High Resolution Schemes for Hyperbolic Conservation Laws; J. Comput. Phys., V49, 1983, pp. 357-393.

[3] A. Harten, On a Class of High Resolution Total-Variation-Stable Finite Difference Schemes; SIAM J. Numer. Anal., V21, 1984, pp. 1-23.

[4] A. Harten, Preliminary Results on the Extension of ENO Schemes to Two-Dimensional Problems; Proceedings of the International Conference on Hyperbolic Problems, Saint-Etienne, January 1986.

[5] A. Harten and S. Osher, Uniformly High Order Accurate Non-Oscillatory Schemes, I; MRC Technical Summary Report #2823, May 1985, to appear in SIAM J. Numer. Anal.

[6] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III; J. Comput. Phys., to appear.

[7] J.D. Lambert, Computational Methods in Ordinary Differential Equations; John Wiley & Sons Ltd., 1973.

[8] P.D. Lax and B. Wendroff, Systems of Conservation Laws; Comm. Pure Appl. Math., V13, 1960, pp. 217-237.

[9] S. Osher and S. Chakravarthy, High Resolution Schemes and the Entropy Condition; SIAM J. Numer. Anal., V21, 1984, pp. 955-984.

[10] S. Osher and S. Chakravarthy, Very High Order Accurate TVD Schemes; IMA Volumes in Mathematics and its Applications, V2, Springer-Verlag, 1986, pp. 229-274.

[11] P. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes; J. Comput. Phys. V43, 1981, pp. 357-372.

[12]] C. Shu, TVB Uniformly High Order Schemes for Conservation Laws; Math. Comp., July 1987, (to appear).

[13] C. Shu, TVD Time Discretizations; Preprint

[14] P.K. Sweby, High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws; SIAM J. Numer. Anal. V21, 1984, pp. 995-1011.

[15] B. Van Leer, Towards the Ultimate Conservative Difference Scheme, II. Monotonicity and Conservation Combined in a Second Order Scheme; J. Comp. Phys. Vl4, (1974) pp. 361-376.

[16] B. Van Leer, Towards the Ultimate Conservation Difference Scheme, V. A Second-Order Sequel to Godunov's Method; J. Comp. Phys., V32, (1979) pp. 101-136.

[17] R. Sanders, A Third Order Accurate Variation Nonexpansive Difference Scheme for Single Nonlinear Conservation Laws; Math Comp., to appear, (1988).

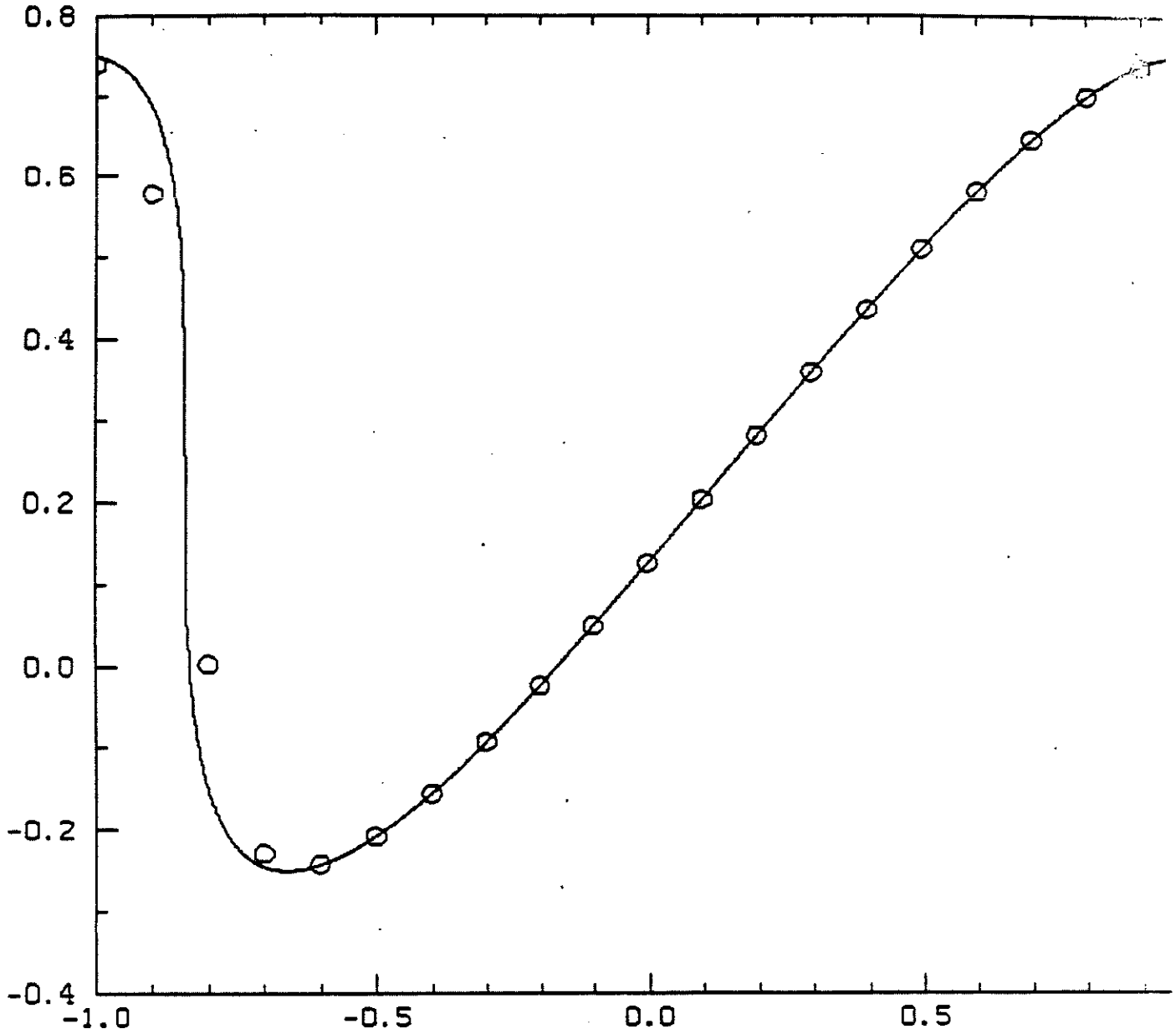

Figure 1 : 3-3-LF-ENO, $\Delta x = \frac{1}{10}$, $t = \frac{2}{\pi}$

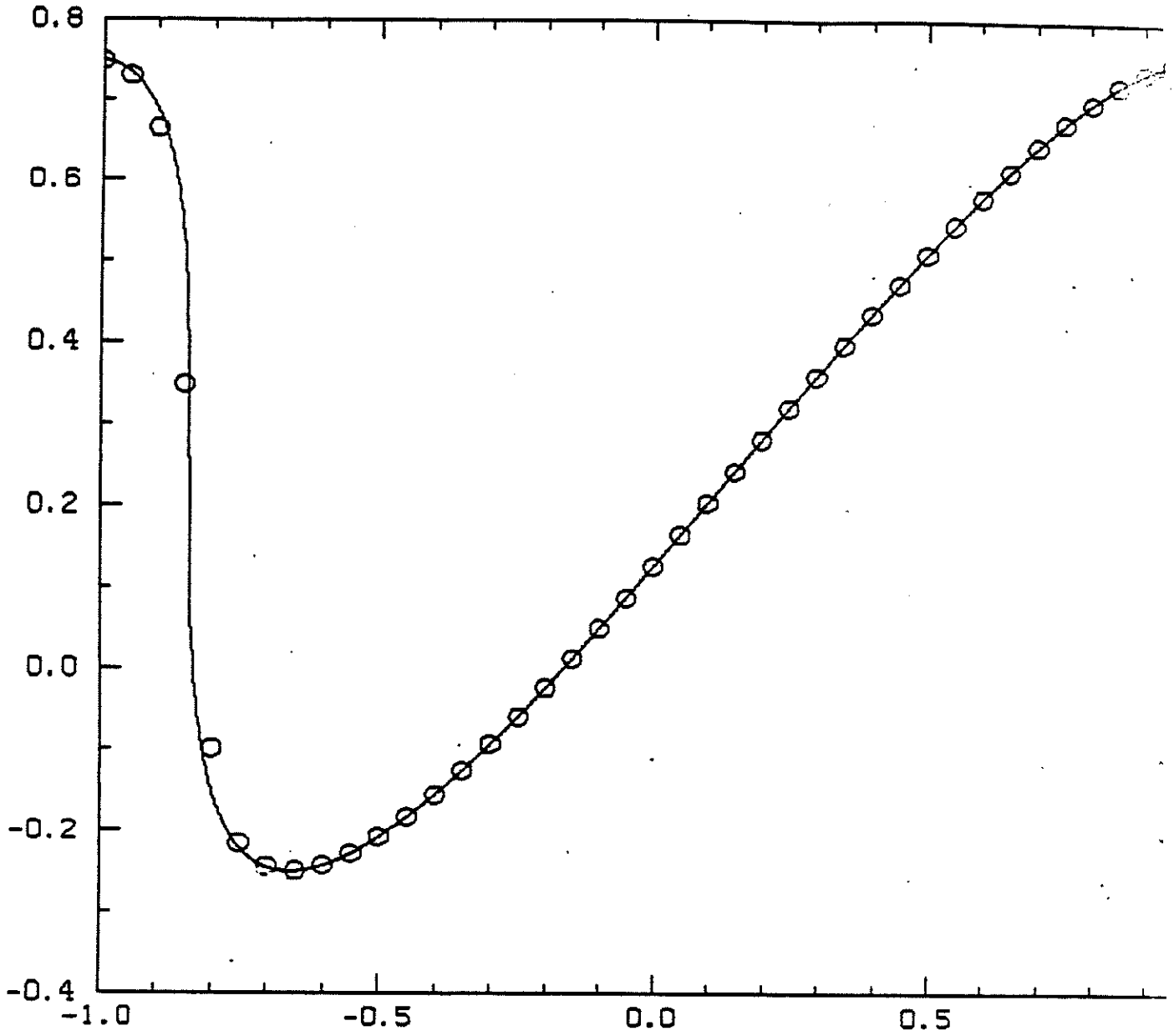

Figure 2 : 3-3-LF-ENO , $\Delta x = \frac{1}{20}$, $t = \frac{2}{\pi}$

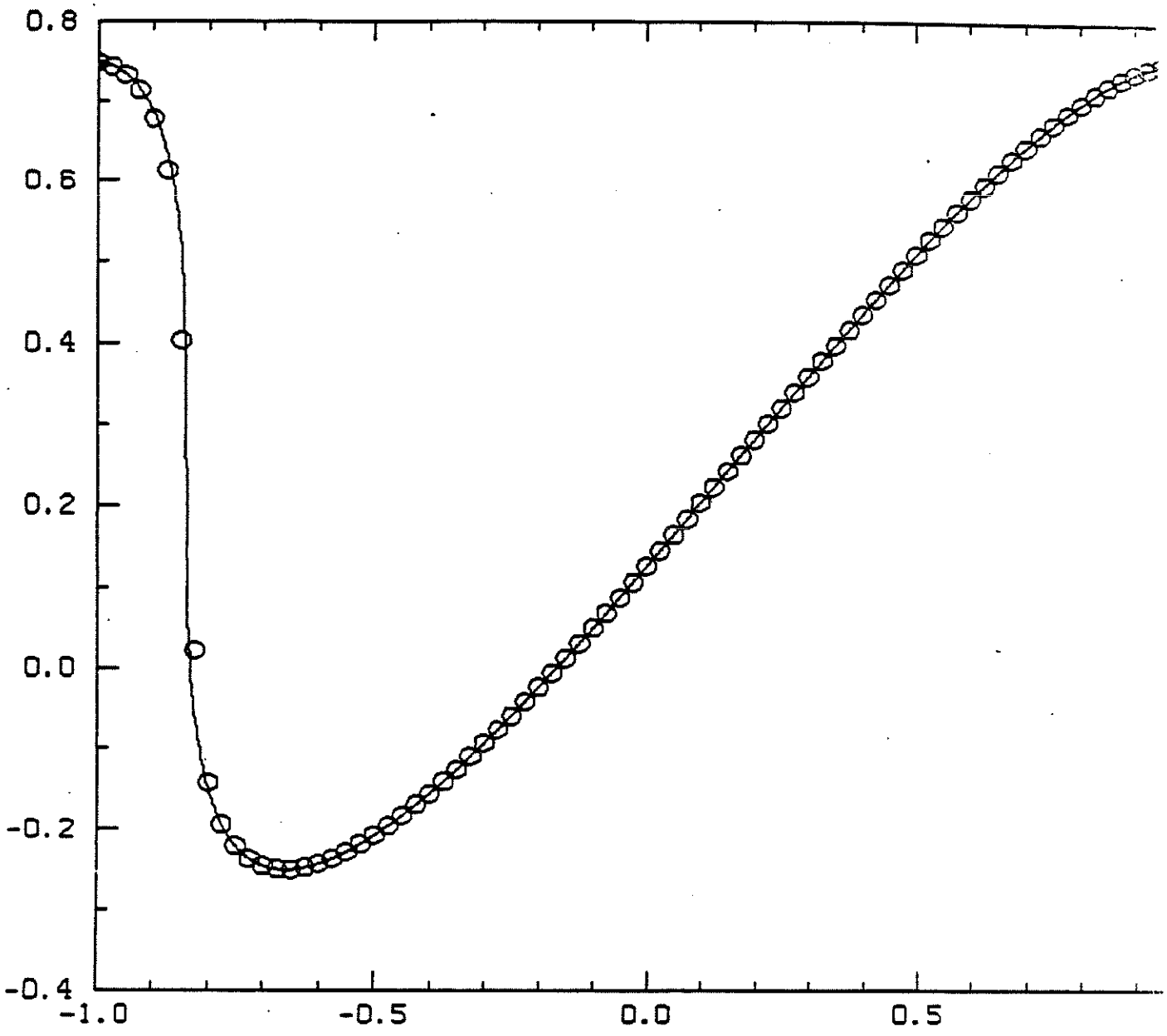

Figure 3 : 3-3-LF-ENO, $\Delta x = \frac{1}{40}$, $t = \frac{2}{\pi}$

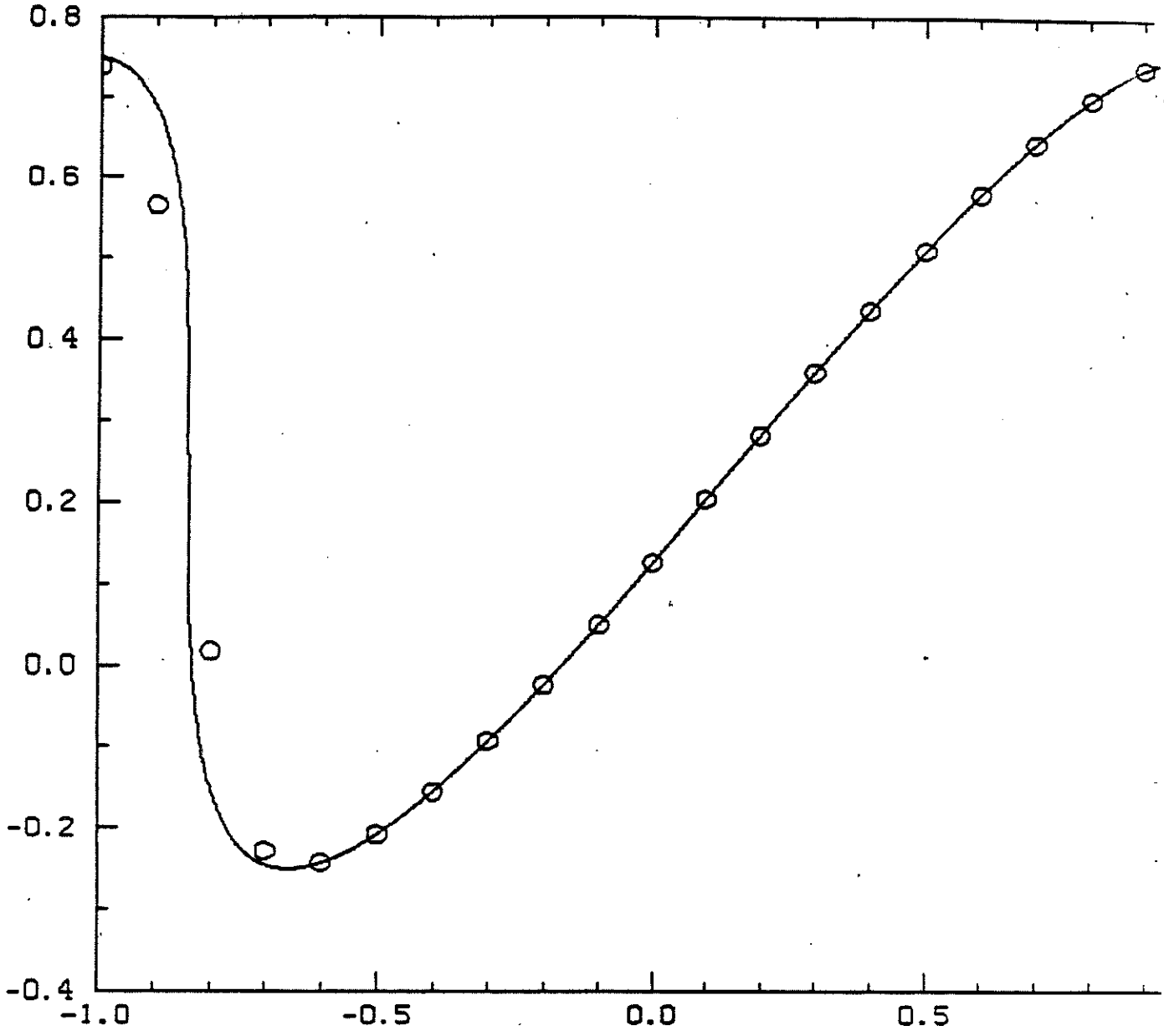

Figure 4 : 4-4-LF-ENO, $\Delta x = \frac{1}{10}$, $t = \frac{2}{\pi}$

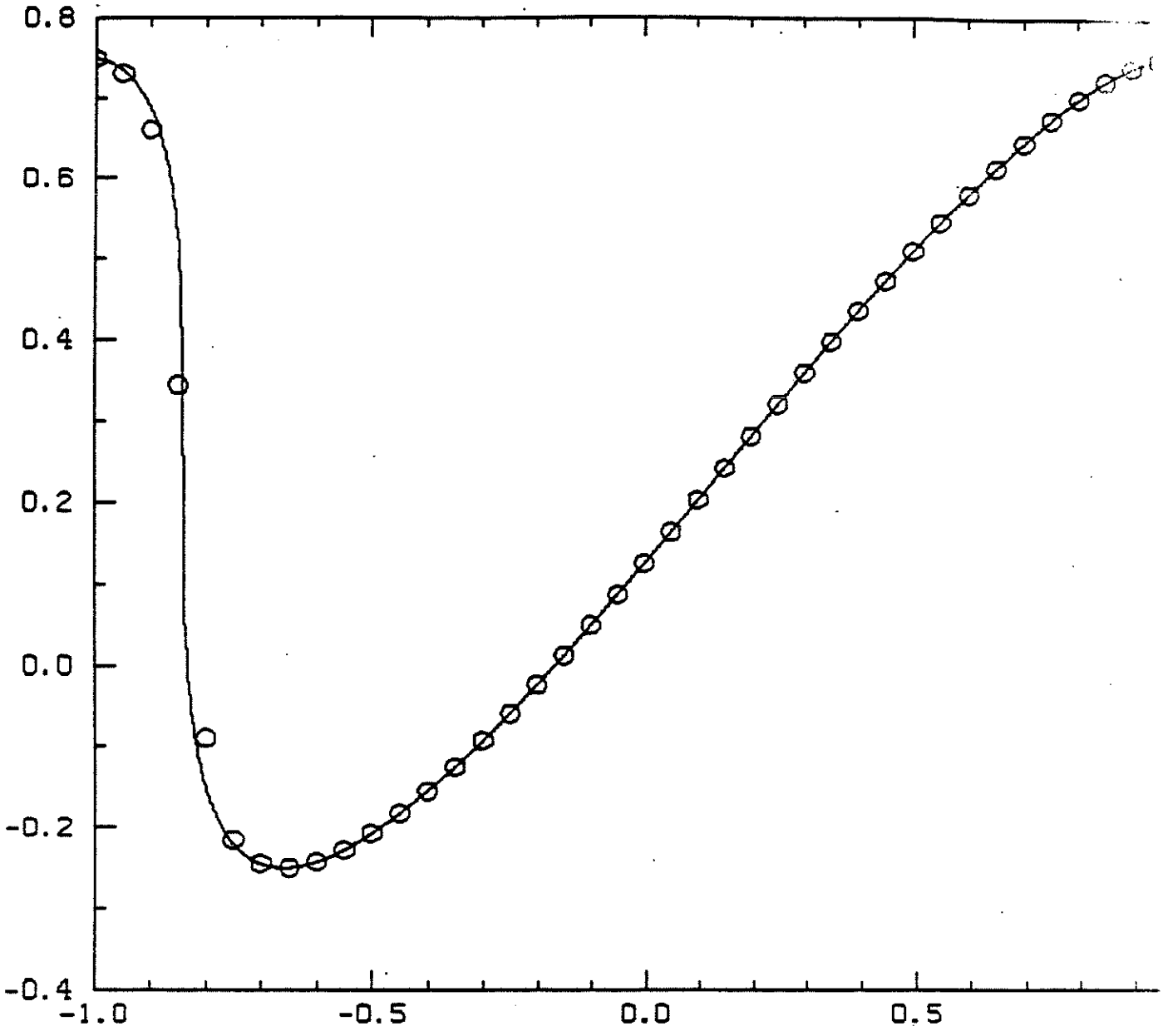

Figure 5 : 4-4-LF-ENO , $\Delta x = \frac{1}{20}$, $\epsilon = \frac{2}{\pi}$

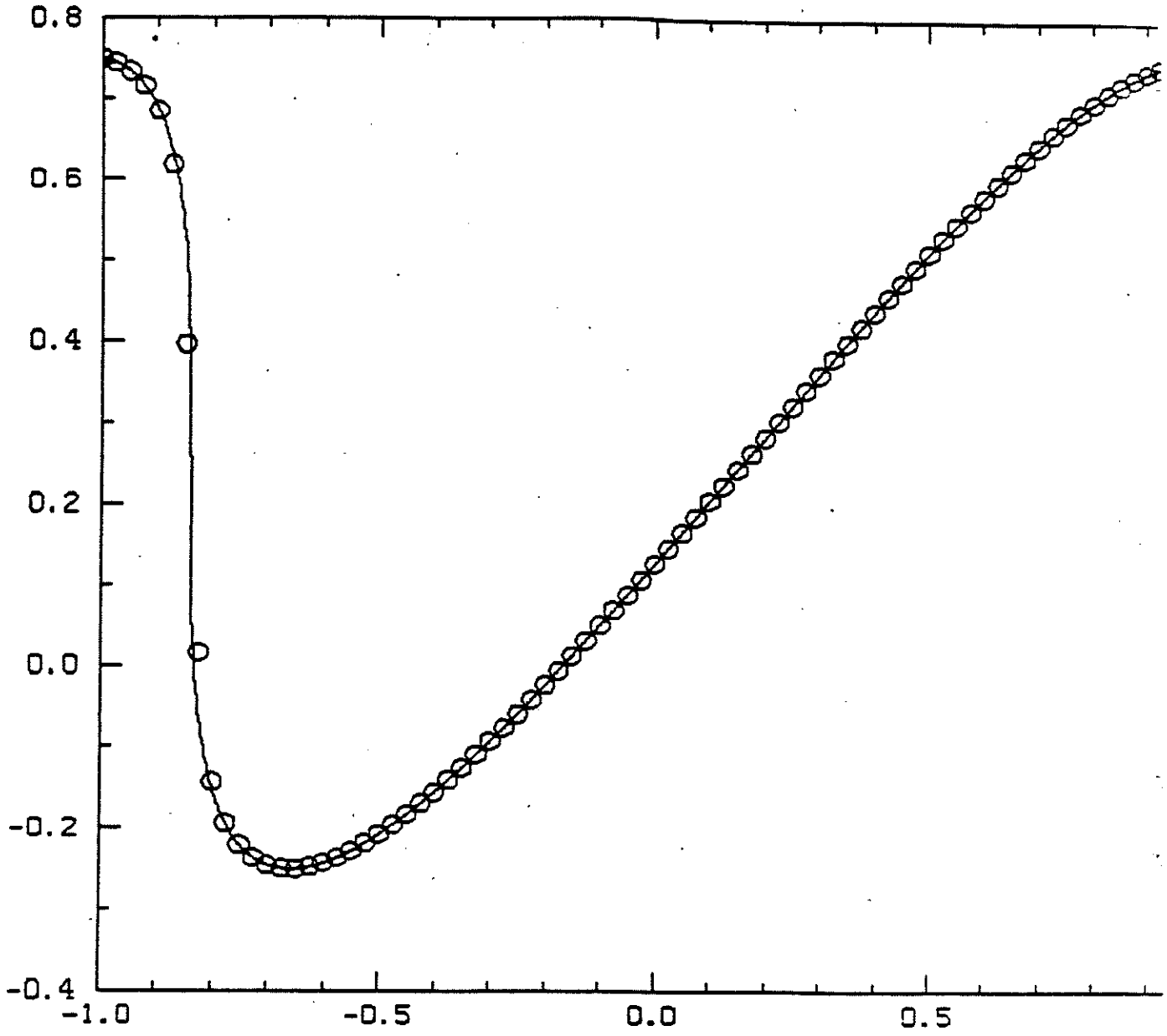

Figure 6 : 4-4-LF-ENO , $\Delta x = \frac{1}{40}$, $t = \frac{2}{\pi}$

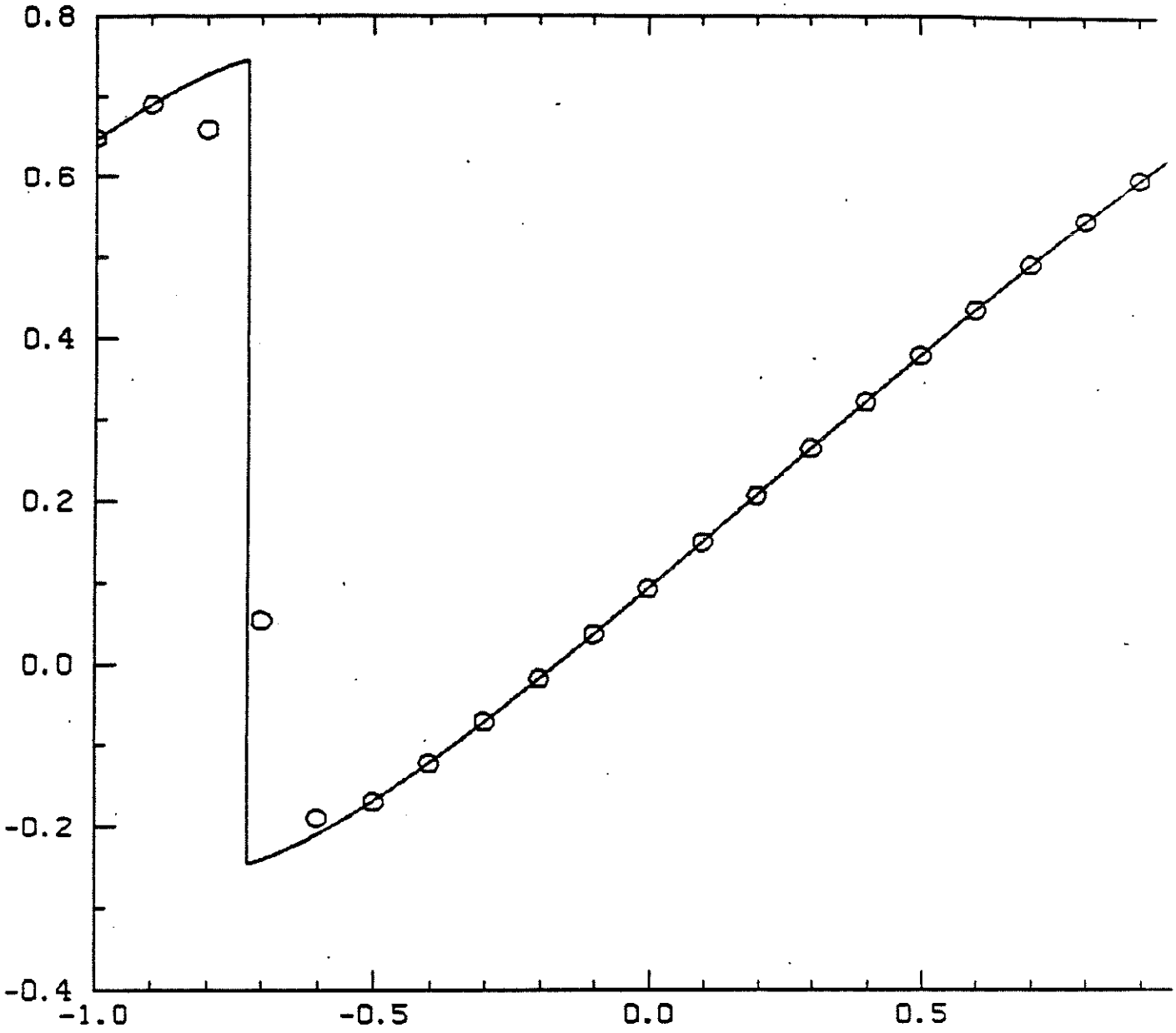

Figure 7 : 3-3-LF-ENO, $\Delta x = \frac{1}{10}$, $t = 1.1$

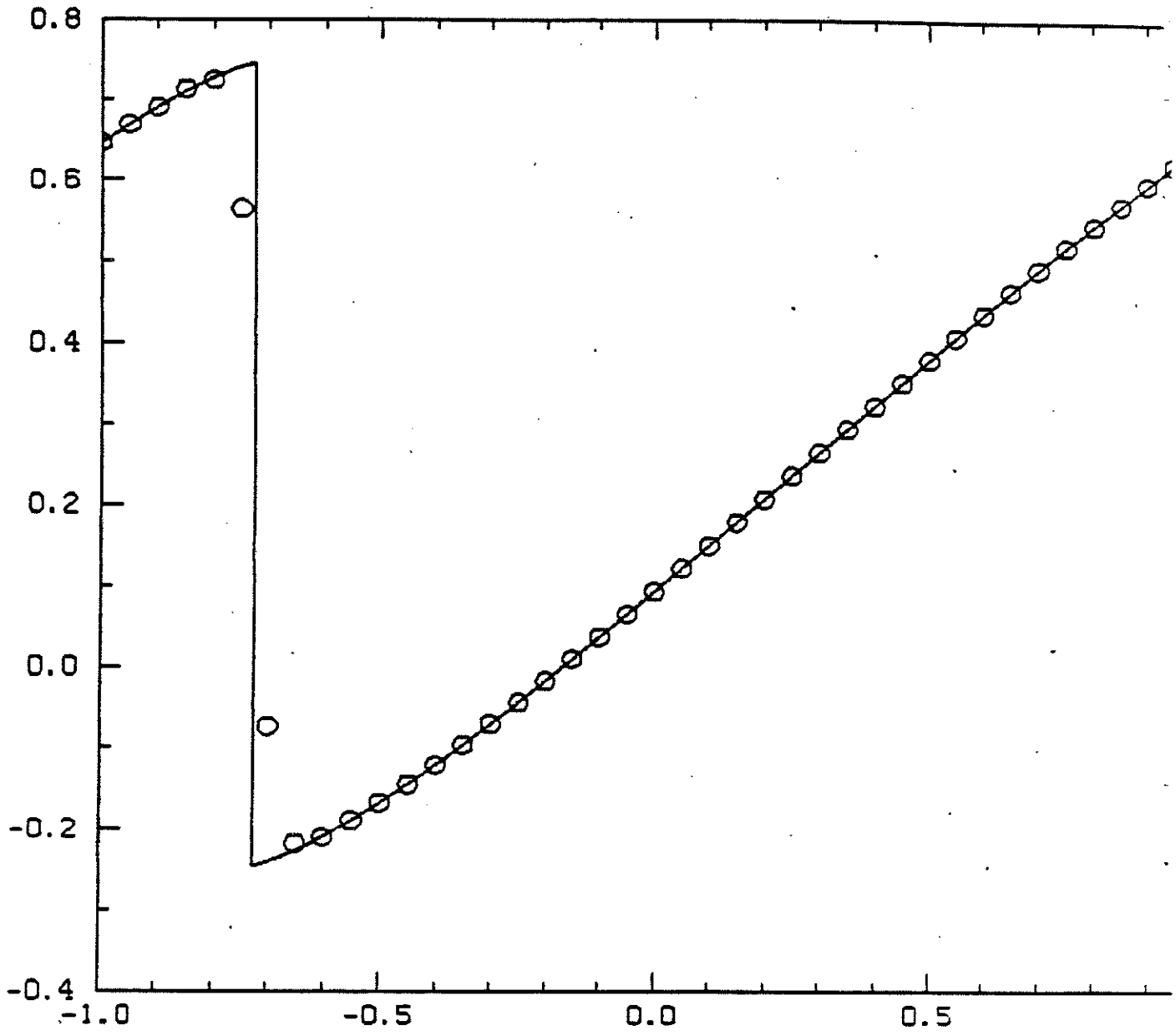

Figure 8 : 3-3-LF-ENO, $\Delta x = \frac{1}{20}$, $t = 1.1$

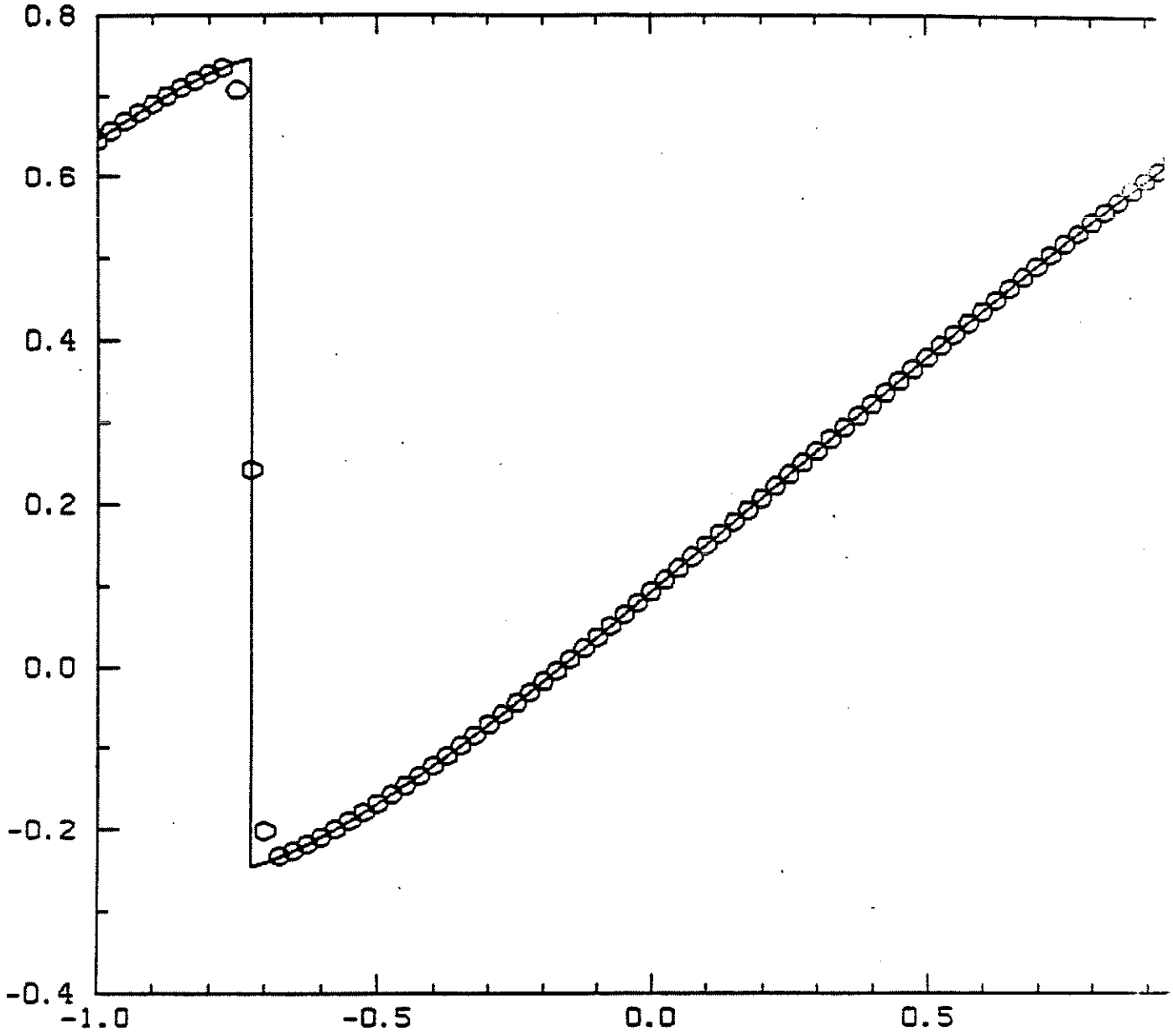

Figure 9 : 3 - 3 - LF - ENO, $\Delta x = \frac{1}{40}$, $t = 1.1$

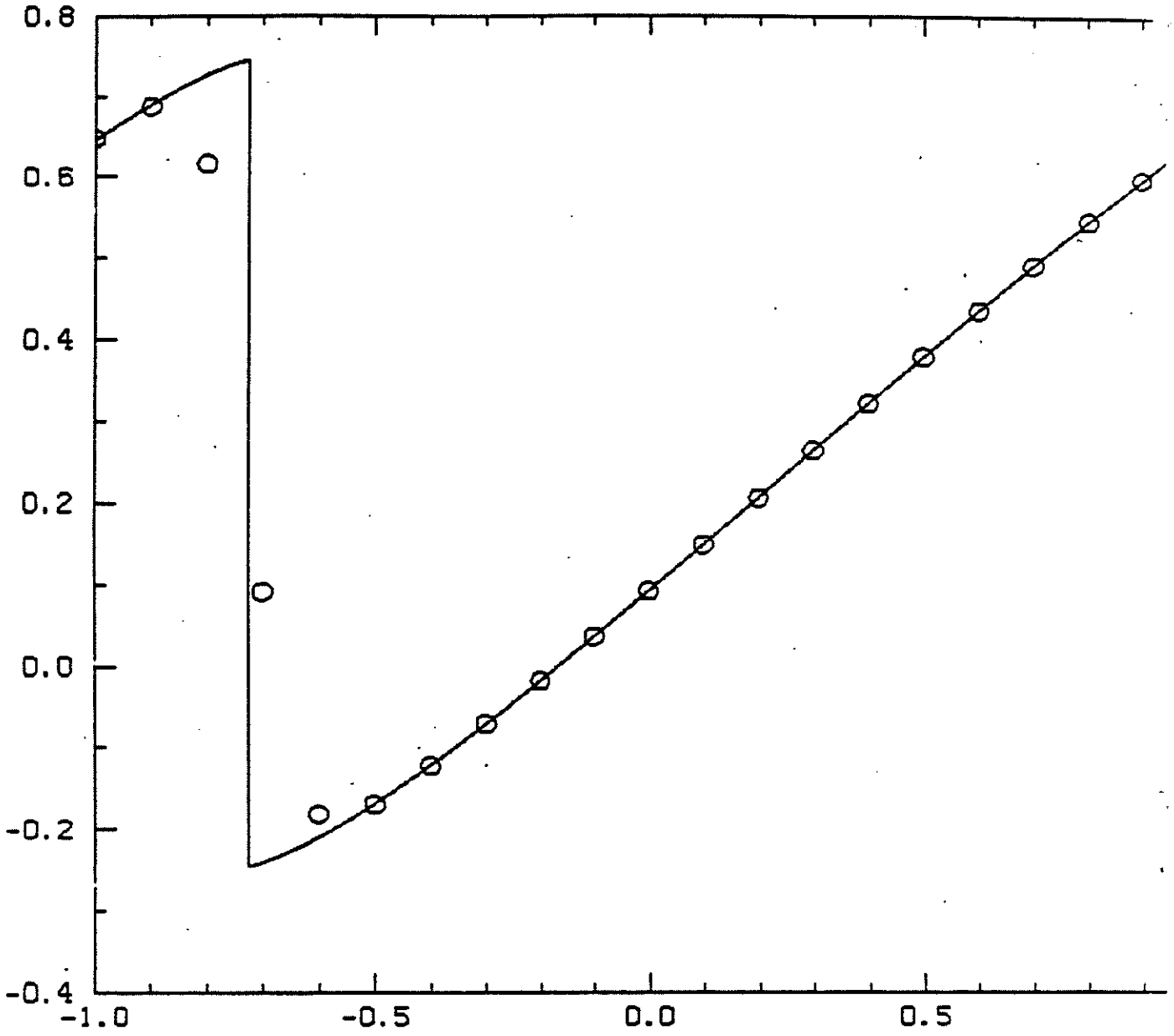

Figure 10: 4-4-LF-ENO, $\Delta x = \frac{1}{10}$, $t = 1.1$

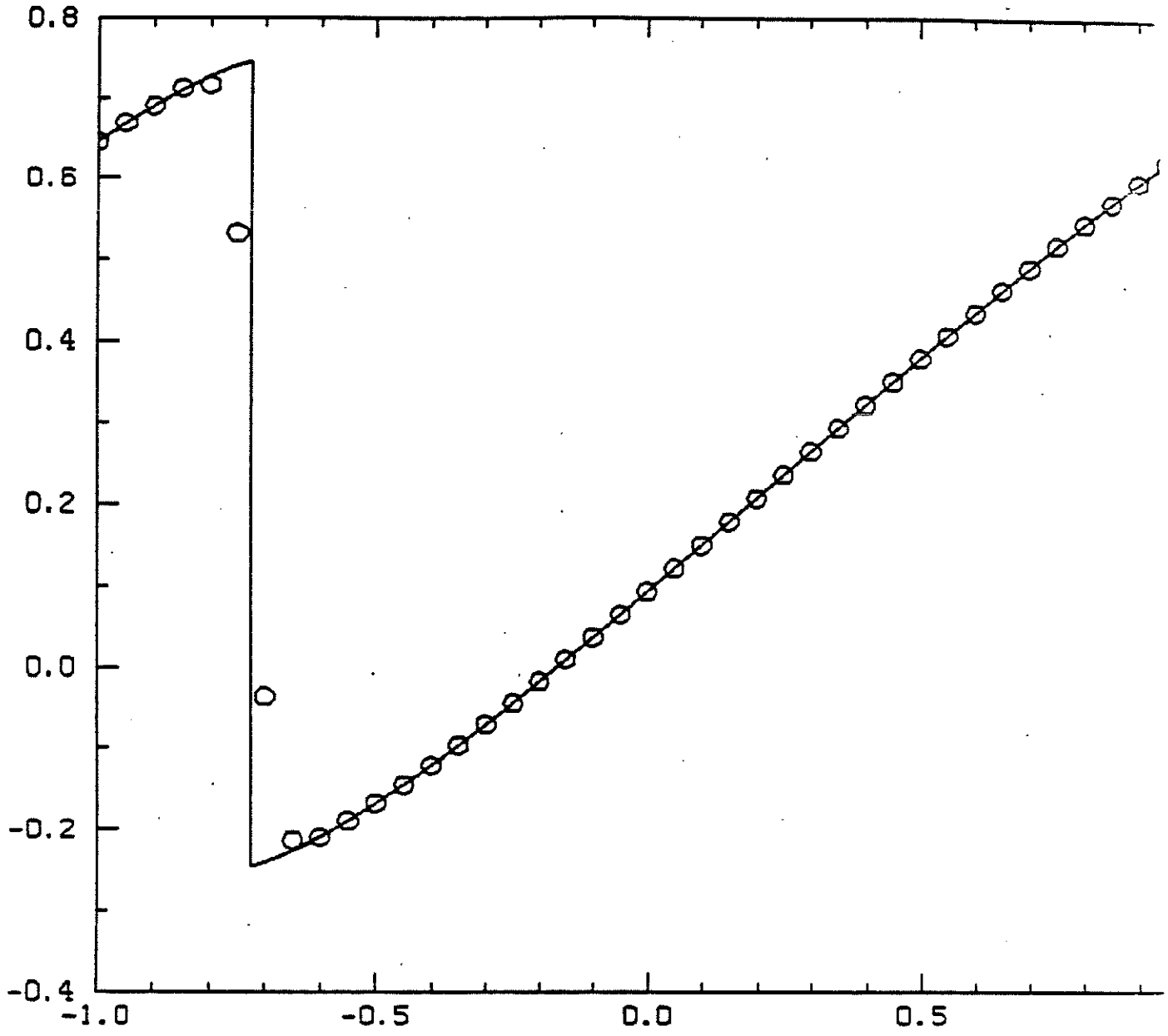

Figure 11 : 4-4-LF-ENO, $\Delta x = \frac{1}{20}$, $t = 1.1$

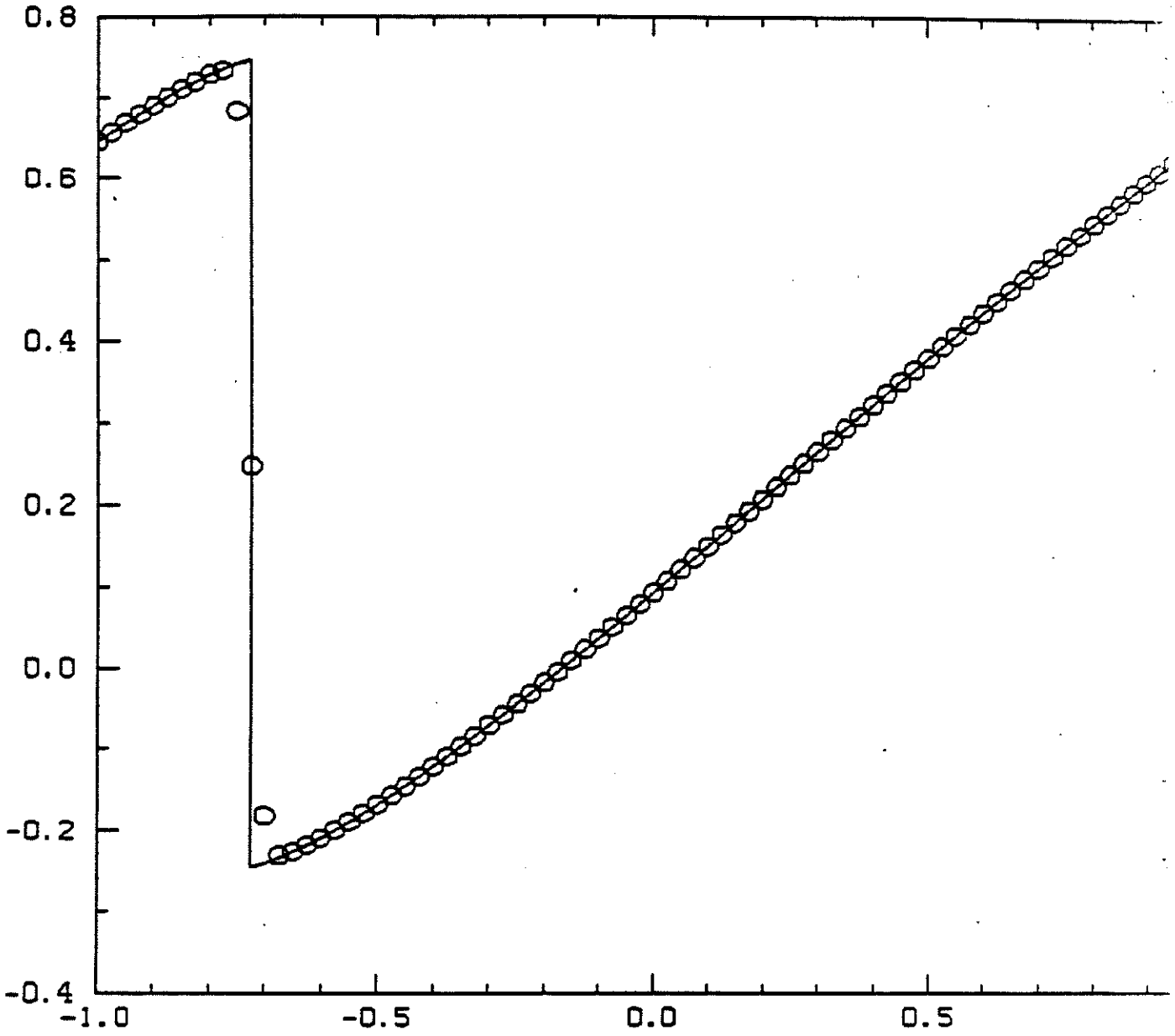

Figure 12: 4-4-LF-ENO, $\Delta x = \frac{1}{40}$, $t = 1.1$

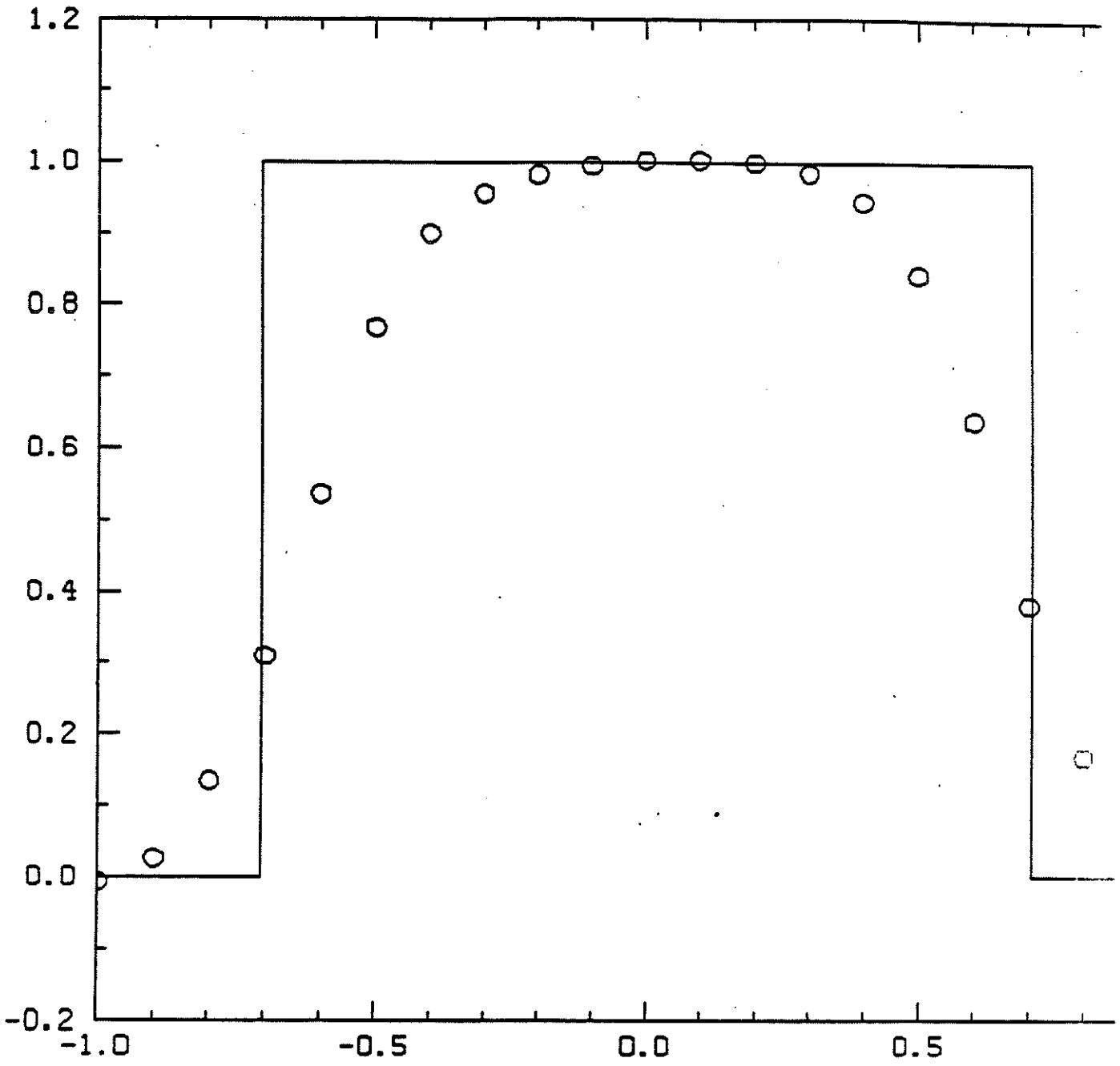

Figure 13 : 3-3-LF-ENO , $y = 0$, $t = 2$

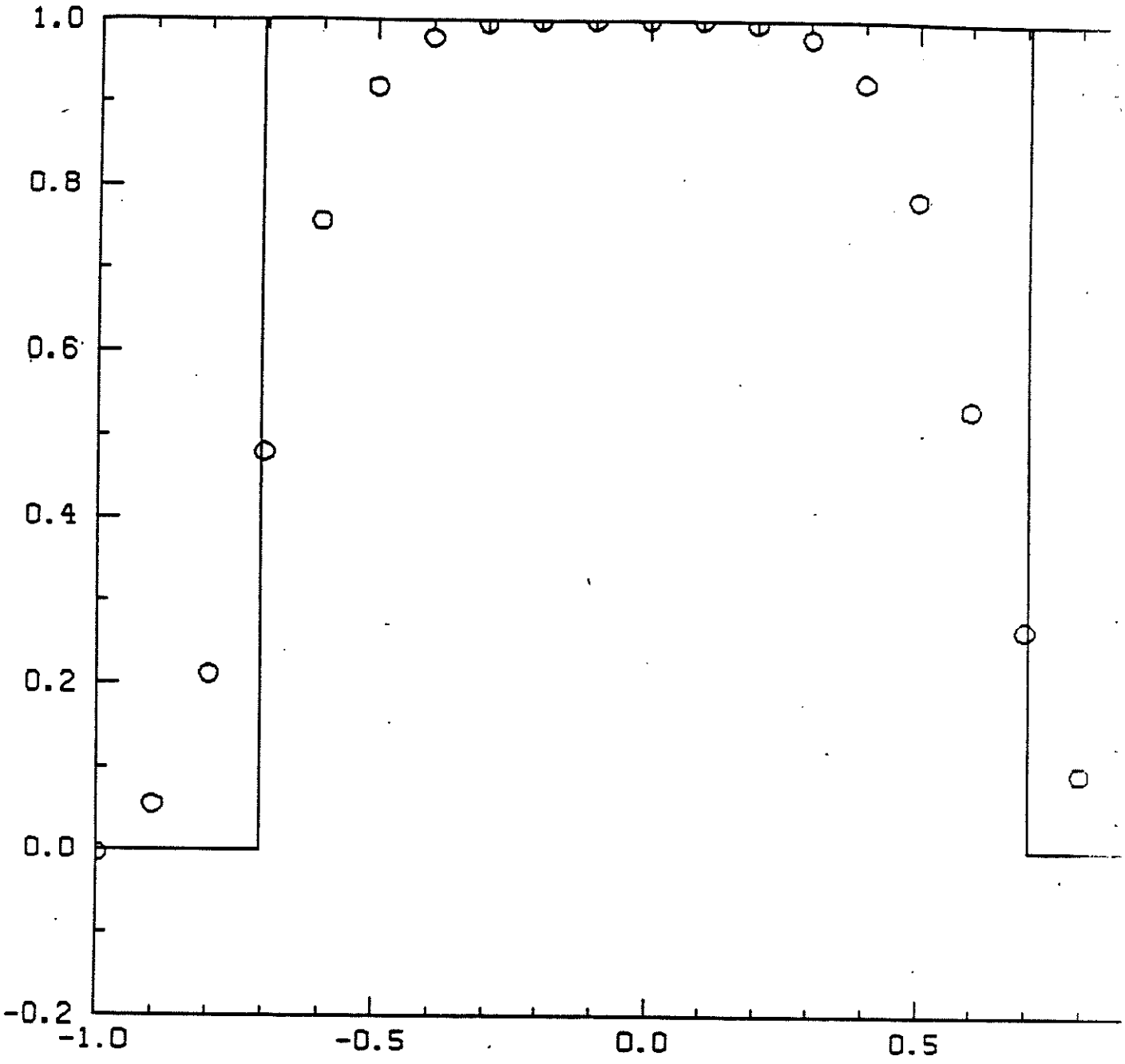

Figure 14 : 4-4-LF-ENO , $y = 0$, $t = 2$

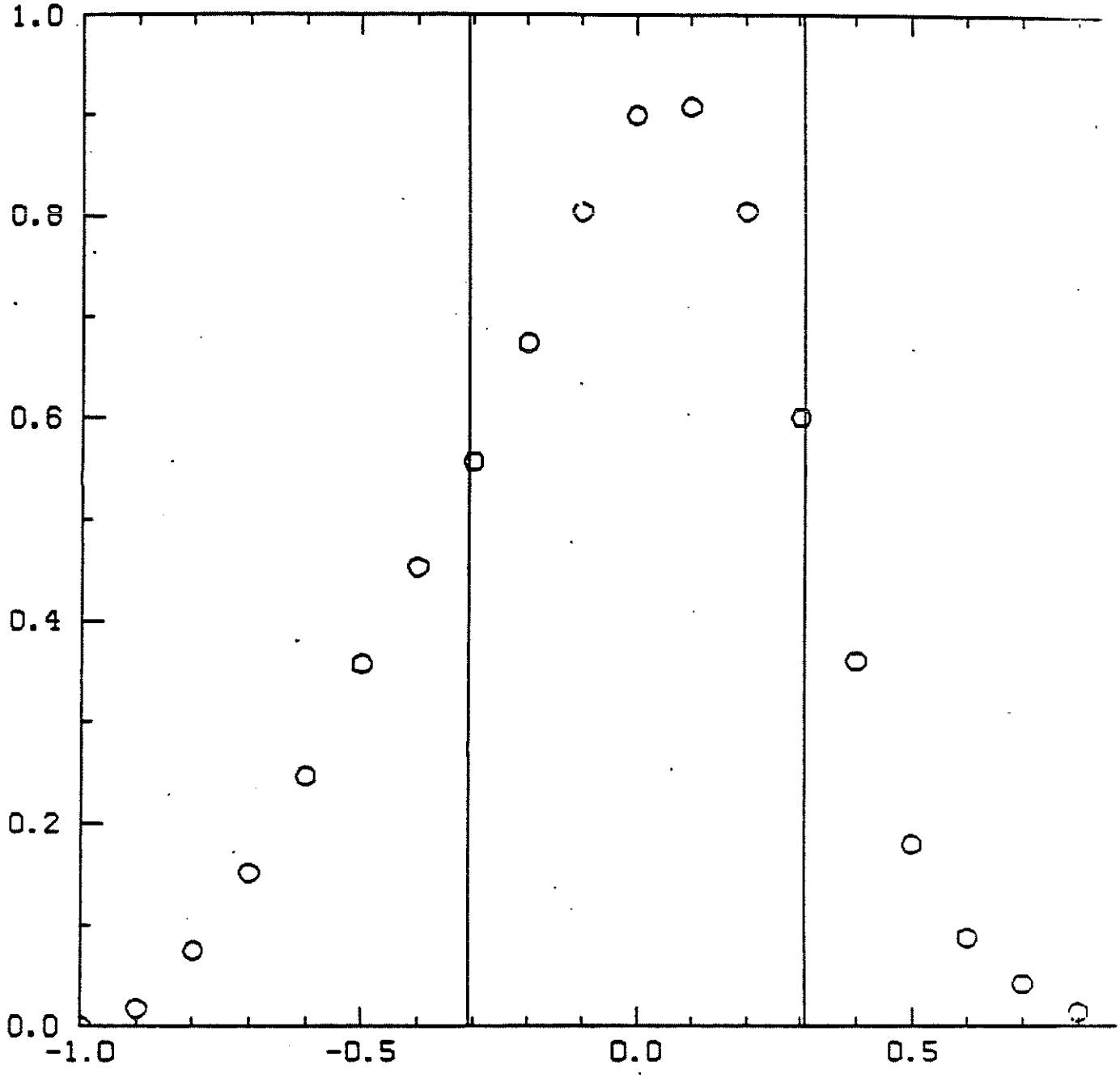

Figure 15 : 3-3-LF-ENO, $y = -0.4$, $t = 2$

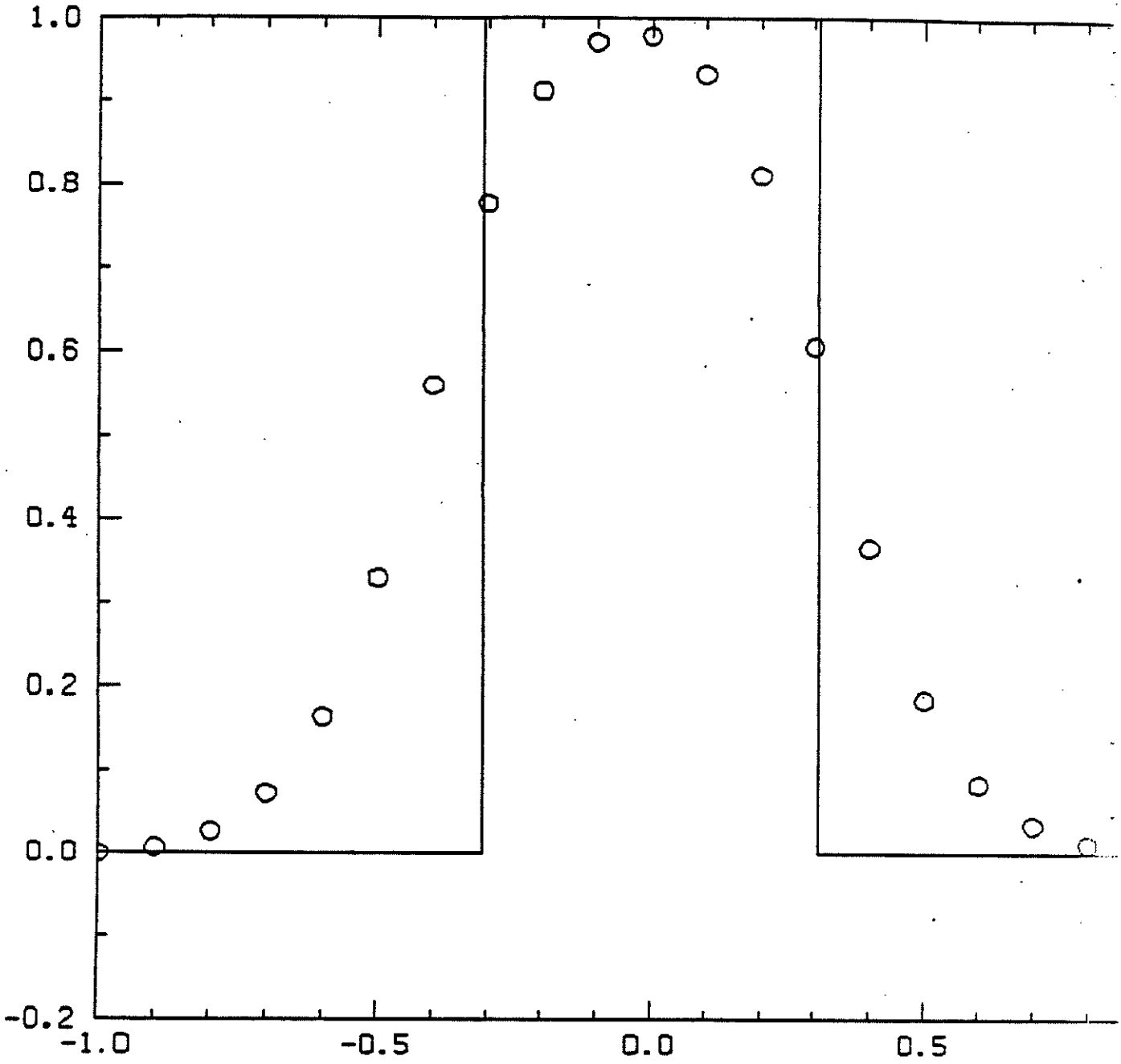

Figure 16 : 4-4-LF-ENO, $y=-0.4$, $t=2$

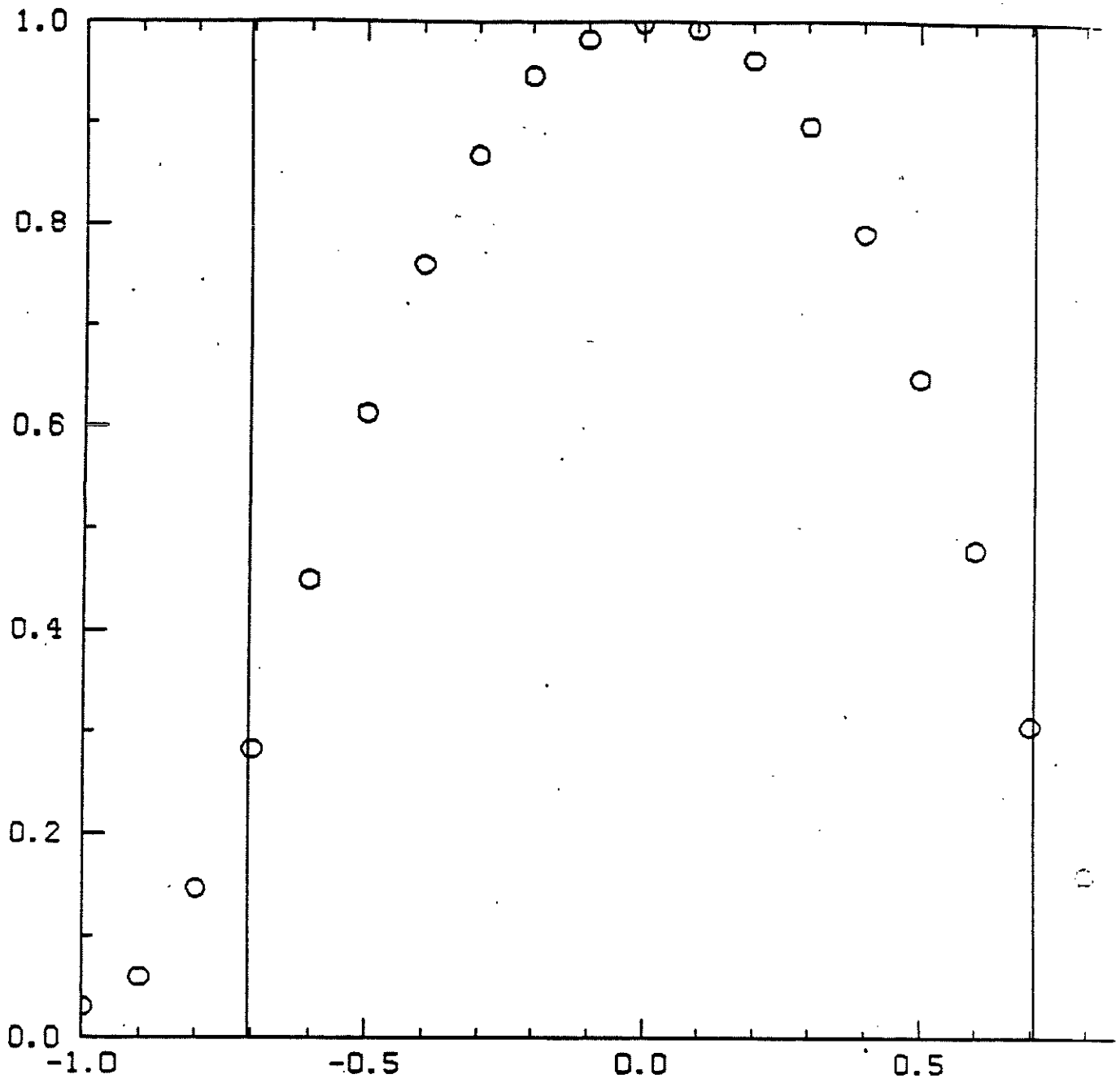

Figure 17: 3-3-LF-ENO, $y = 0$, $t = 16$

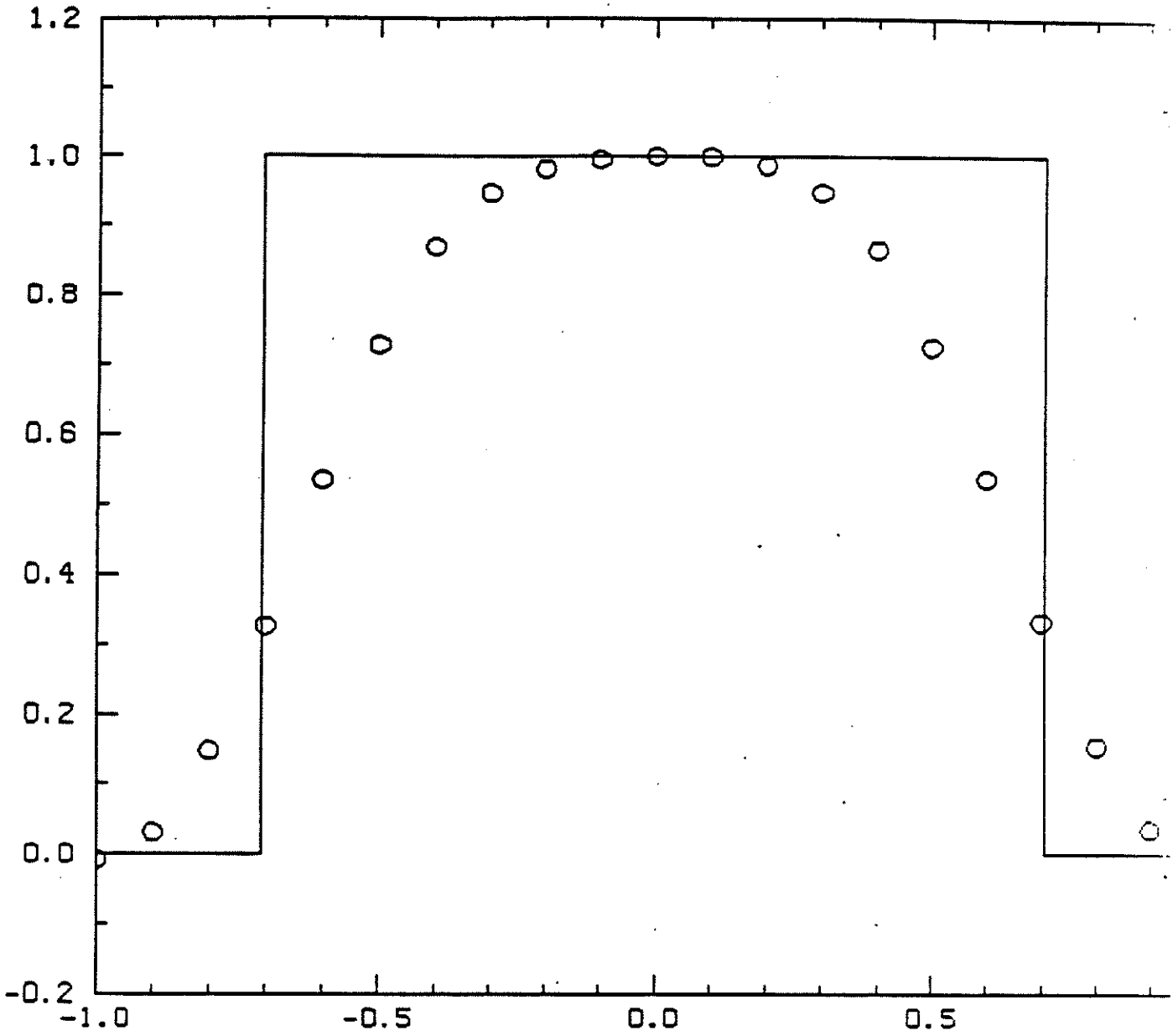

Figure 18: 4-4-LF-ENO, $y=0$, $t=16$

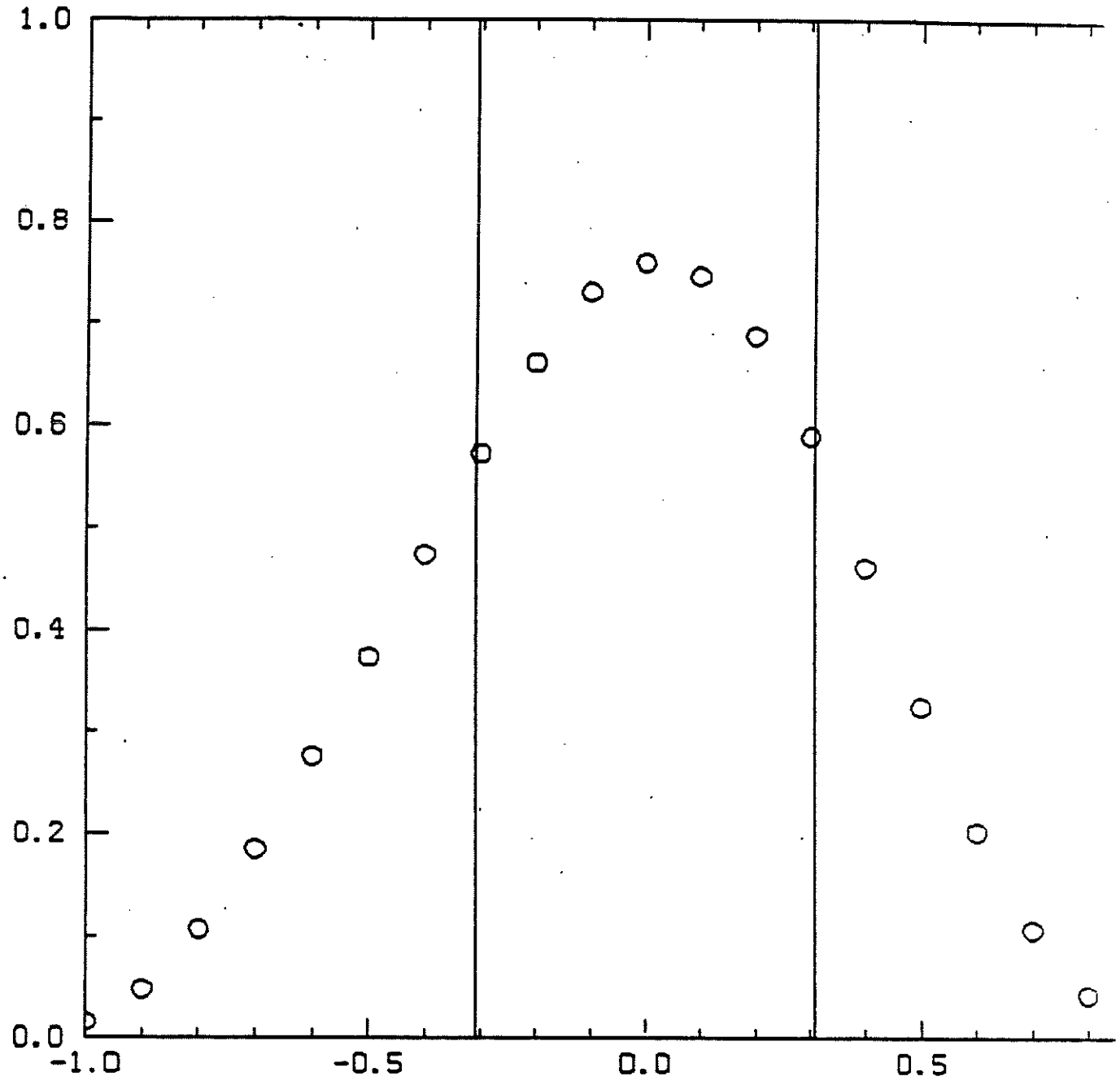

Figure 19 : 3-3-LF-ENO, $y = -c.4$, $t = 16$

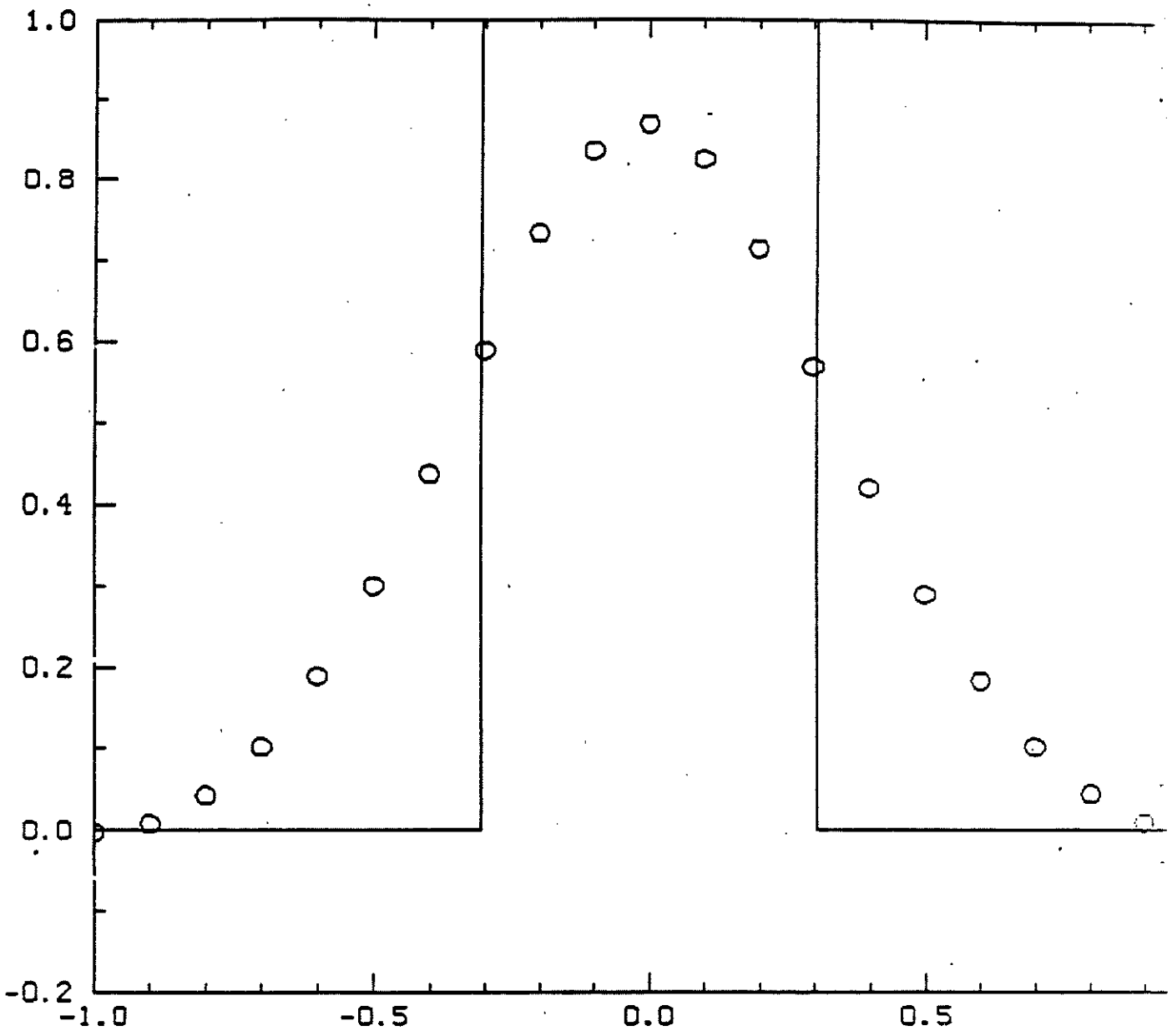

Figure 20: 4-4-LF-ENO, $y=-0.4$, $t=16$

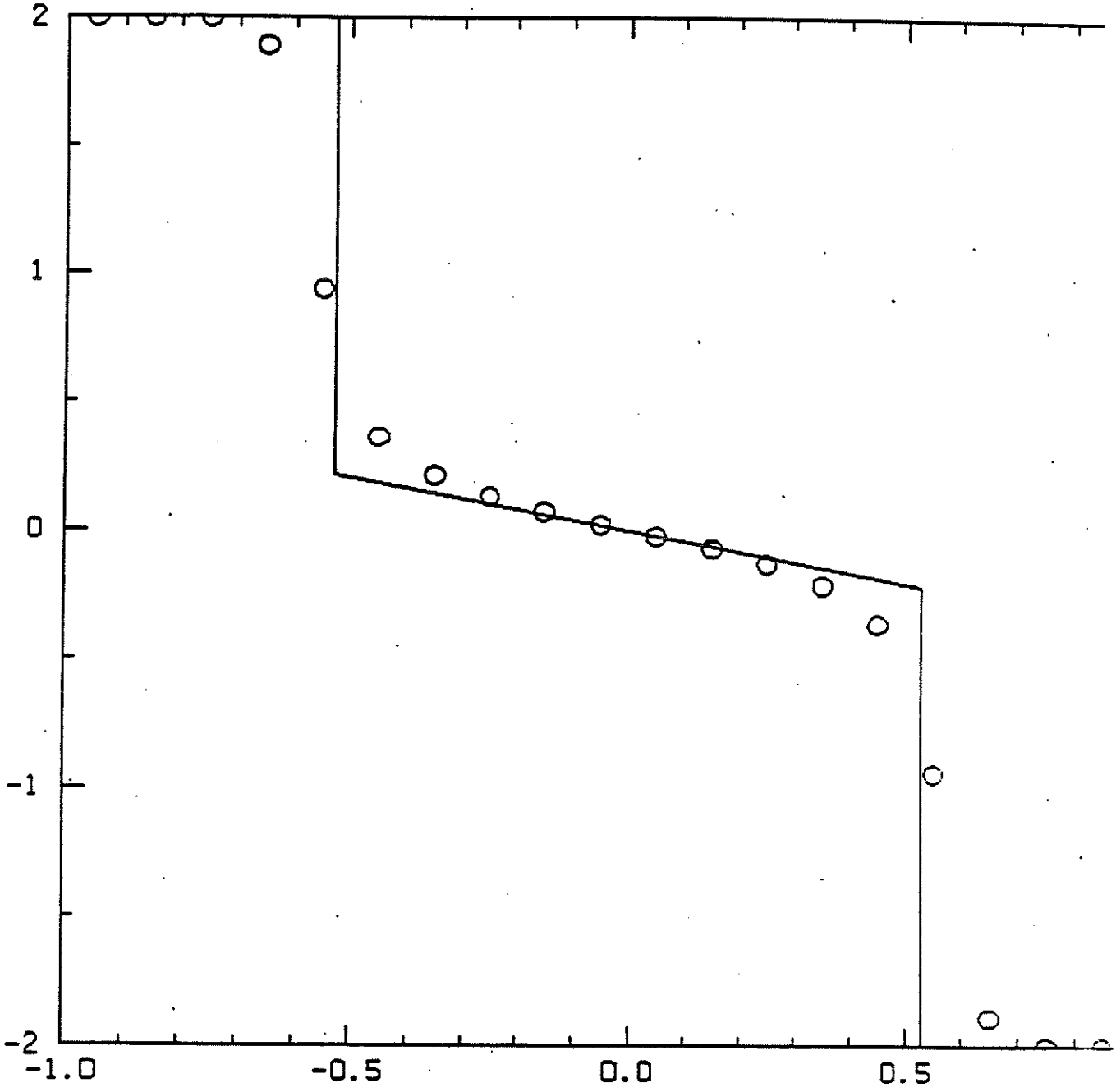

Figure 21 : 3-3-LF-ENO, $\Delta x = \frac{1}{10}$

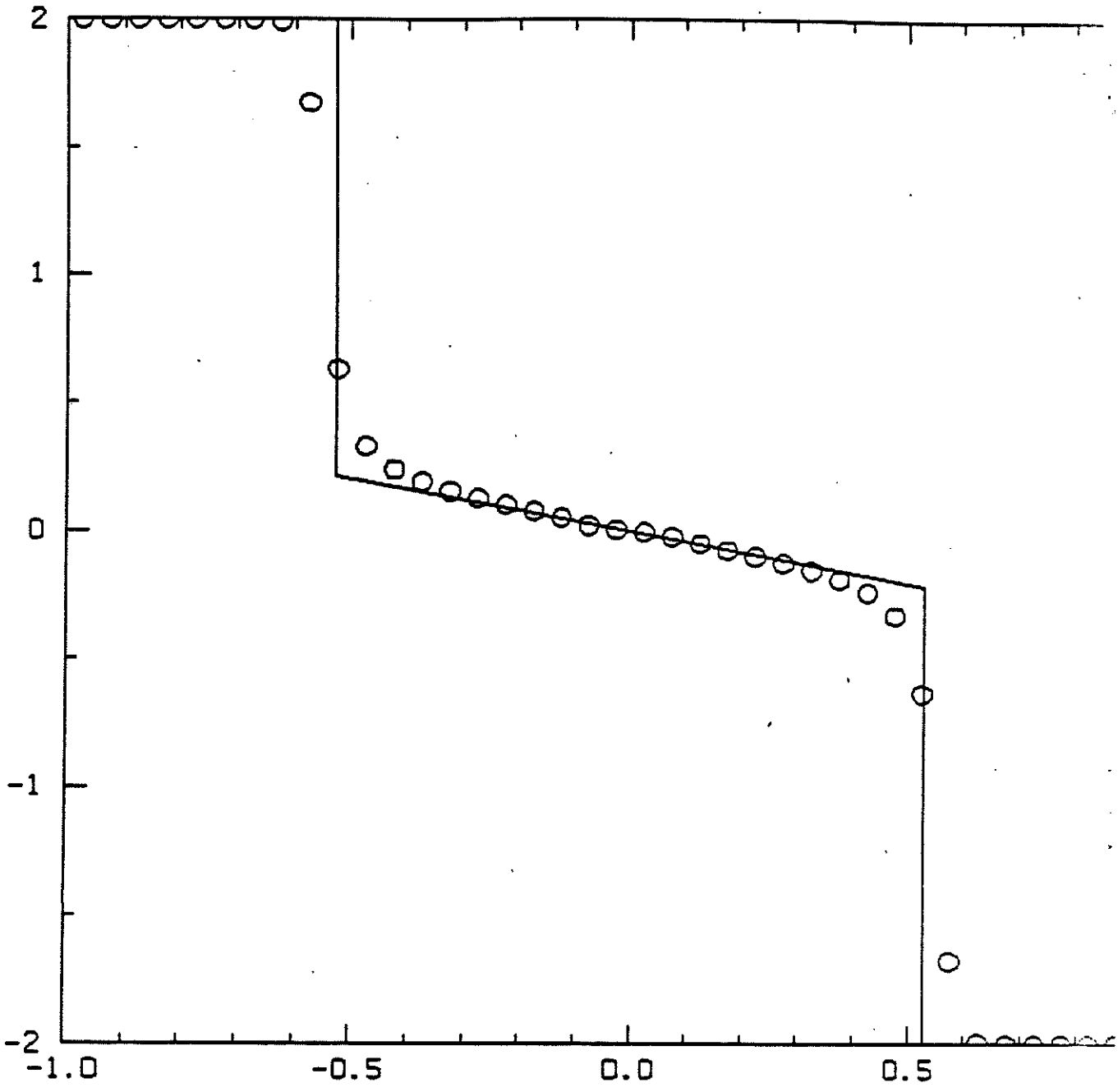

Figure 22 : 3-3-LF-ENO, $\Delta x = \frac{1}{20}$

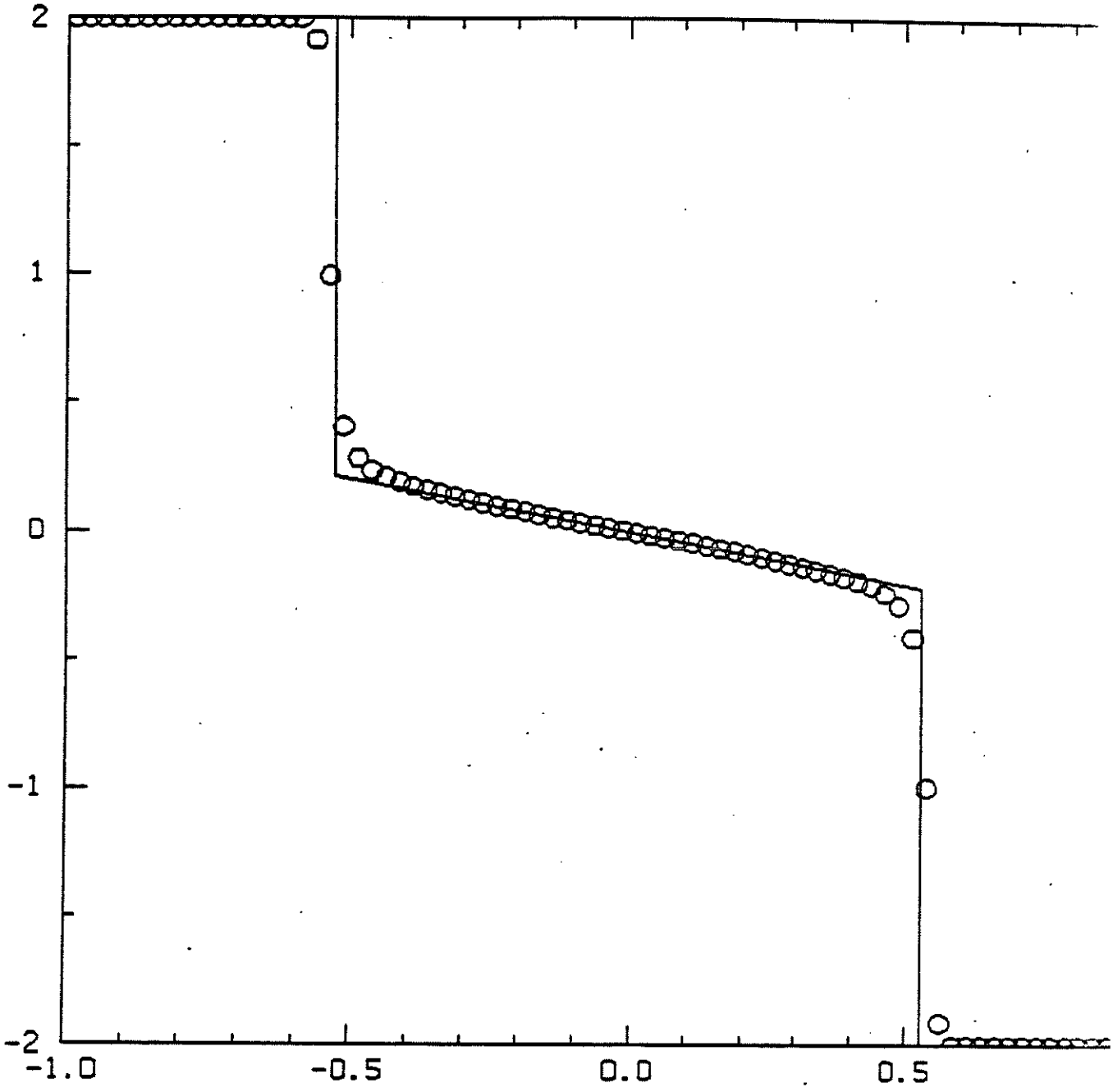

Figure 23: 3-3-LF-ENO, $\Delta x = \frac{1}{40}$

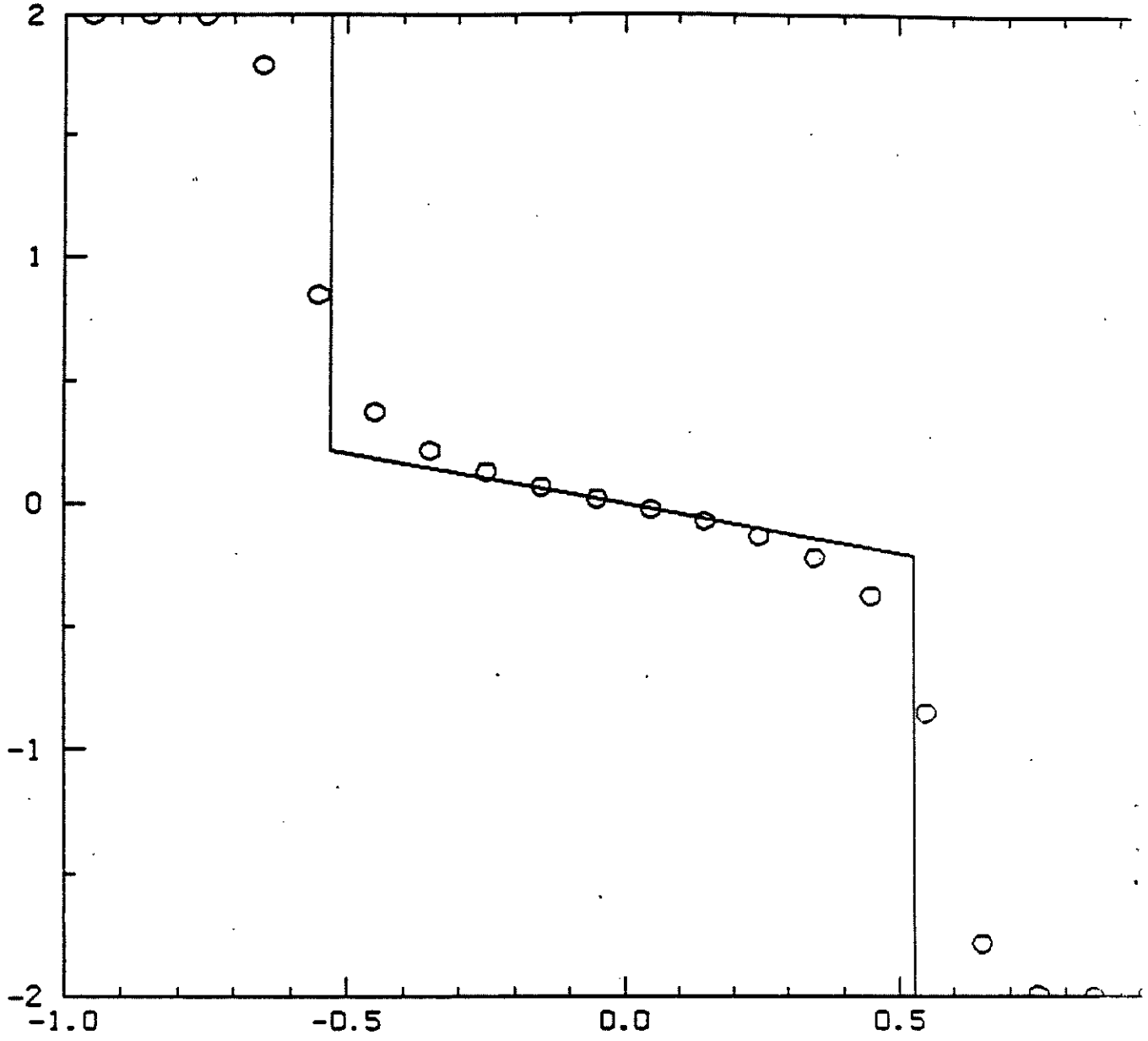

Figure 24 : 4-4-LF-ENO, $\Delta x = \frac{1}{10}$

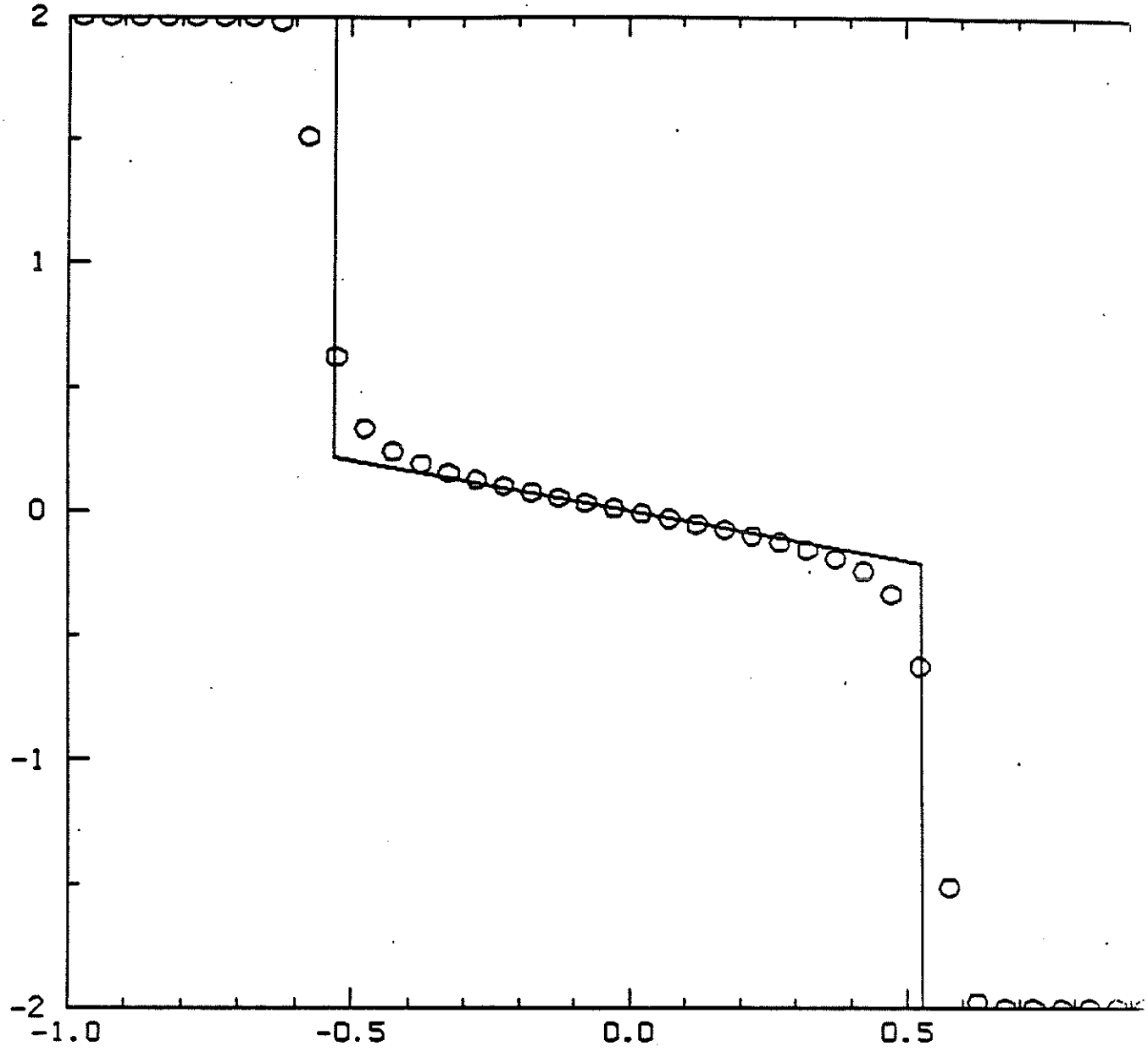

Figure 25 : 4-4-LF-ENO , $\Delta x = \frac{1}{20}$

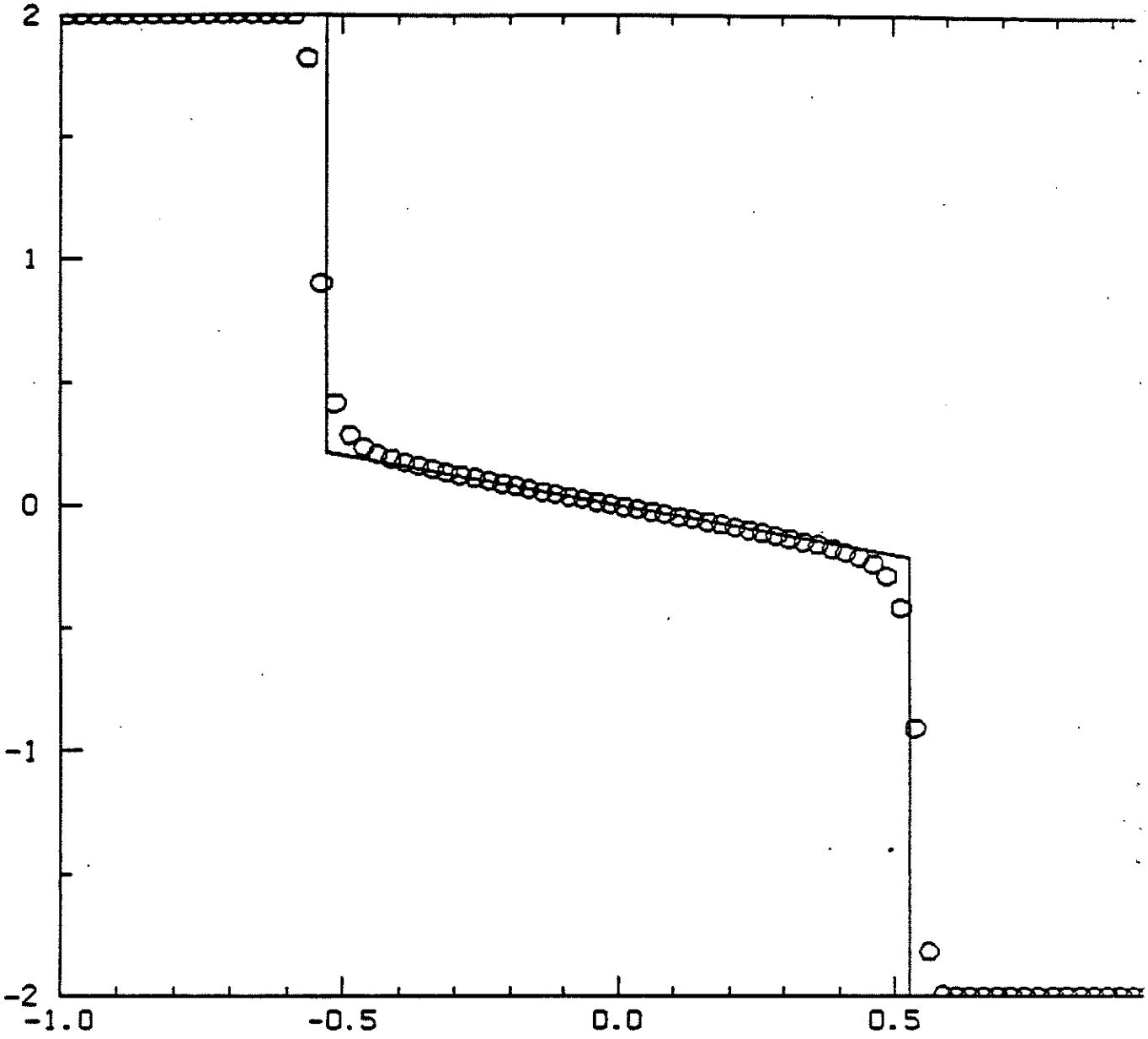

Figure 26 : 4-4-LF-ENO , $\Delta x = \frac{1}{40}$

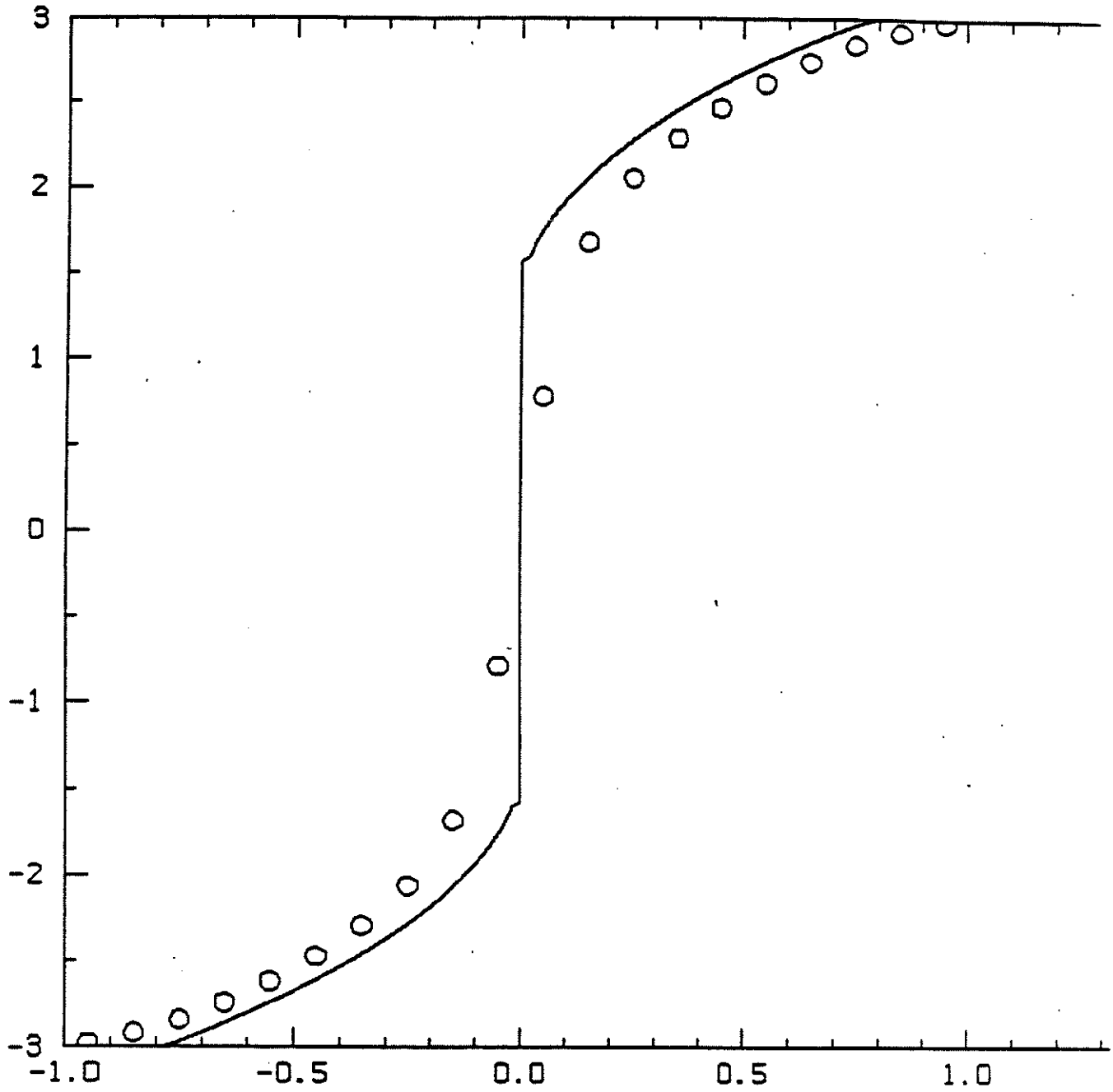

Figure 27 : 3-3-LF-ENO , $\Delta x = \frac{1}{10}$

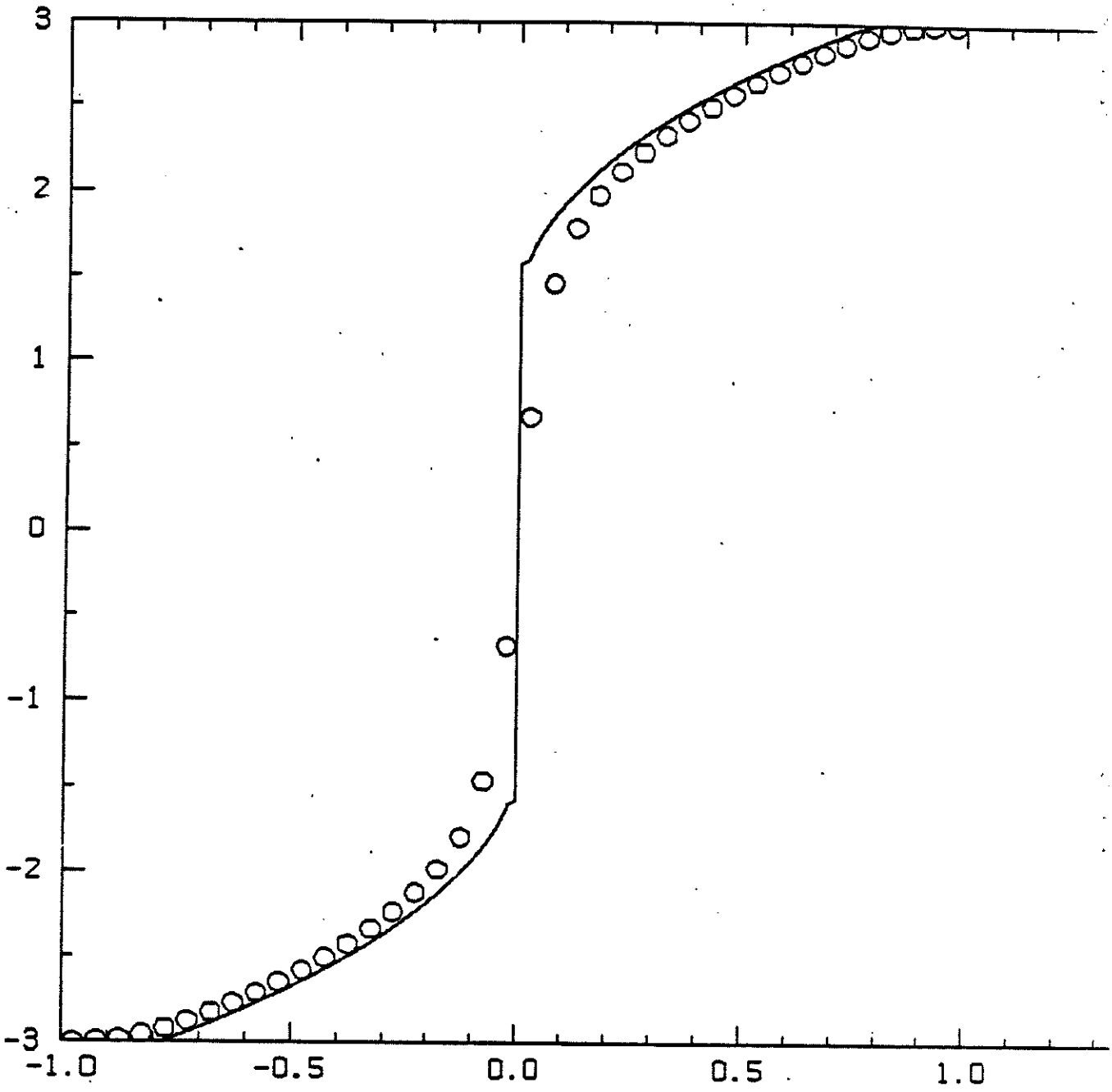

Figure 28: 3-3-LF-ENO, $\Delta x = \frac{1}{20}$

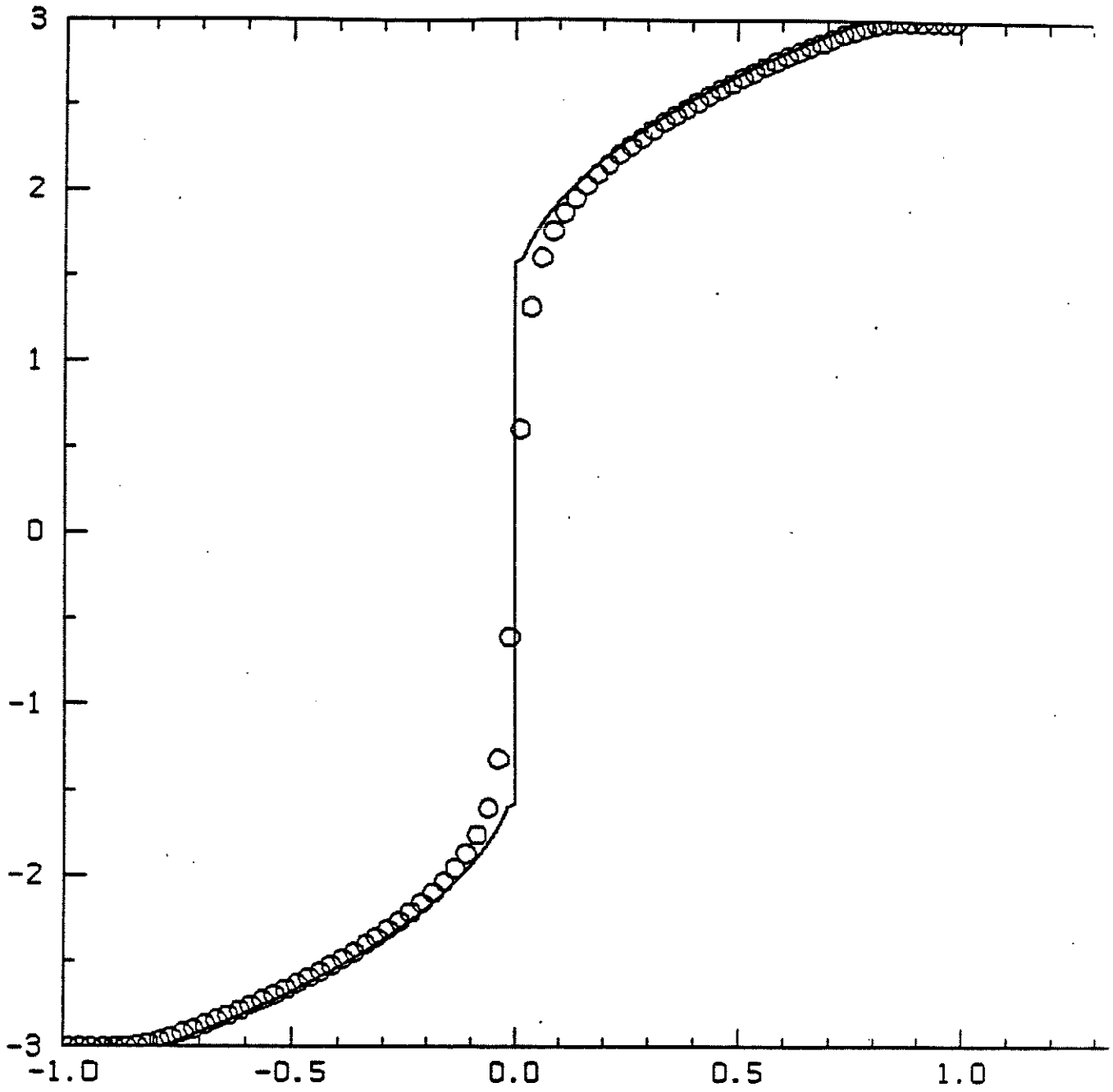

Figure 29: 3-3-LF-ENO, $\Delta x = \frac{1}{4c}$

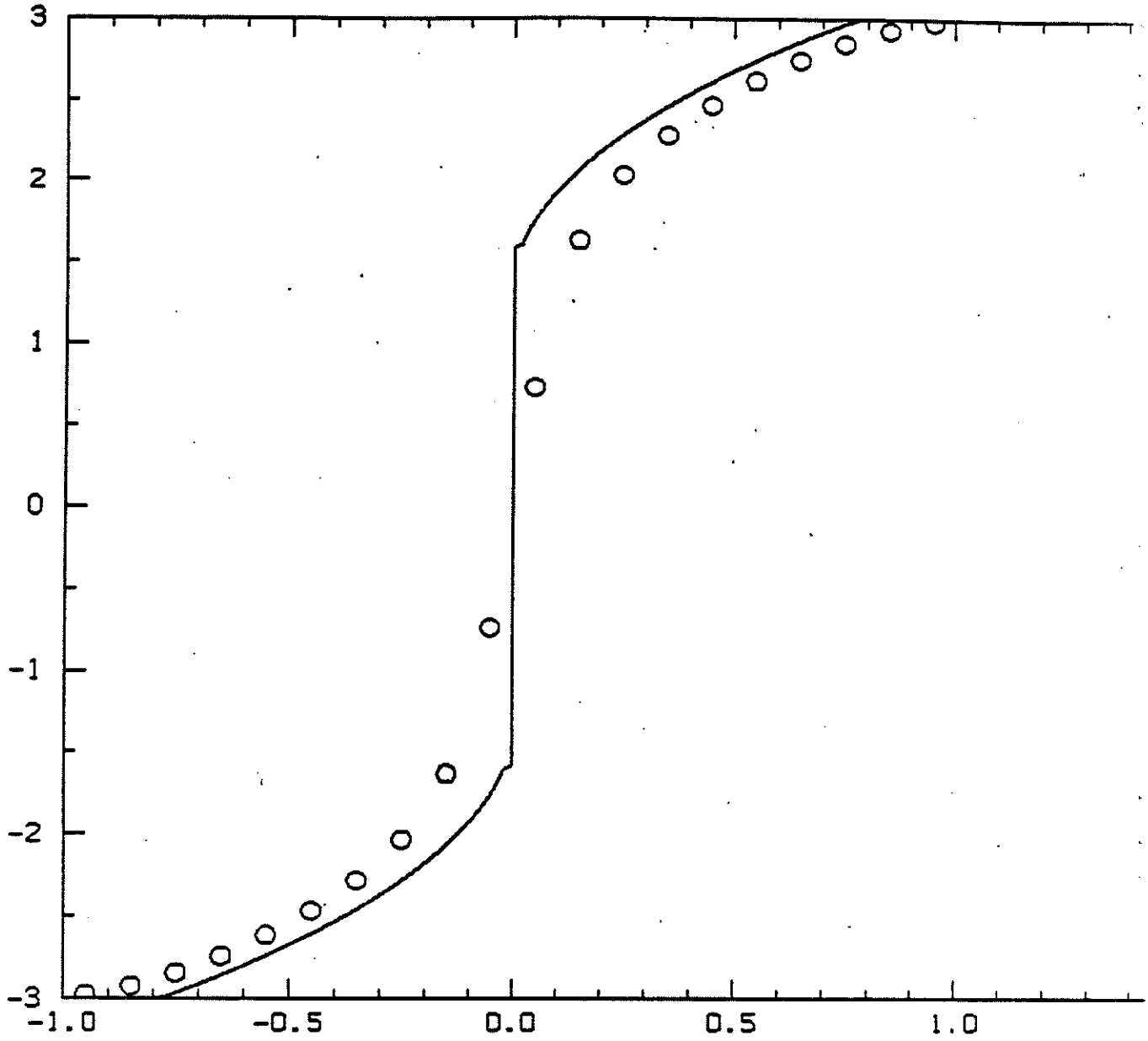

Figure 30 : 4-4-LF-ENO , $\Delta x = \frac{1}{10}$

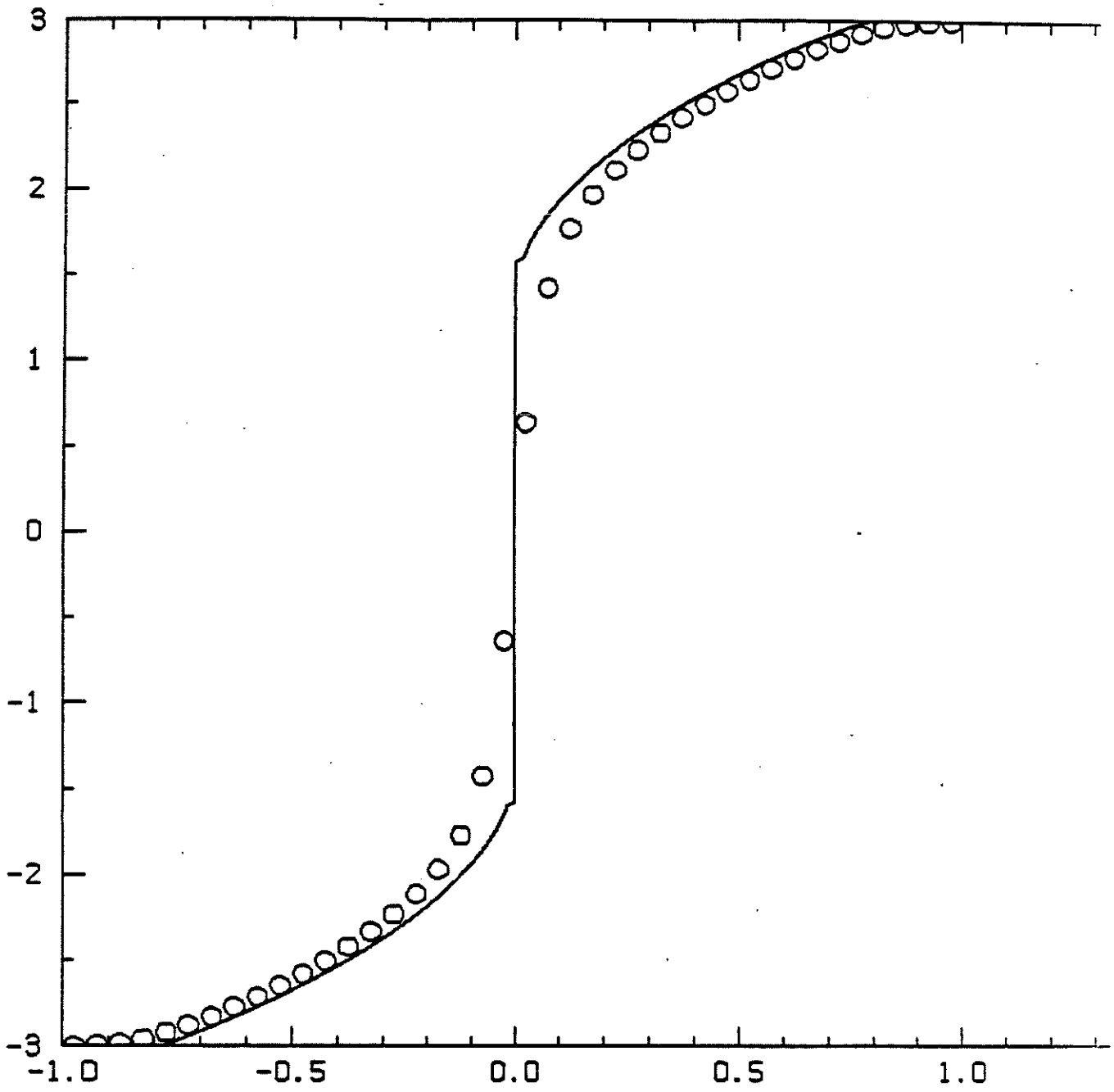

Figure 31 : 4-4- LF - ENO , $\Delta x = \frac{1}{20}$

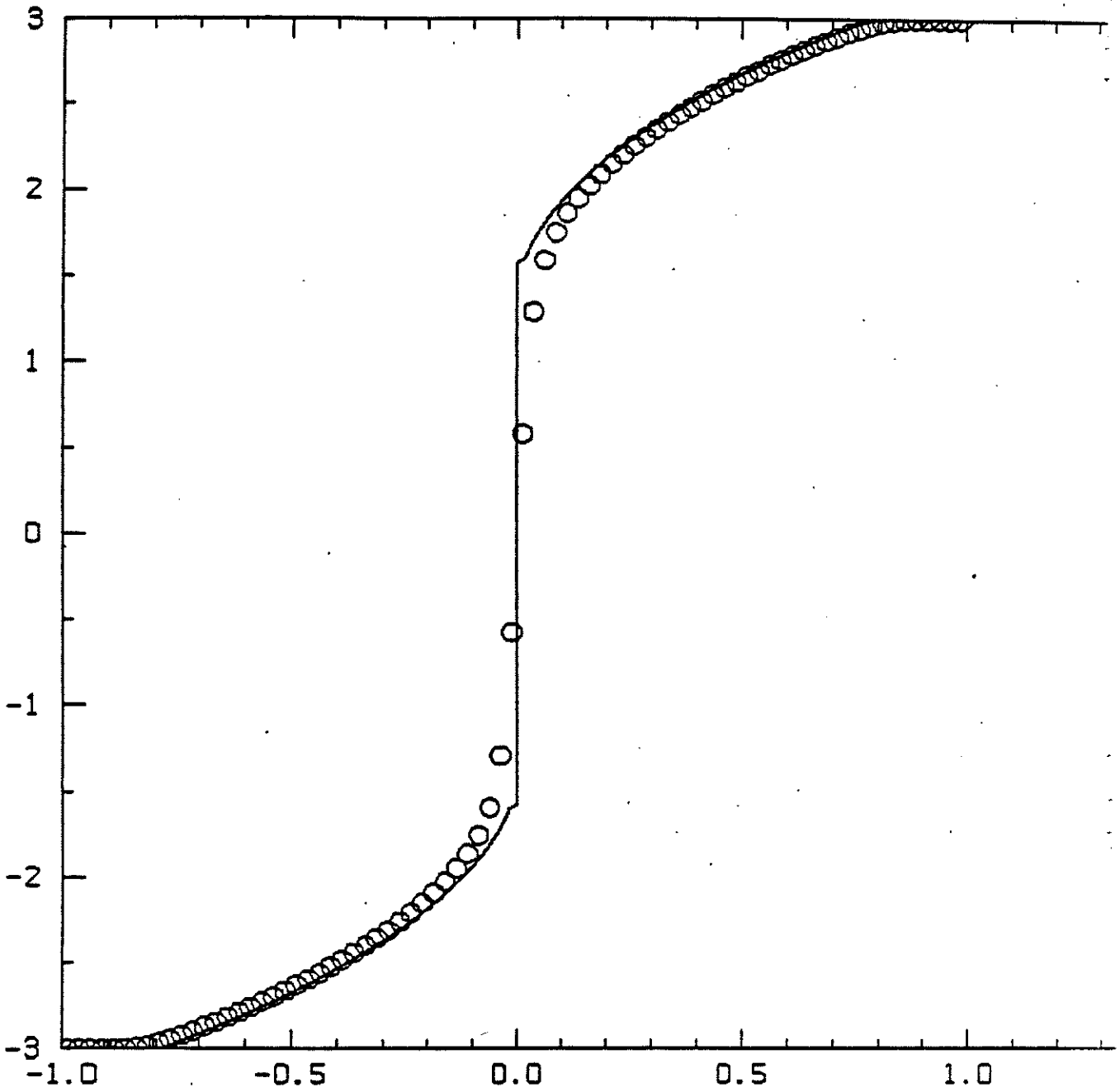

Figure 32: 4-4-LF-ENO, $\Delta x = \frac{1}{40}$

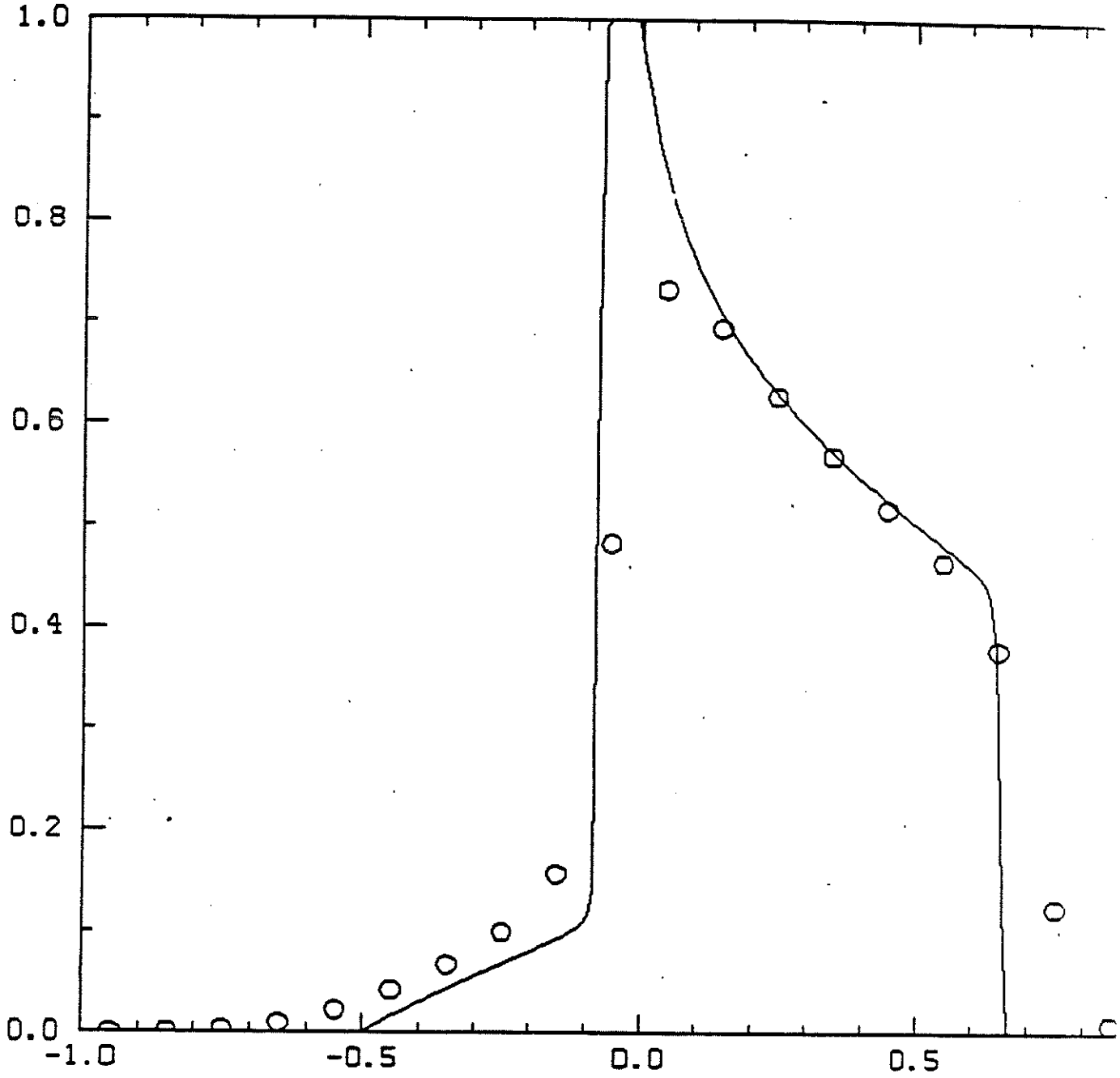

Figure 33 : 3-3-LF-ENO : $\Delta x = \frac{1}{10}$

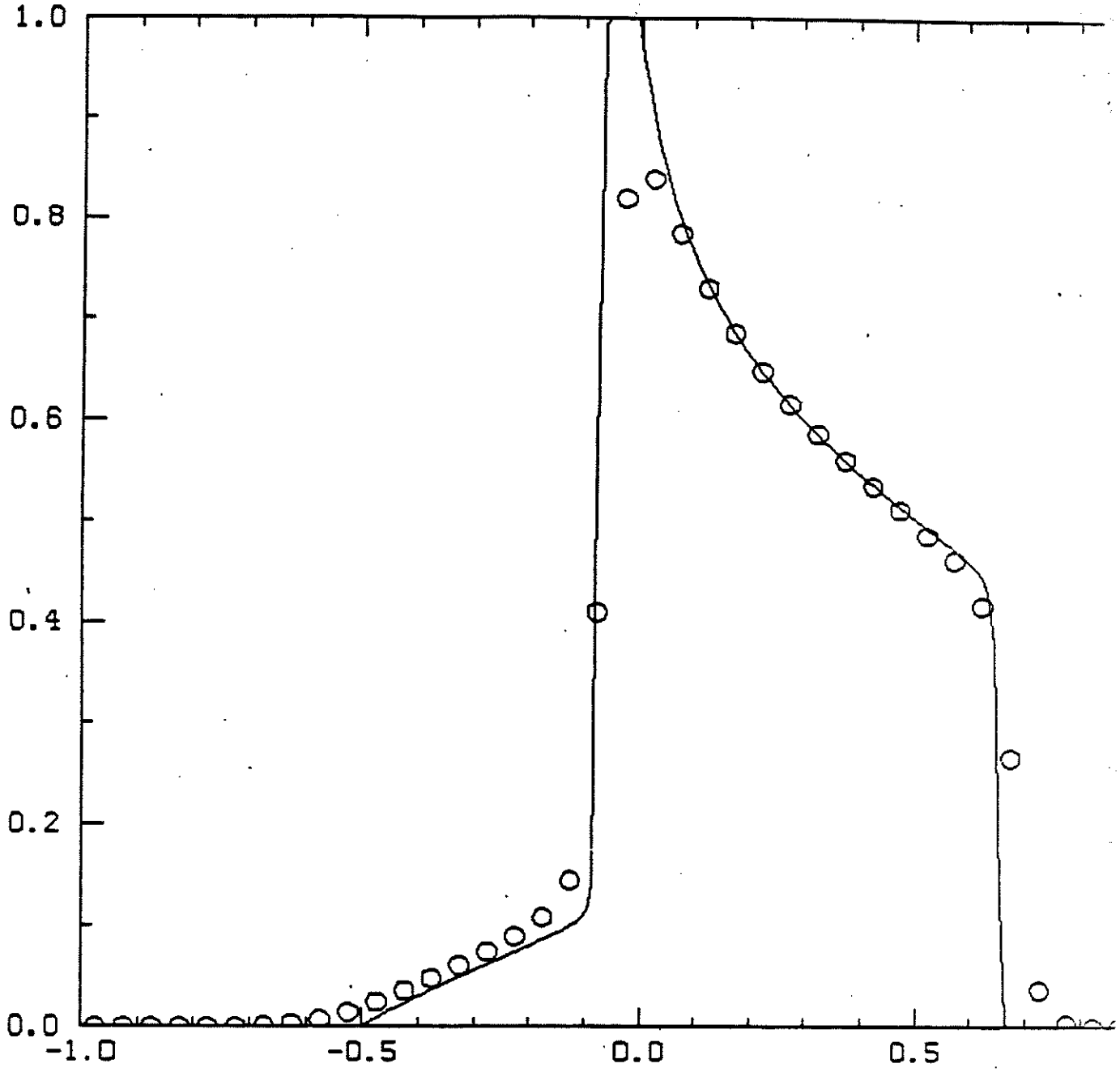

Figure 34 : 3-3-LF-ENO, $\Delta x = \frac{1}{20}$

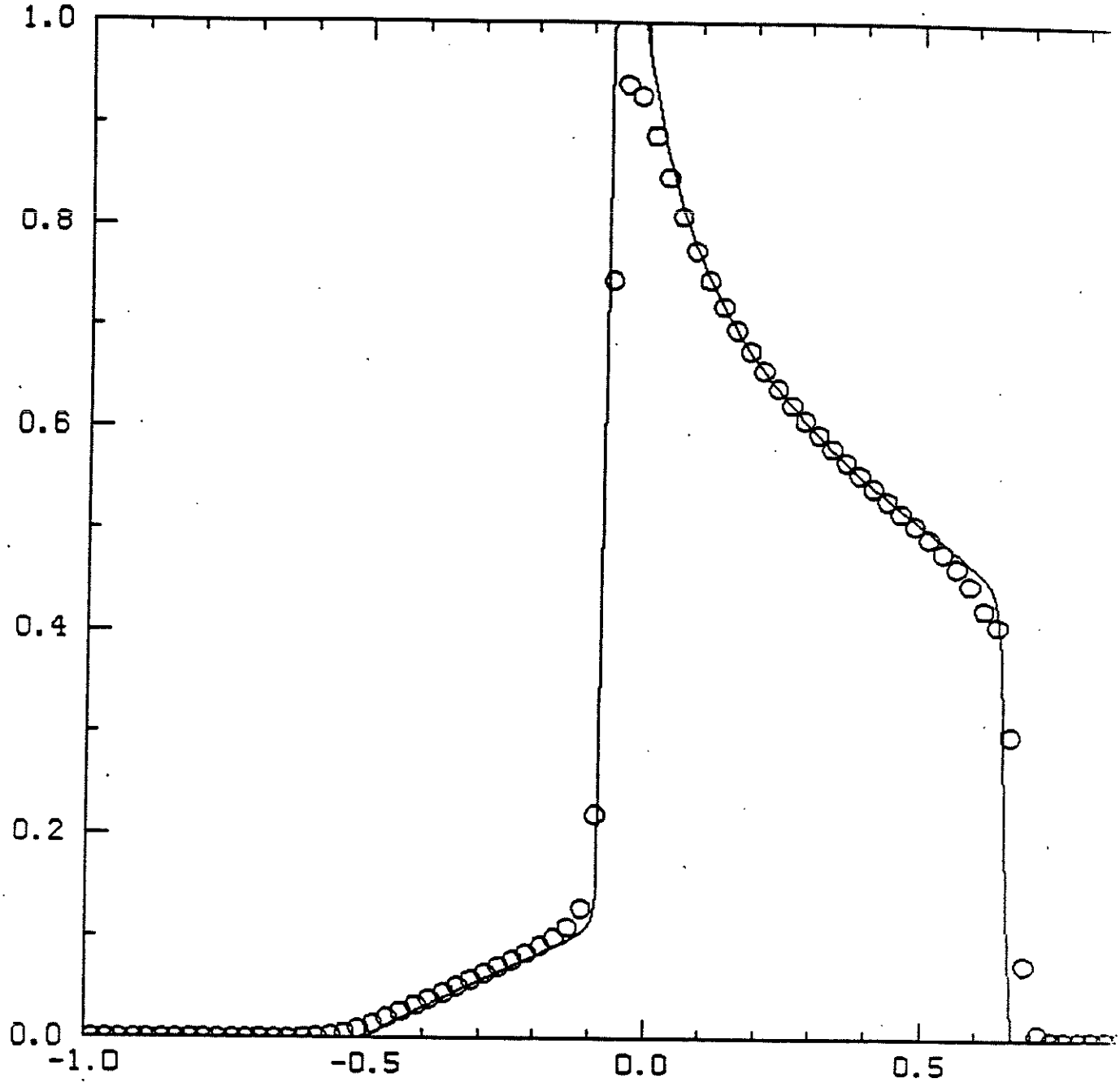

Figure 35 : 3-3-LF-ENO , $\Delta x = \frac{L}{40}$

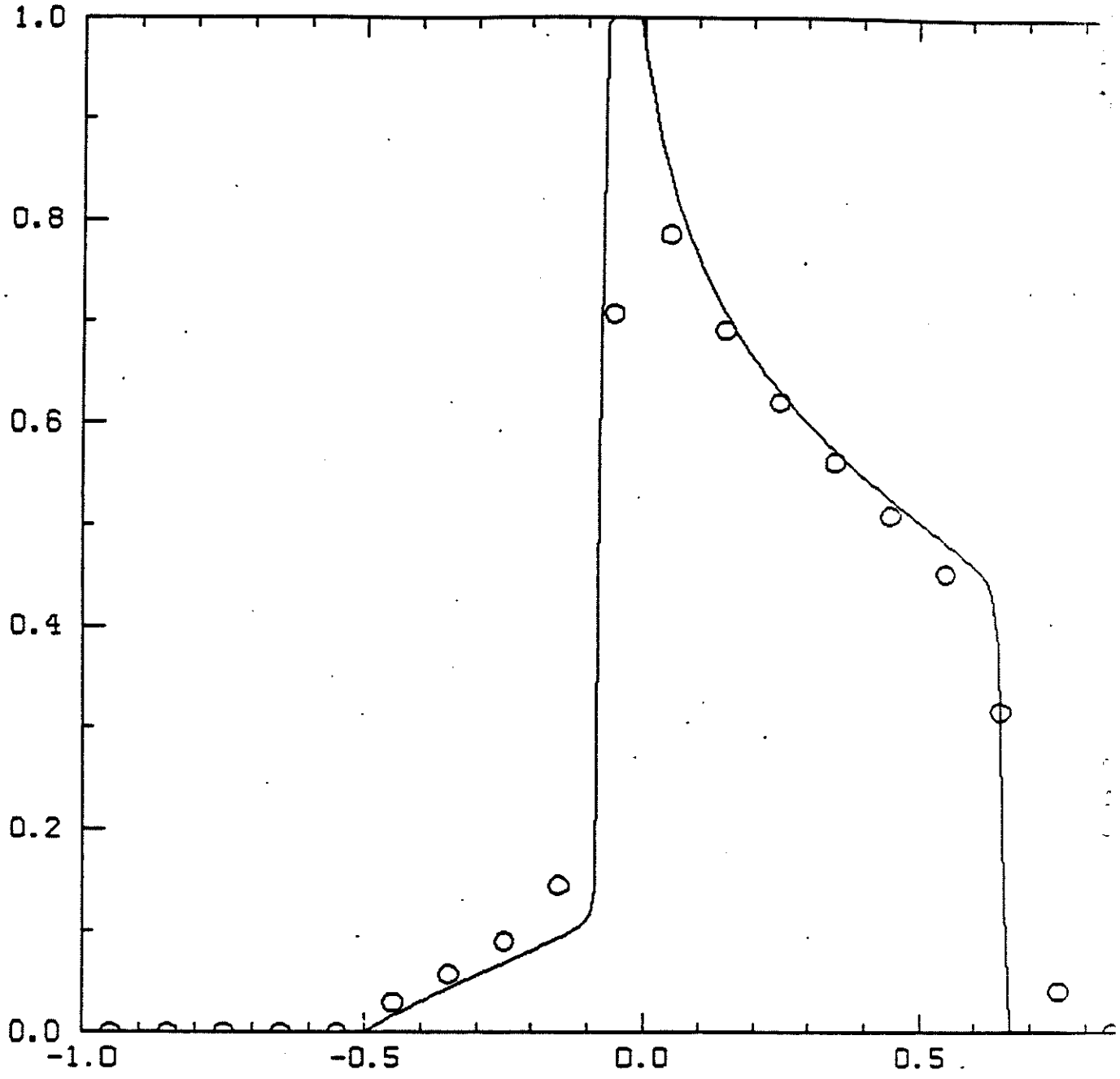

Figure 36 : 3-3-EO-ENO, $\Delta x = \frac{1}{10}$

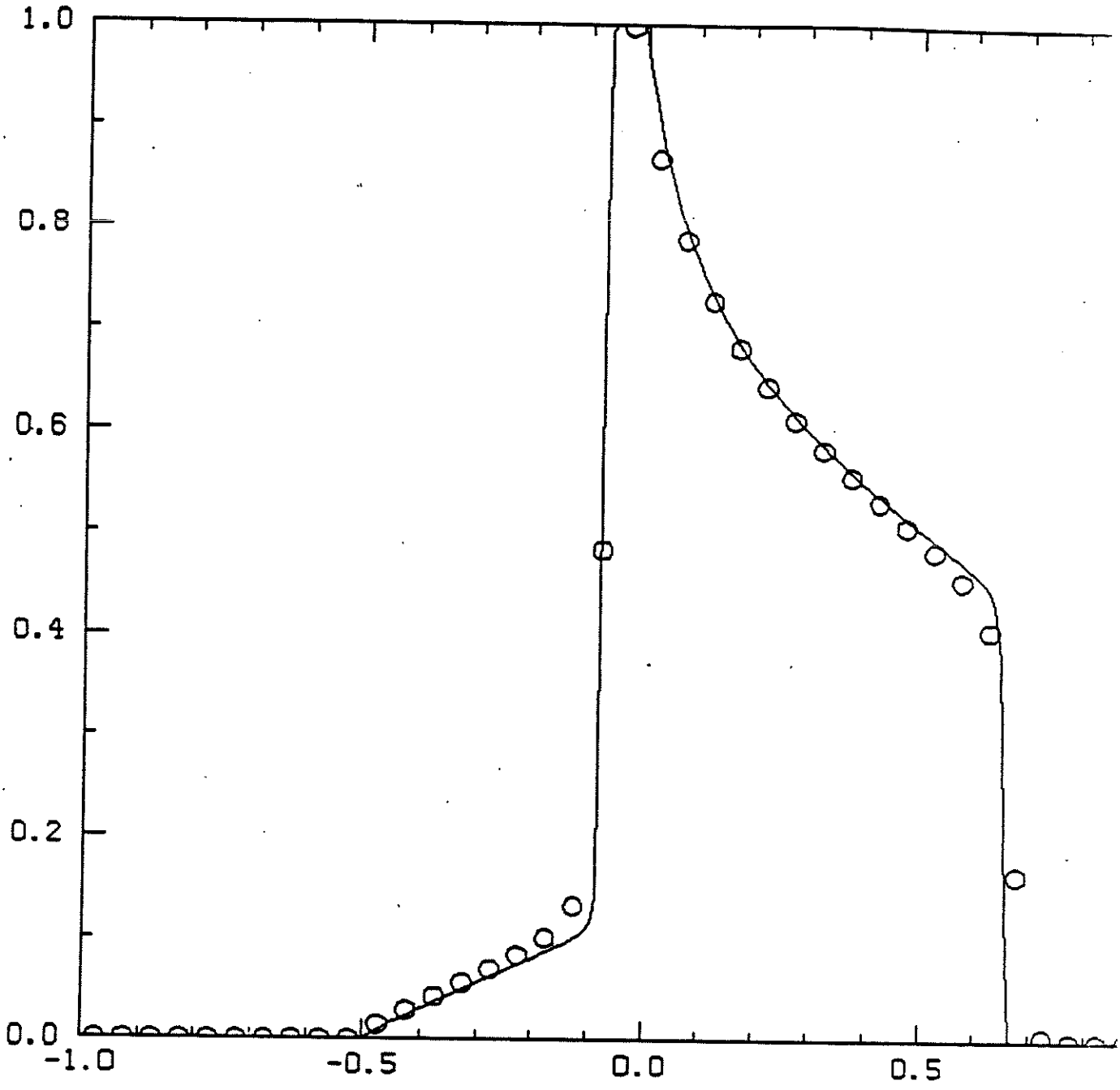

Figure 37 : 3-3 - EO - ENO , $\Delta x = \frac{1}{20}$

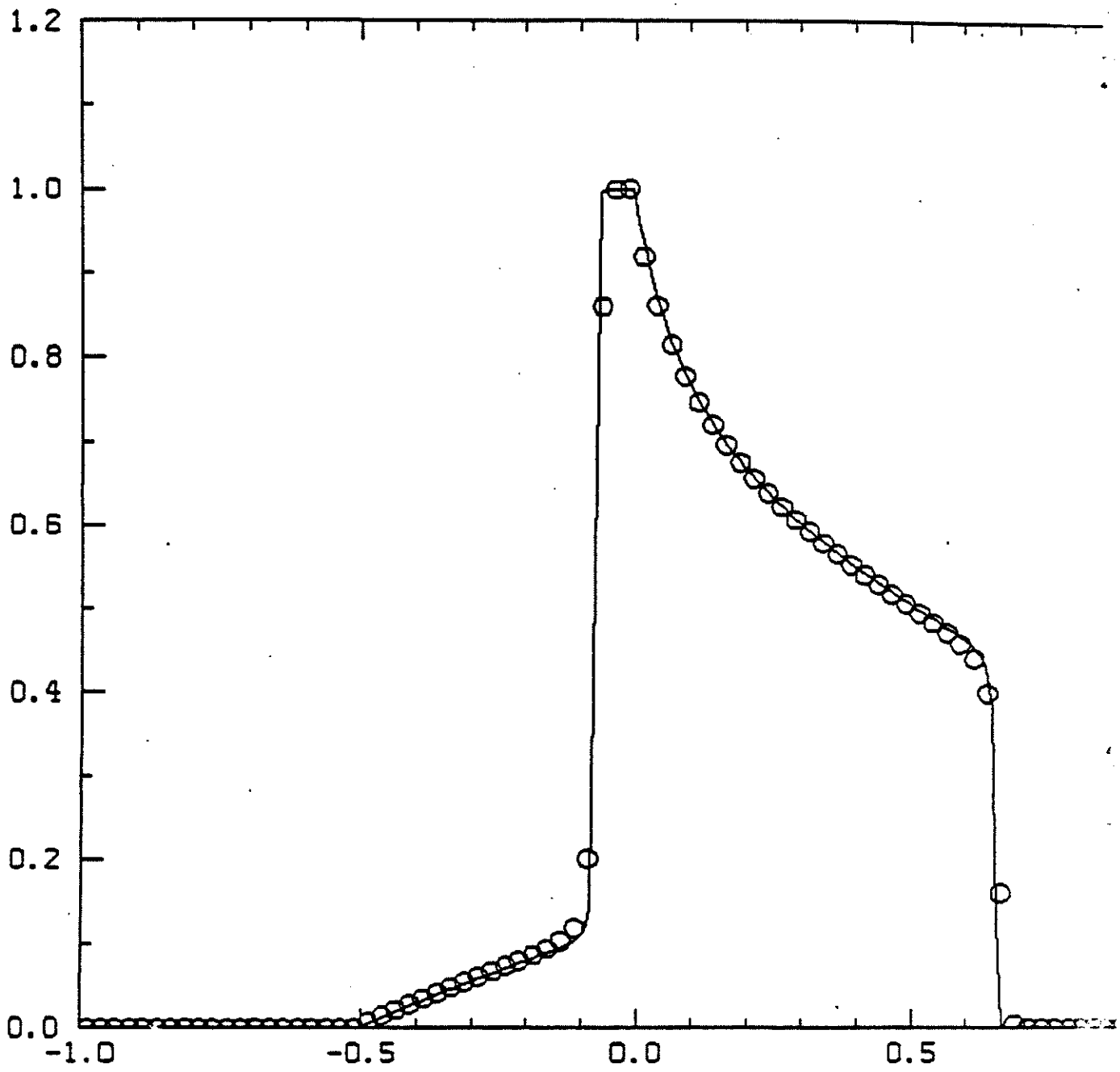

Figure 38 : 3 - 3 - EO - ENO , $\Delta x = \frac{1}{40}$