A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws I: Scalar Equations

Tariq D. Aslam*
e-mail: aslam@lanl.gov

June 8, 1998

Abstract

A level set algorithm for tracking discontinuities in hyperbolic conservation laws is presented. The algorithm uses a simple finite difference approach, analogous to the method of lines scheme presented in [20]. The zero of a level set function is used to specify the location of the discontinuity. Since a level set function is used to describe the front location, no extra data structures are needed to keep track of the location of the discontinuity. Also, two solution states are used at all computational nodes, one corresponding to the "real" state, and one corresponding to a "ghost node" state, analogous to the "Ghost Fluid Method" of [6]. High order pointwise convergence is demonstrated for linear and nonlinear conservation laws, even at discontinuities and in multiple dimensions. The solutions are compared to standard high order shock capturing schemes. This paper focuses on scalar conservation laws. Level set tracking for systems of conservation laws in multi-dimensions will be presented in future work [2].

*Los Alamos National Laboratory, Los Alamos, NM 87545 - performed under the auspices of the U.S. Department of Energy

LA-UR-98-2194

Title: A LEVEL SET ALGORITHM FOR TRACKING DISCONTINUITIES IN HYPERBOLIC CONSERVATION LAWS I: SCALAR EQUATIONS

Author(s): TARIQ D. ASLAM

Submitted to: JOURNAL OF COMPUTATIONAL PHYSICS

1 Introduction

While high order shock capturing schemes have proven to be invaluable tools in solving hyperbolic conservation laws, they generally do not converge at high order in the presence of discontinuities. Typically, there will be a few "intermediate" points within a numerical shock profile. The state at these "intermediate" points are in error by an O(1) amount, causing L_1 convergence to be at best first order accurate, and L_∞ convergence to be zeroth order accurate. These errors are even worse in the case of a linear discontinuity (usually less than first order in the L_1 norm), since the the number of intermediate points typically grows with time (an exception would be a scheme that employs artificial compression, for example [23]).

For scalar equations, high order rates of convergence can be achieved if one measures convergence at a finite distance away from the location of discontinuities, for examples see [20] [14]. This is due to the fact that the characteristics of a scalar hyperbolic equation will either run in parallel to the discontinuity (linear) or travel into the discontinuity (nonlinear). So, it is expected that O(1) errors near a discontinuity will not effect the solution a finite distance away from the discontinuity under mesh refinement.

For nonlinear systems of conservation laws, there will be information that passes through a shock wave, and typically there will be a loss of high order convergence, even *away* from discontinuities. These errors also show up as "noise" in the non-shock characteristic fields. This is observed in slowly moving shock waves [18] [17] [1]. This "noise" will typically limit the rate of convergence, even a finite distance away from shock waves. As pointed out in [1], there may be no way of eliminating this noise without using subcell resolution or tracking [3] [12] [7].

This paper is devoted to presenting a level set technique [15] for tracking linear and nonlinear discontinuities for scalar conservation laws. The algorithm is based on a method of lines, finite difference framework. Since a level set formulation is used to denote the location of the discontinuity, there is no extra front logic required. All variables are located numerically on a uniform Cartesian grid, and are updated with standard method of lines, essentially non-oscillatory (ENO) schemes.

The O(1) errors introduced by most schemes can be attributed to interpolating across a discontinuity. It will be shown that the present algorithm is similar to Harten's subcell resolution technique [7], in that this algorithm does not interpolate across a discontinuity, and the difference algorithm always "sees" a continuous solution, even near discontinuities. Harten achieves this by replacing a standard interpolation scheme, in cells that contain a discontinuity, with one that extrapolates in a conservative manner from smooth regions of the flow. Although the subcell method works quite well in one dimension, there seems to be no easy way to extend it to multidimensional problems, or to problems involving nonlinear discontinuities (shocks) [20]. There are four main differences between Harten's method and the present work: 1) Linear and nonlinear discontinuities can be treated, 2) Multidimensional problems are relatively straightforward, 3) Conservation is achieved under mesh refinement. The scheme has conservation errors of the same order as the truncation errors, but still converges to the proper weak solution. 4) The zero of a level set function is used to specify the location of the discontinuity, thus one must know the initial location of any discontinuities to initialize the level set function; no discontinuity detectors are used, and any shocks that form later in time, away from the zero of the level set function, will be captured.

The outline of the paper is as follows. The mathematical formulation is presented in Section 2. Section 3 describes the fifth order numerical implementation. In Section 4, examples are presented to demonstrate the high rates of convergence that are achieved by the level set tracking method, even near discontinuities. And finally, Section 5 discusses future aspects of this method with respect to multidimensional systems of conservation laws.

2 Mathematical Formulation

2.1 Shock jump condition

Here, we present the mathematical formulation of the level set tracking method for scalar conservation laws. We wish to solve the scalar conservation law

$$u_{,t} + f(u)_{,x} = 0. \tag{1}$$

where $_{,t}$ and $_{,x}$ denote partial derivatives with respect to x and t. Equation (1) is hyperbolic, and admits discontinuous solutions. Depending on the form of $f(u)$, these discontinuities may be linear or nonlinear. Of particular interest is the formulation of a numerical method that treats the propagation of these discontinuities accurately. Shock waves, where u is discontinuous, are of greatest importance. Other discontinuities, such as derivatives of u in the case of rarefaction corners, are also important, but shock capturing schemes already typically converge at second order in the L_1 norm and first order in the L_∞ norm in these cases (an exception would be the creation of a self similar rarefaction wave, see [4] [19]). Here, a method for dealing with linear and nonlinear discontinuities in u is presented.

It is well known that a shock will travel at a speed that depends, in general, on the value of u on both sides of the discontinuity as well as the flux function, $f(u)$. For a scalar conservation law, the shock speed is given by:

$$s = \frac{[f(u)]}{[u]} = \frac{f(u_r) - f(u_l)}{u_r - u_l}, \tag{2}$$

where u_r is the value of u just to the right of the shock wave, and u_l is the value of u just to the left of the shock. While equation (2) guarantees conservation, it doesn't necessarily guarantee that the weak solution will be the proper viscosity limiting solution. For the solution to be a shock wave, the following entropy condition must also be satisfied:

$$f'(u_l) \geq s \geq f'(u_r) \tag{3}$$

Note, the standard entropy condition, $f'(u_l) > s > f'(u_r)$, doesn't admit discontinuities in linear equations, whereas equation (3) does, see [11]. If equation (3) is not satisfied, the solution will be a rarefaction wave. Again, the focus of this paper is to deal with shock waves, since they introduce the largest numerical errors. Nonconvex flux functions have a slightly more complicated entropy condition, since both shocks and rarefactions can originate from a single discontinuity, see [11] and the references therein. Issues relating to nonconvex flux functions are not addressed here.

2.2 Representation

The key idea to avoiding discretizing across discontinuities is to have two solution states, u_1 and u_2, at all locations, each of which is continuous across the discontinuity. Note that this approach is similar to the "Ghost Fluid Method" of [6], where a level set function, ψ, is used to determine the location of the discontinuity. The real solution state at a point is selected to be u_1 or u_2, depending on the sign of the level set function at that location.

Denoting the real solution state as u, we have

$$u = \begin{cases} u_1, & \text{if } \psi > 0 \\ u_2, & \text{if } \psi \leq 0 \end{cases} \tag{4}$$

and ψ is the continuous level set function, whose zero is located at the discontinuity. The ghost state, u_g, is given by

$$u_g = \begin{cases} u_1, & \text{if } \psi \leq 0 \\ u_2, & \text{if } \psi > 0 \end{cases} \tag{5}$$

Since this algorithm works with the variables u_1, u_2 and ψ, and never uses u directly, all variables are typically continuous. To make this clear, consider representing the following discontinuous function:

$$u = \begin{cases} \cos(x), & \text{if } x \leq 0 \\ \sin(x), & \text{if } x > 0 \end{cases} \tag{6}$$

This can be represented by the following u_1, u_2 and ψ:

$$u_1 = \sin(x) \tag{7}$$

$$u_2 = \cos(x) \tag{8}$$

$$\psi = x \tag{9}$$

Importantly, even though u is discontinuous, u_1, u_2 and ψ are all continuous. Clearly, discretizing continuous functions will yield higher order convergence than using the discontinuous function directly.

2.3 Solution

Here, we describe how to solve equation (1) using u_1, u_2 and ψ. For initial conditions, define u_1, u_2 and ψ such that equation (4) is satisfied at the initial time. Since u_1 and u_2 can take on any value for $\psi \leq 0$ and $\psi > 0$ respectively, and still satisfy equation equation (4), this representation is not unique. It will be advantageous to extend u_1 and u_2 smoothly into their respective "ghost node regions." The nonuniqueness also extends to the level set function. Any level set function, ψ, whose zero corresponds to the discontinuity is adequate. Here, the signed distance function is usually used to initialize $\psi(t=0)$.

Clearly, when $\psi > 0$, we need to solve

$$u_{1,t} + f(u_1)_{,x} = 0, \tag{10}$$

since $u = u_1$ when $\psi > 0$. Likewise, when $\psi \leq 0$, we need to solve:

$$u_{2,t} + f(u_2)_{,x} = 0. \tag{11}$$

For smoothness, it is desirable to solve equations (10) and (11) everywhere. But, since u_2 (u_1) will be considered a shocked state of u_1 (u_2) when $\psi > 0$ $(\psi \leq 0)$, we have to check to see if u_2 (u_1) satisfies the shock entropy condition equation (3). Appropriate left and right states in the shock entropy condition can be cast in terms of u_1, u_2 and ψ by:

$$u_l = \begin{cases} u_1, & \text{if } \psi_{,x} \leq 0 \\ u_2, & \text{if } \psi_{,x} > 0 \end{cases} \tag{12}$$

$$u_r = \begin{cases} u_1, & \text{if } \psi_{,x} > 0 \\ u_2, & \text{if } \psi_{,x} \leq 0 \end{cases} \tag{13}$$

And the shock entropy condition can be recast (for a convex flux function) as:

$$f'(u_l) \geq f'(u_r) \tag{14}$$

If $\psi > 0$ ($\psi \leq 0$) and the state u_2 (u_1) does not satisfy the shock entropy condition equation (14), we set $u_2 = u_1$ ($u_1 = u_2$). This will only affect the smoothness of u_1 or u_2 in their ghost node states, and will locally reduce to a shock capturing scheme. This will only affect the accuracy of solution when an initial discontinuity will form a self-similar rarefaction, and not a shock.

Once entropy satisfying states u_1 and u_2 have been given, we define the shock speed by equation (2). This shock speed function, s, will be determined everywhere, and this speed is used in the level set equation to propagate the level set function, ψ:

$$\psi_{,t} + s\psi_{,x} = 0. \tag{15}$$

Importantly, at $\psi = 0$, s will be the proper shock speed for the discontinuity in u. Also, if u_1 and u_2 are both smooth functions near $\psi = 0$, then s will also be smooth. This is important numerically, since accuracy will be lost if s is not smooth near $\psi = 0$.

So, in summary, we initialize u_1 and u_2 with smooth functions, and set ψ to be the signed distance function from the discontinuity location (or some other smooth function whose zero corresponds to the initial location of the discontinuity). This satisfies equation (4) at the initial time. Then, making sure that the ghost node states satisfy the shock entropy condition (for all time), solve equations (10), (11) and (15) everywhere. The real solution, u, can be recovered easily by using equation (4) anytime the solution state is required.

2.4 Justification

Clearly, the algorithm should work for a linear problem, where the characteristics are parallel to any discontinuity, and are also state independent. It is not obvious that the method will work for nonlinear problems, where the shock speed is determined by both states, u_1 and u_2. In particular, how does the algorithm prevent seemingly arbitrary initial conditions in the ghost state from polluting the real state? The answer lies in applying the shock entropy condition. We consider specific examples next. The arguments presented here are not formal proofs, but simple reasons why the method should work.

Consider equation (1) with a convex flux function, where $f'(u)$ is a monotonically increasing function of u. Also, take $\psi(x)$ to be a decreasing function of x. Assume that a single discontinuity exists at the initial time. Therefore, to the left of $\psi = 0$, $u = u_1$, and to the right of $\psi = 0$, $u = u_2$. So, there exists a ghost state for u_1 to the right of $\psi = 0$, and a ghost state for u_2 to the left of $\psi = 0$. In particular, we want to see if it is possible to pick a ghost node state that will yield a spurious solution. Since, for this example, we have $\psi_{,x} < 0$, we have the following shock entropy condition to the right of $\psi = 0$:

$$u_1 \geq u_2 \tag{16}$$

And, to the left of $\psi = 0$ we have:

$$u_2 \leq u_1 \tag{17}$$

Clearly, when $u_1 = u_2$ the problem reduces back to the shock capturing algorithm, and it is expected to achieve the proper solution. Furthermore, if $u_1 > u_2$ for $\psi < 0$, then any disturbance in the ghost state, u_1, will travel away from the discontinuity faster than the real state, u_2. The analogous result holds for $\psi > 0$. Therefore, the solution will not depend on what function is given in the ghost state, since that information will travel away from the true discontinuity at least as fast the true characteristic. Again, for numerical purposes, it is advantageous to extend the ghost state in a continuous fashion, so as not to lose numerical accuracy.

At worst, the above algorithm will reduce to a standard shock capturing algorithm. For example, examine what happens given the following initial conditions: $u_1(x, t = 0) = 0$, $u_2(x, t = 0) = 1$ and $\psi(x, t = 0) = -x$ (which corresponds to $u(x, t = 0) = H(x)$, where H is the Heaviside function, whose solution will be a self similar rarefaction fan.) For this example, both the u_1 ghost state for $\psi < 0$ and the u_2 ghost state for $\psi > 0$ don't satisfy the shock entropy condition, so they are initially set equal to each other in their ghost node region, and therefore $u_1(x, t = 0) = u_2(x, t = 0) = u(x, t = 0) = H(x)$. Clearly, this will result in the same numerical solution as discretizing u directly. If the ghost node states were not checked for entropy violation, the solution will be a shock, which moves at a speed according to equation (2), which satisfies conservation, but is an entropy violating solution.

Another example would be: $u(x, t = 0) = H(-x)$. Clearly, the real solution is a shock moving at speed $(f(1)-f(0))$. If we take $u_1(x, t = 0) = 1$, $u_2(x, t = 0) = 0$ and $\psi(x, t = 0) = -x$, then we obtain the proper solution. But what happens if we modify the u_1 ghost state? Let's examine what happens when $u_1(x > 0, t = 0) = -1$. This still satisfies equation (4), but doesn't satisfy (14). The solution to u_1 will have a shock traveling at speed $(f(1) - f(-1))/2$ which, for a convex flux function, will be slower than the true shock speed, $(f(1) - f(0))$. When this happens, information from the ghost region of u_1 will influence the solution, and the correct solution will not be achieved. Again, projecting the ghost values into entropy satisfying states seems to always give the proper viscosity limiting solution.

2.5 Extension to two dimensions

The above formulation extends easily to multidimensional problems. In particular, the scalar conservation law in two dimensions is:

$$u_{,t} + f(u)_{,x} + g(u)_{,y} = 0. \tag{18}$$

Again, equation (4) can be used to represent the solution. The states u_1 and u_2 are evolved according to

$$u_{1,t} + f(u_1)_{,x} + g(u_1)_{,y} = 0 \tag{19}$$

$$u_{2,t} + f(u_2)_{,x} + g(u_2)_{,y} = 0, \tag{20}$$

and the level set function, ψ, according to

$$\psi_{,t} + \vec{s} \cdot \vec{\nabla}\psi = 0, \tag{21}$$

where

$$\vec{s} = \frac{[f(u)]}{[u]}\hat{\imath} + \frac{[g(u)]}{[u]}\hat{\jmath}. \tag{22}$$

Again, one needs to make sure that the ghost node state satisfies the shock entropy condition. In multiple dimensions, for a convex flux function, it is sufficient to check if the characteristics flow into the shock. In two dimensions, the characteristic velocity is given by:

$$\vec{c} = f'(u)\hat{\imath} + g'(u)\hat{\jmath}. \tag{23}$$

Also, the orientation of a shock will be given by

$$\hat{n} = \frac{\vec{\nabla}\psi}{|\vec{\nabla}\psi|} \tag{24}$$

So, the characteristic speeds for the states u_1 and u_2 in the normal direction are given by

$$c_1 = \hat{n} \cdot (f'(u_1)\hat{\imath} + g'(u_1)\hat{\jmath}), \tag{25}$$

$$c_2 = \hat{n} \cdot (f'(u_2)\hat{\imath} + g'(u_2)\hat{\jmath}). \tag{26}$$

For the states u_1 and u_2 to satisfy the shock entropy condition, the following must be true:

$$c_2 \geq c_1 \tag{27}$$

In one dimension, equation (27) is equivalent to equation (3).

3 Discretization

Here, one particular discretization is presented. Numerical results are given in the next section. Recall that equations (10), (11), (15) along with the shock entropy condition need be discretized. Notice equations (10) and (11) are scalar conservation laws, equation (15) is a Hamilton-Jacobi-like partial differential equation, and the entropy condition is an algebraic constraint on the ghost state.

3.1 Grid

A uniform Cartesian grid is used to discretize the domain $x \in (x_{min}, x_{max})$, with $N_x + 1$ equally spaced nodes. The numerical solution of u_1 is denoted by $u_1(i,n)$, where i is the spatial node number corresponding to the location $x_i = x_{min} + i\Delta x$, where $\Delta x = (x_{max} - x_{min})/N_x$. And n is the time level corresponding to $t_n = n\Delta t$, where $\Delta t = t_{final}/N_t$. The states u_2 and ψ are denoted similarly.

3.2 Time integration

An explicit method of lines approach is taken. The results presented here use a third order TVD Runge-Kutta time integrator [20]. For example, the solution to $u_1(i,n)$ will be advanced from $t = t_n$ to $t = t_{n+1}$ via:

$$
\begin{aligned}
u_1(i,*) &= u_1(i,n) + \Delta t L(u_1(i,n)) \\
u_1(i,**) &= \frac{3}{4}u_1(i,n) + \frac{1}{4}\Delta t L(u_1(i,*)) \\
u_1(i,n+1) &= \frac{1}{3}u_1(i,n) + \frac{2}{3}u_1(i,**) + \frac{2}{3}\Delta t L(u_1(i,**)).
\end{aligned}
\tag{28}
$$

Here, the $*$ and $**$ represent intermediate stages of the Runge-Kutta integration. The updates for u_2 and ψ are similar. Note that the $L()$ operator corresponds to the spatial flux differences for u_1 and u_2, or derivatives of ψ. These are described in the next section.

3.3 Spatial discretization

3.3.1 Project ghost nodes into entropy satisfying state

As described in section 2.3, it is necessary to make sure that the ghost node state be an entropy satisfying state. This is done at the beginning of every Runge-Kutta cycle. As stated in section 2.3, the ghost node state is only set to the real state if equation (14) in one dimension, or (27) in two dimensions is not satisfied. If the entropy condition is satisfied, then the ghost node state is not modified. The entropy condition is only an algebraic constraint on the ghost node state, but requires knowledge of $\psi_{,x}$ (and $\psi_{,y}$ in two dimensions). These spatial derivatives are obtained by averaging the two choices in equation (50)

3.3.2 Conservative discretization for u_1 and u_2

For $L(u_1(i,n))$ and $L(u_2(i,n))$ we use the fifth order weighted ENO (WENO) scheme of [10], with a local Lax-Friedrichs solver. This scheme is a conservative flux difference method, which has been shown to be stable, and yield the proper viscosity vanishing solution to equation (1). The operator $L(u_1(i,n))$ is given by:

$$
L(u_1(i,n)) = -(\hat{f}_{i+1/2} - \hat{f}_{i-1/2})/\Delta x \tag{29}
$$

where $\hat{f}_{i+1/2}$ and $\hat{f}_{i-1/2}$ are numerical approximations to the flux function, $f(u)$. In particular, for the local Lax-Friedrichs scheme, we take

$$
\hat{f}_{i+1/2} = \hat{f}_i^+ + \hat{f}_{i+1}^- \tag{30}
$$

where

$$\hat{f}_i^+ = WENO5(f_{i-2}^+, f_{i-1}^+, f_i^+, f_{i+1}^+, f_{i+2}^+) \tag{31}$$

$$\hat{f}_{i+1}^- = WENO5(f_{i+3}^-, f_{i+2}^-, f_{i+1}^-, f_i^-, f_{i-1}^-) \tag{32}$$

and

$$f_i^+ = \frac{1}{2}(f(u_1(i,n)) + \alpha u_1(i,n)) \tag{33}$$

$$f_i^- = \frac{1}{2}(f(u_1(i,n)) - \alpha u_1(i,n)) \tag{34}$$

and

$$\alpha = \max(|f'(u_1(i,n))|, |f'(u_1(i+1,n))|). \tag{35}$$

The function $WENO5(a,b,c,d,e)$ is defined next. First, define three interpolated values:

$$q_1 = \frac{a}{3} - \frac{7b}{6} + \frac{11c}{6} \tag{36}$$

$$q_2 = -\frac{b}{6} + \frac{5c}{6} + \frac{d}{3} \tag{37}$$

$$q_3 = \frac{c}{3} + \frac{5d}{6} - \frac{e}{6} \tag{38}$$

and three indicators of smoothness:

$$IS_1 = 13(a-2b+c)^2 + 3(a-4b+3c)^2 \tag{39}$$

$$IS_2 = 13(b-2c+d)^2 + 3(d-b)^2 \tag{40}$$

$$IS_3 = 13(c-2d+e) + 3(3c-4d+e)^2 \tag{41}$$

and take

$$\alpha_1 = \frac{1}{(\epsilon + IS_1)^2} \tag{42}$$

$$\alpha_2 = \frac{6}{(\epsilon + IS_2)^2} \tag{43}$$

$$\alpha_3 = \frac{3}{(\epsilon + IS_3)^2} \tag{44}$$

and finally

$$WENO5(a,b,c,d,e) = \frac{\alpha_1 q_1 + \alpha_2 q_2 + \alpha_3 q_3}{\alpha_1 + \alpha_2 + \alpha_3}. \tag{45}$$

In all computations presented here, $\epsilon = 10^{-6}$, as suggested in [14], [10] and [9]. The operator for u_2 is the same, with u_2 replacing u_1. Note, for the linear advection equation, the local Lax-Friedrichs scheme is equivalent to the standard upwind discretization.

3.3.3 Level set discretization

For the level set equation (15), we have the operator

$$L(\psi(i,n)) = -s(i,n)\hat{\psi}_{,x} \tag{46}$$

Where $s(i,n)$ is the shock speed and $\hat{\psi}_{,x}$ is the numerical approximation to $\psi_{,x}$. First, the shock speed at each node location, $s(i,n)$, is determined from u_1 and u_2 as:

$$s(i,n) = \frac{f(u_1(i,n)) - f(u_2(i,n))}{u_1(i,n) - u_2(i,n)} \tag{47}$$

The $\psi_{,x}$ derivative is approximated using a fifth order WENO scheme for Hamilton-Jacobi equations [9]. Define the first order difference operators:

$$D_i^- = \frac{\psi(i,n) - \psi(i-1,n)}{\Delta x} \tag{48}$$

$$D_i^+ = \frac{\psi(i+1,n) - \psi(i,n)}{\Delta x}. \tag{49}$$

The numerical spatial derivative, $\hat{\psi}_{,x}$ is then given by:

$$\hat{\psi}_{,x} = \begin{cases} WENO5(D_{i-2}^-, D_{i-1}^-, D_i^-, D_{i+1}^-, D_{i+2}^-), & \text{if } s(i,n) \geq 0 \\ WENO5(D_{i+2}^+, D_{i+1}^+, D_i^+, D_{i-1}^+, D_{i-2}^+), & \text{otherwise.} \end{cases} \tag{50}$$

4 Examples

Here, we present solutions to various scalar hyperbolic conservation laws using both the level set tracking algorithm and traditional shock capturing algorithm. Notice that the discretization presented in section 3 will have a truncation error of O(Δt^3)+O(Δx^5) in smooth regions. All computations are performed with $\Delta t \propto \Delta x^{5/3}$, which effectively yields a truncation error of

$O(\Delta x^5)$ in smooth regions. For all cases, the number of timesteps, N_t, and number of spatial node, N_x, are noted in the Tables and Figures.

4.1 Linear Advection Equation

The algorithm presented in Sections 2 and 3 is tested on a standard test problem [8] [7] [21]. The equation to be solved is the linear advection equation:

$$u_{,t} + u_{,x} = 0 \tag{51}$$

with periodic boundary conditions at $x = \pm 1$ and subject to the initial conditions:

$$u = \begin{cases} 2(x+1) - \frac{1}{6}\sin(3\pi(x+\frac{3}{2})), & -1 < x < -\frac{1}{2} \\ -(x-\frac{1}{2})\sin(\frac{3}{2}\pi(x-\frac{1}{2})^2), & -\frac{1}{2} < x < \frac{1}{6} \\ \sin(2\pi(\frac{1}{2}-x)), & \frac{1}{6} < x < \frac{1}{2} \\ \sin(2\pi(x-\frac{1}{2})), & \frac{1}{2} < x < \frac{5}{6} \\ 2(x-1) - \frac{1}{6}\sin(3\pi(x-\frac{1}{2})), & \frac{5}{6} < x < 1. \end{cases} \tag{52}$$

Notice that this initial condition has discontinuities at $x = -\frac{1}{2}$, $x = \frac{1}{6}$ and $x = \frac{5}{6}$. Also, there is a discontinuity in derivative at $x = \frac{1}{2}$, see Figure 1. This function is represented by equation (4) with

$$u_1 = \begin{cases} 2(x+1) - \frac{1}{6}\sin(3\pi(x+\frac{3}{2})), & -1 < x < -\frac{1}{6} \\ \sin(2\pi(\frac{1}{2}-x)), & -\frac{1}{6} < x < \frac{2}{3} \\ 2(x-1) - \frac{1}{6}\sin(3\pi(x-\frac{1}{2})), & \frac{2}{3} < x < 1, \end{cases} \tag{53}$$

$$u_2 = \begin{cases} \sin(2\pi(x-\frac{1}{2})), & -1 < x < -\frac{5}{6} \\ -(x-\frac{1}{2})\sin(\frac{3}{2}\pi(x-\frac{1}{2})^2), & -\frac{5}{6} < x < \frac{1}{3} \\ \sin(2\pi(x-\frac{1}{2})), & \frac{1}{3} < x < 1, \end{cases} \tag{54}$$

$$\psi = \begin{cases} x + \frac{7}{6}, & -1 < x < -\frac{5}{6} \\ -x - \frac{1}{2}, & -\frac{5}{6} < x < -\frac{1}{6} \\ x - \frac{1}{6}, & -\frac{1}{6} < x < \frac{1}{3} \\ -x + \frac{1}{2}, & \frac{1}{3} < x < \frac{2}{3} \\ x - \frac{5}{6}, & \frac{2}{3} < x < 1. \end{cases} \tag{55}$$

Although u_1 and u_2 are also discontinuous and ψ is discontinuous in derivative, they are all C_∞ at the true discontinuity locations of u. And since most shock capturing schemes will be convergent away from the discontinuities (at least in a scalar problem), it is expected that using equation (4), and operating on u_1, u_2 and ψ will be more accurate than discretizing u directly. This is confirmed numerically. At $t = 2$ with N_t timesteps, the error in the

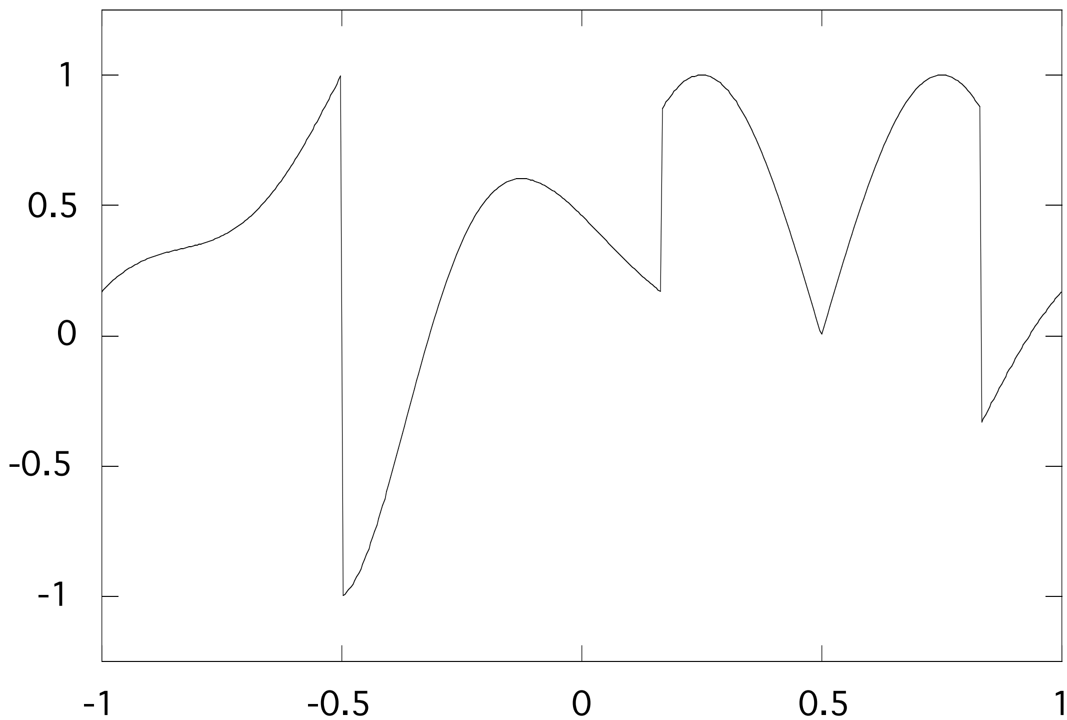

Figure 1: Plot of initial conditions corresponding to equation (52).

numerical solution using the discrete L_1 and L_∞ norms is measured. These norms measure the pointwise convergence of the numerical solution to the exact solution (all points in the numerical solution are included, not just the points away from discontinuities). The errors are denoted by E_1 and E_∞, and the order at which they converge are denoted by R_1 and R_∞, respectively. Also, a subscript LST indicates the level set tracking algorithm, and SC indicates the WENO5 shock capturing algorithm, see Table 1. Notice that the level set tracking algorithm converges at fifth order in both the L_1 and L_∞ norms. The standard shock capturing algorithm converges at roughly 5/6 order in the L_1 norm. This is commensurate with the notion that a captured linear discontinuity will smear at a rate proportional to $N_t^{1/(r+1)}$, where r is the order of the scheme [5]. The L_∞ error for the capturing was roughly constant, and equal to 1/2 the maximum jump in u, as expected. Figure 2 shows the solution of the WENO5 shock capturing algorithm, and Figure 3 shows the solution using the WENO5 level set tracking algorithm. Notice that this is the roughly the same resolution used in [7], with comparable results. Importantly, the level set tracking algorithm maintains a perfect discontinuity in u, while achieving high order pointwise convergence.

TABLE 1: Numerical accuracy for 1D linear advection.

N_x	N_t	E_{1-LST}	R_{1-LST}	$E_{\infty-LST}$	$R_{\infty-LST}$	E_{1-SC}	R_{1-SC}
61	75	7.24e-3		3.46e-2		2.37e-1	
121	235	3.32e-4	4.52	1.64e-3	4.40	1.18e-1	1.01
241	740	1.04e-5	4.99	6.58e-5	4.64	6.31e-2	0.90
481	2340	2.91e-7	5.16	2.90e-6	4.51	3.47e-2	0.86
961	7425	9.58e-9	4.93	1.54e-7	4.23	1.93e-2	0.85
1921	23555	2.36e-10	5.34	3.51e-9	5.46	1.08e-2	0.84

4.2 Burgers' Equation

Here, the level set tracking algorithm is tested on the nonlinear Burgers' equation:

$$u_{,t} + (\frac{u^2}{2})_{,x} = 0 \tag{56}$$

with periodic boundary conditions at $x = 0$ and $x = 1$ subject to the initial conditions:

$$u = \begin{cases} \frac{1}{2}(1+\cos(2\pi x)), & \frac{1}{3} < x < \frac{2}{3} \\ \frac{1}{2} + \sin(2\pi x), & \text{otherwise} \end{cases} \tag{57}$$

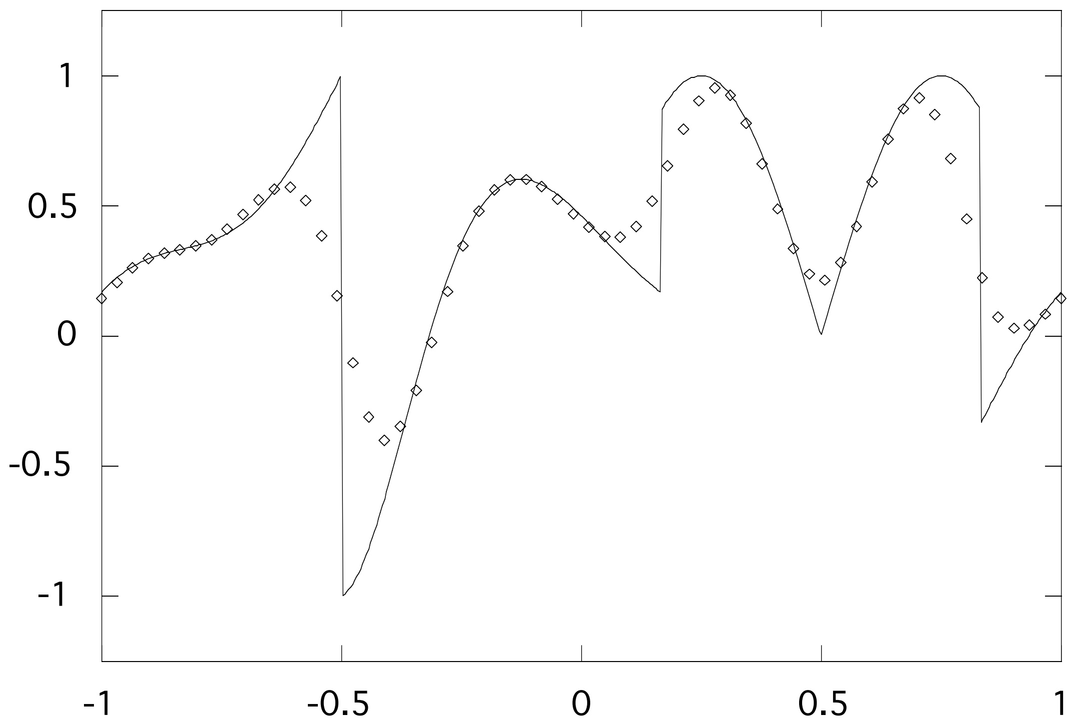

Figure 2: Plot of WENO5 shock capturing solution at $t = 2$ with $N_x = 61$, $\diamond$, and exact solution, solid line.

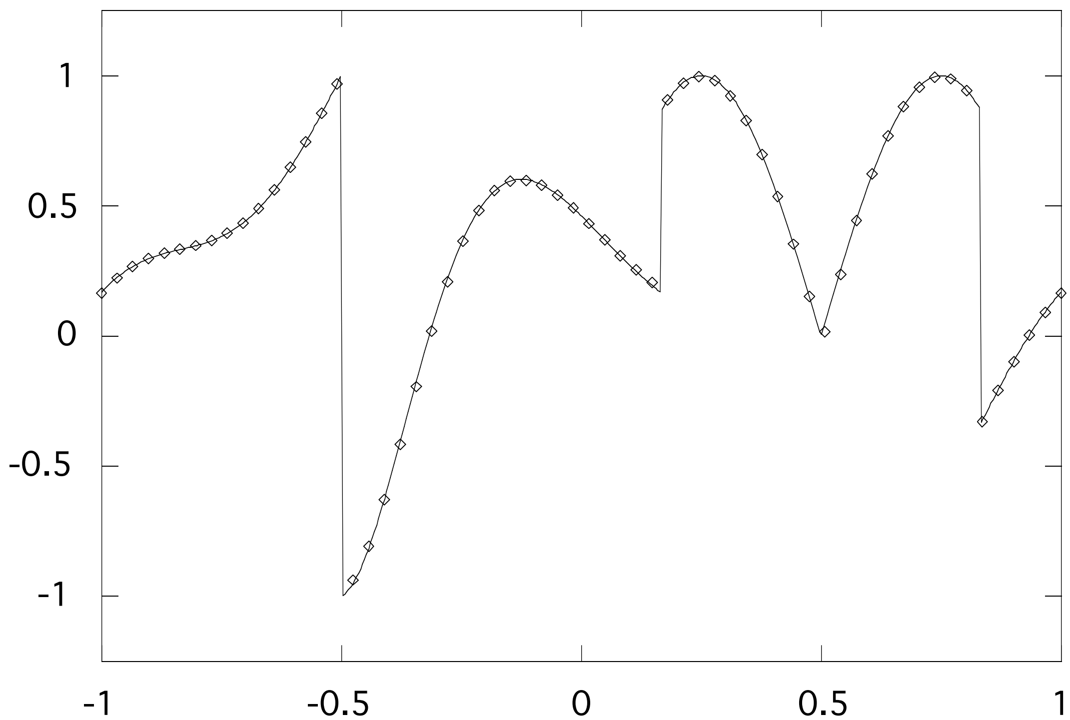

Figure 3: Plot of WENO5 level set tracking solution at $t = 2$ with $N_x = 61$, $\diamond$, and exact solution, solid line.

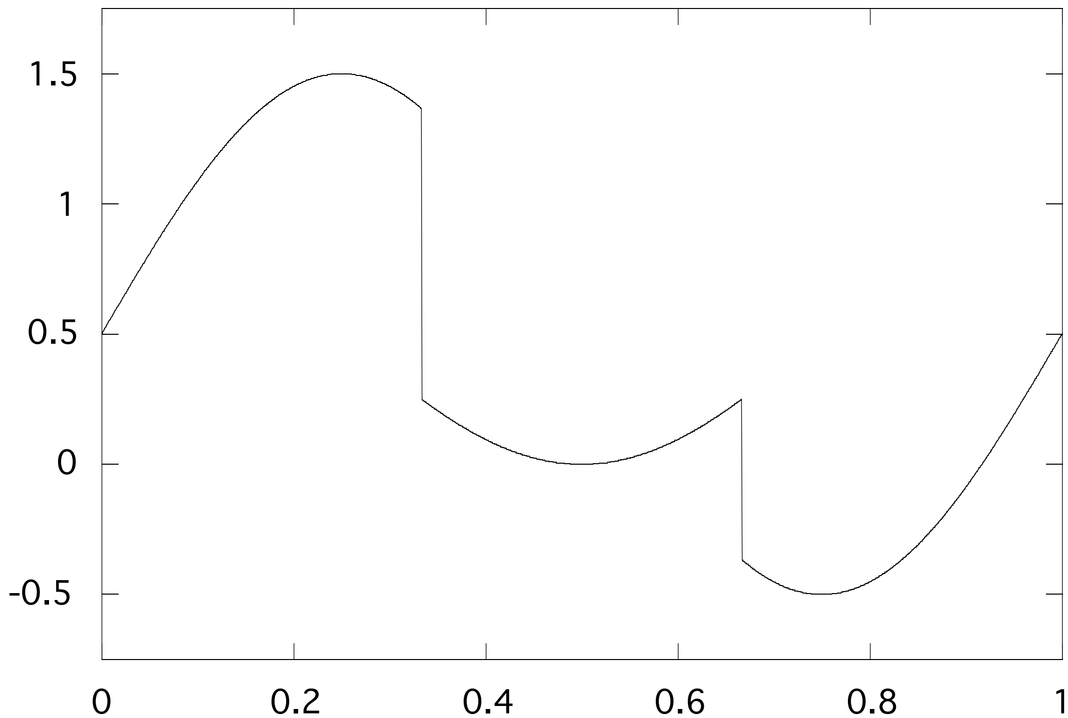

Figure 4: Plot of initial conditions corresponding to equation (57).

Notice that this initial condition has discontinuities at $x = \frac{1}{3}$ and $x = \frac{2}{3}$, see Figure 4. This function is represented by equation (4) with

$$u_1 = \frac{1}{2} + \sin(2\pi x), \tag{58}$$

$$u_2 = \frac{1}{2}(1 + \cos(2\pi x)), \tag{59}$$

$$\psi = \begin{cases} \frac{1}{3} - x, & 0 \leq x < \frac{1}{2} \\ x - \frac{2}{3}, & \frac{1}{2} \leq x < 1 \end{cases} \tag{60}$$

Again, u_1 and u_2 are also discontinuous and ψ is discontinuous in derivative, but they are all C_∞ at the true discontinuity locations of u. The L_1 and

L_∞ errors and rates of convergence, R_1 and R_∞, are measured at $t = 0.2$. These are measured by comparing with a numerical solution using twice as fine of a grid, thus we measure the pointwise self convergence of the solution. Fifth order convergence in both the L_1 and L_∞ norm is achieved for the level set tracking algorithm, while first order is achieved in the L_1 norm for the capturing scheme, see Table 2. Line plots at $t = 0.2$ are shown for both the capturing scheme, Fig. 5, and the level set tracking scheme, Fig. 6.

It is interesting to run the problem further in time, since at roughly $t \approx 0.369$ the two shocks collide. At this time, the $\psi < 0$ region disappears, and the algorithm then captures the remaining shock. Even in this case, the level set tracking scheme achieves the correct solution, but after $t \approx 0.369$, it reduces back to a capturing scheme. It would be interesting to see if using two level sets, and three solution states could accurately track the merging of two tracked shocks into one.

TABLE 2: Numerical accuracy for Burgers' equation.

N_x	N_t	E_{1-LST}	R_{1-LST}	$E_{\infty-LST}$	$R_{\infty-LST}$	E_{1-SC}	R_{1-SC}
40	15	1.84e-4		1.09e-3		1.48e-2	
80	50	8.16e-6	4.49	6.70e-5	4.03	7.84e-3	0.92
160	150	1.67e-7	5.61	9.15e-6	2.87	3.43e-3	1.19
320	480	9.40e-9	4.15	6.53e-7	3.81	1.47e-3	1.22
640	1525	2.56e-10	5.20	2.48e-8	4.72	7.63e-4	0.95
1280	4840	4.72e-12	5.76	2.70e-10	6.52	2.76e-4	1.47

4.3 Two Dimensional Linear Advection Equation

Here, a two dimensional test is conducted. The equation to be solved is the two dimensional linear advection equation:

$$u_{,t} + u_{,x} + u_{,y} = 0 \tag{61}$$

with periodic boundary conditions at $x = 0$, $x = 1$, $y = 0$ and $y = 1$ and subject to the initial conditions:

$$u = \begin{cases} \cos(2\pi x) + \cos(2\pi y) + 2, & \text{if } ((x - 1/2)^2 + (y - 1/2)^2)^{1/2} < 1/3 \\ \cos(2\pi x) + \cos(2\pi y) - 2, & \text{otherwise} \end{cases} \tag{62}$$

Notice that this initial condition has a discontinuity along a circle of radius, $R = 1/3$, whose center is located at $(x, y) = (1/2, 1/2)$. This function is represented by equation (4) with

$$u_1 = \cos(2\pi x) + \cos(2\pi y) + 2 \tag{63}$$

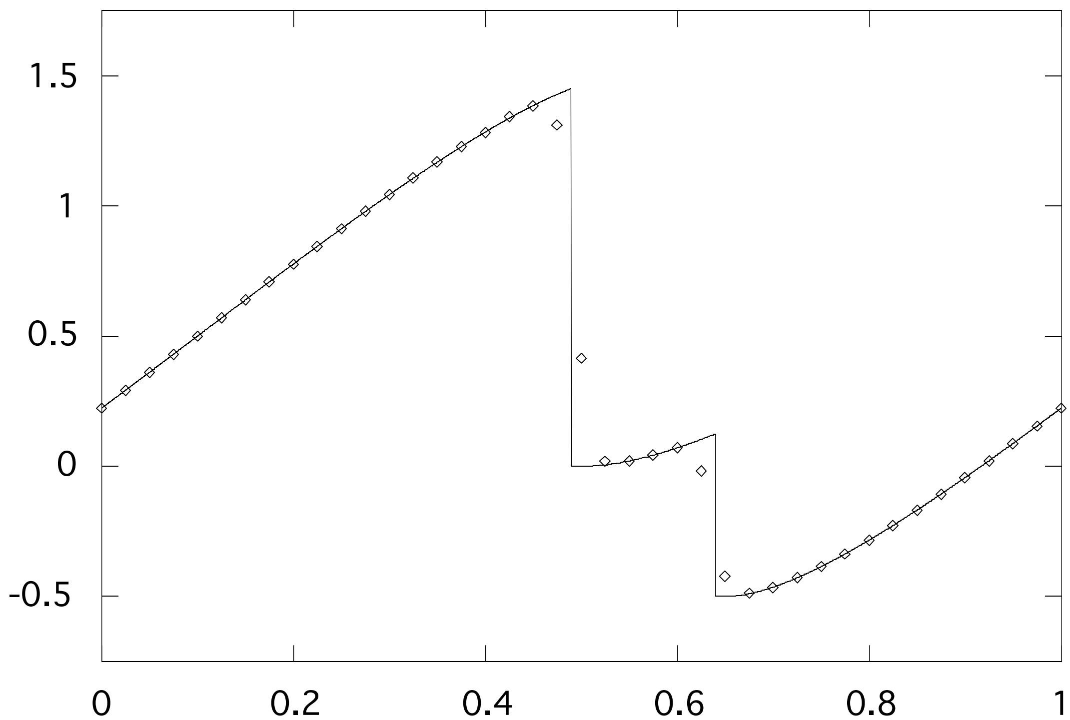

Figure 5: Plot of WENO5 shock capturing solution at $t = 0.2$ with $N_x = 40$, $\diamond$, and converged level set tracking solution ($N_x = 2560$), solid line.

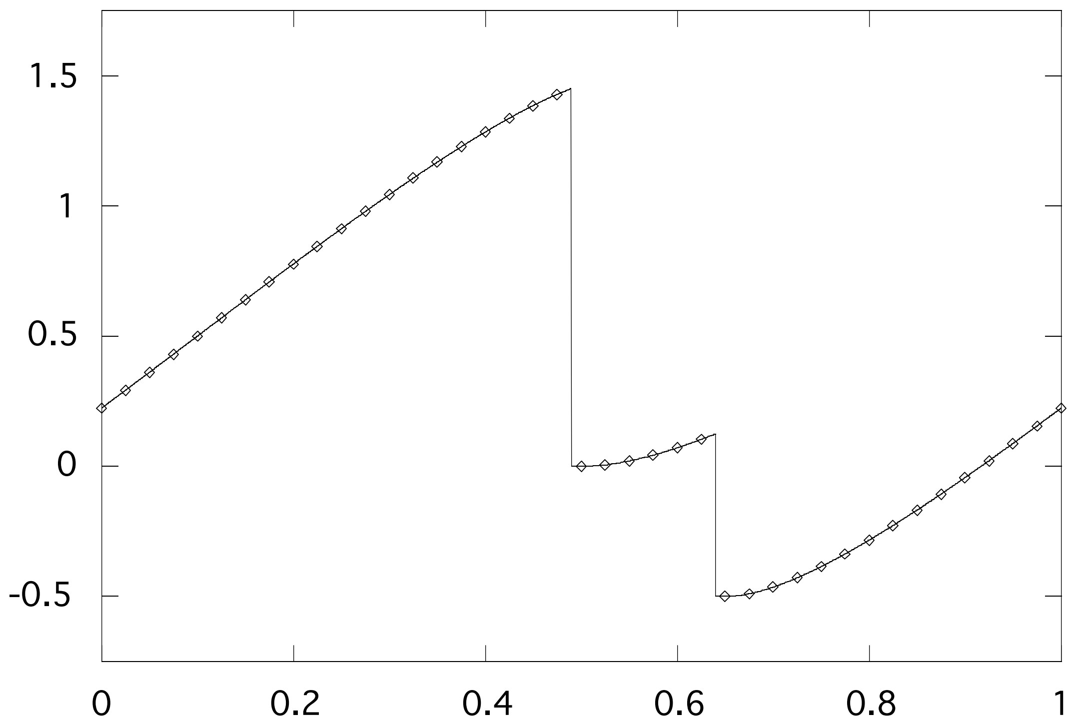

Figure 6: Plot of WENO5 level set tracking solution at $t = 0.2$ with $N_x = 40$, $\diamond$, and converged level set tracking solution ($N_x = 2560$), solid line.

$$u_2 = \cos(2\pi x) + \cos(2\pi y) - 2 \tag{64}$$

$$\psi = \frac{1}{3} - ((x-1/2)^2 + (y-1/2)^2)^{1/2} \tag{65}$$

Since this problem is linear, the level set equation will become:

$$\psi_{,t} + \psi_{,x} + \psi_{,y} = 0. \tag{66}$$

And also, there is no need to check for an entropy satisfying ghost node state. The y derivatives are calculated in an analogous fashion as the x derivatives. At $t = 1$ with N_t timesteps, the error in the numerical solution using the discrete L_1 and L_∞ norms is measured, see Table 3. Notice that the level set tracking algorithm converges at fifth order in both the L_1 and L_∞ norms. Again, for this linear problem, the standard shock capturing algorithm converges at roughly 5/6 order in the L_1 norm. Figure 7 shows a surface plot of the solution of the WENO5 shock capturing algorithm, and Figure 8 shows the solution using the WENO5 level set tracking algorithm. Figure 9 show a line plot of the shock capturing solution at $t = 1$ at $y = 1/2$. Figure 10 shows a line plot of the level set tracking solution at $t = 1$ at $y = 1/2$.

TABLE 3: Numerical Accuracy for 2D linear advection.

$N_x = N_y$	N_t	E_{1-LST}	R_{1-LST}	$E_{\infty-LST}$	$R_{\infty-LST}$	E_{1-SC}	R_{1-SC}
20	50	2.43e-3		5.99e-3		4.78e-1	
40	160	7.50e-5	5.01	2.08e-4	4.85	2.70e-1	0.83
80	500	2.30e-6	5.03	6.59e-6	4.98	1.52e-1	0.82
160	1600	7.00e-8	5.04	2.05e-7	5.01	8.54e-2	0.83

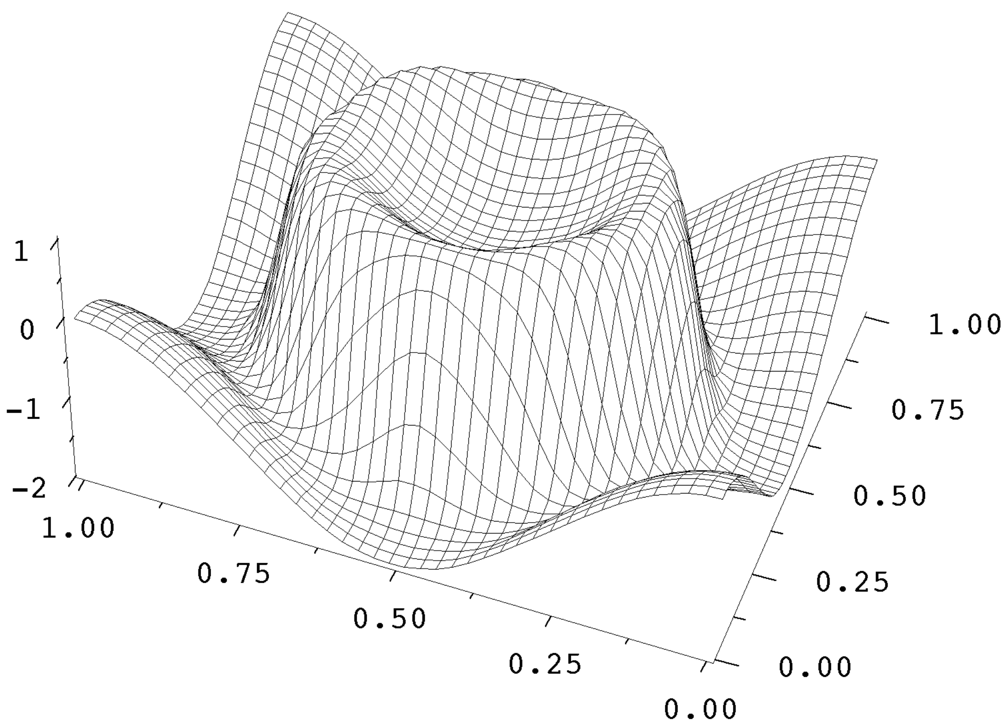

Figure 7: Surface plot of WENO5 shock capturing solution at $t = 1$ with $N_x = N_y = 40$.

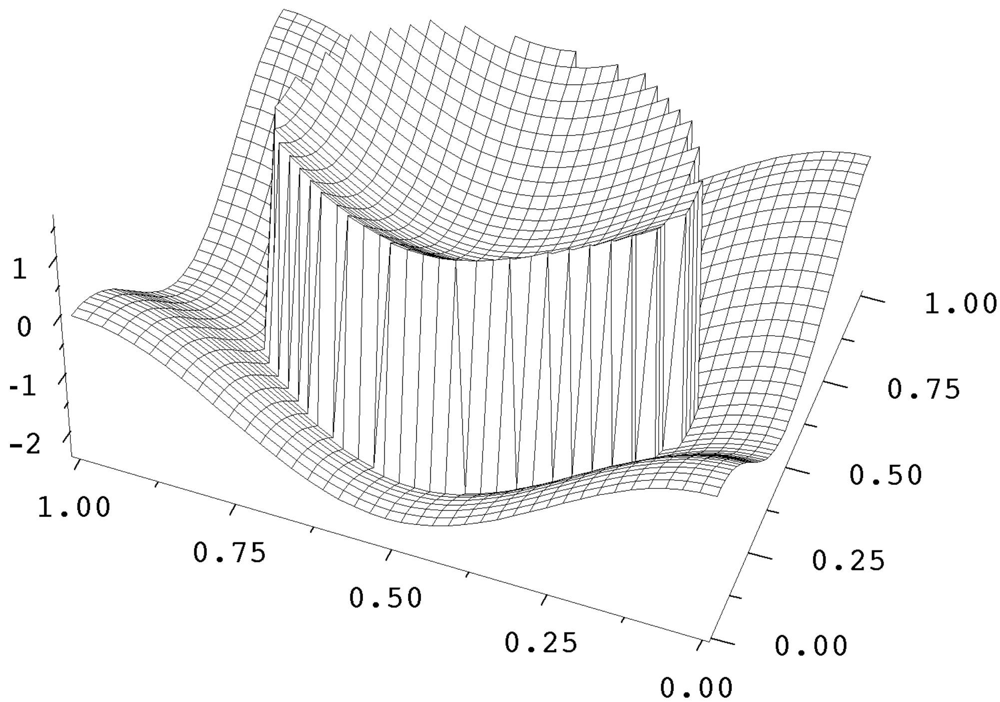

Figure 8: Surface plot of WENO5 level set tracking solution at $t = 1$ with $N_x = N_y = 40$.

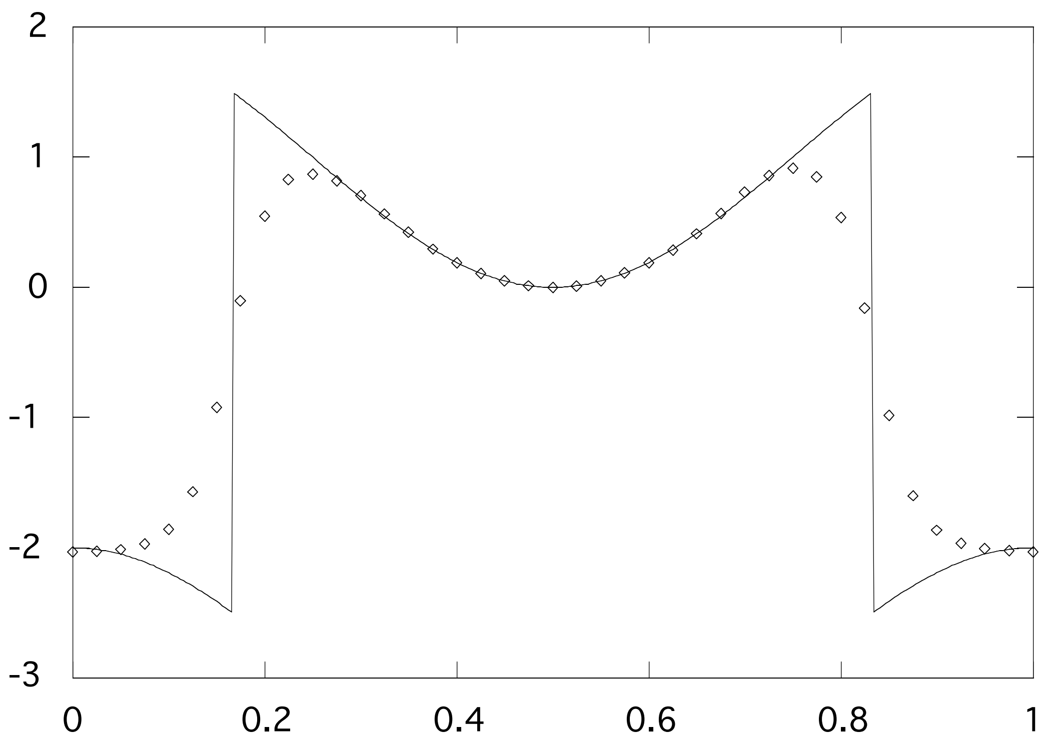

Figure 9: Plot of WENO5 shock capturing solution at $y = 0.5, t = 1$ with $N_x = N_y = 40$, $\diamond$, and exact solution, solid line.

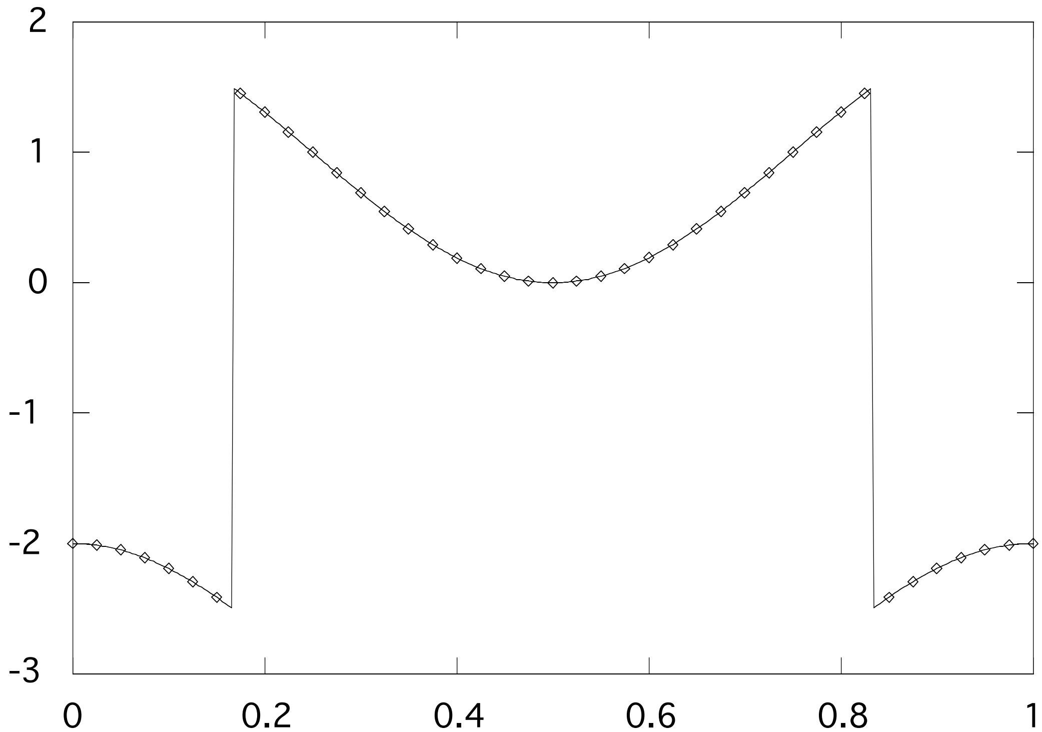

Figure 10: Plot of WENO5 level set tracking solution at $y = 0.5, t = 1$ with $N_x = N_y = 40$, $\diamond$, and exact solution, solid line.

4.4 Two Dimensional Solid Body Rotation

Here, a non-constant coefficient two dimensional test is conducted. The equation to be solved is the two dimensional linear advection equation, corresponding to solid body rotation:

$$u_{,t} + (-2\pi y u)_{,x} + (2\pi x u)_{,y} = 0 \tag{67}$$

subject to the initial conditions:

$$u = \begin{cases} 1 - y^2 \cos(2x), & \text{if } ((x-0.3)^2 + (y-0.3)^2)^{1/2} < 1/3 \\ 0, & \text{otherwise} \end{cases} \tag{68}$$

Numerically, linear extrapolation is used at all boundaries, $x = -1$, $x = 1$, $y = -1$ and $y = 1$. Notice that this initial condition has a discontinuity along a circle of radius, $R = 1/3$, whose center is located at $(x, y) = (0.3, 0.3)$. This function is represented by equation (4) with

$$u_1 = 1 - y^2 \cos(2x) \tag{69}$$

$$u_2 = 0 \tag{70}$$

$$\psi = \frac{1}{3} - ((x-0.3)^2 + (y-0.3)^2)^{1/2} \tag{71}$$

Since this problem is linear, the level set equation will become:

$$\psi_{,t} - 2\pi y \psi_{,x} + 2\pi x \psi_{,y} = 0. \tag{72}$$

Again, there is no need to check for an entropy satisfying ghost node state. At $t = 1$ (1 full rotation) with N_t timesteps, the error in the numerical solution using the discrete L_1 and L_∞ norms is measured, see Table 4. Notice that the level set tracking algorithm converges at fifth order in both the L_1 and L_∞ norms. Again, for this linear problem, the standard shock capturing algorithm converges at roughly 5/6 order in the L_1 norm. Figure 11 shows a surface plot of the solution of the WENO5 shock capturing algorithm, and Figure 12 shows the solution using the WENO5 level set tracking algorithm. Figure 13 show a line plot of the shock capturing solution at $t = 1$ at $x = 1/2$. Figure 14 shows a line plot of the level set tracking solution at $t = 1$ at $x = 1/2$.

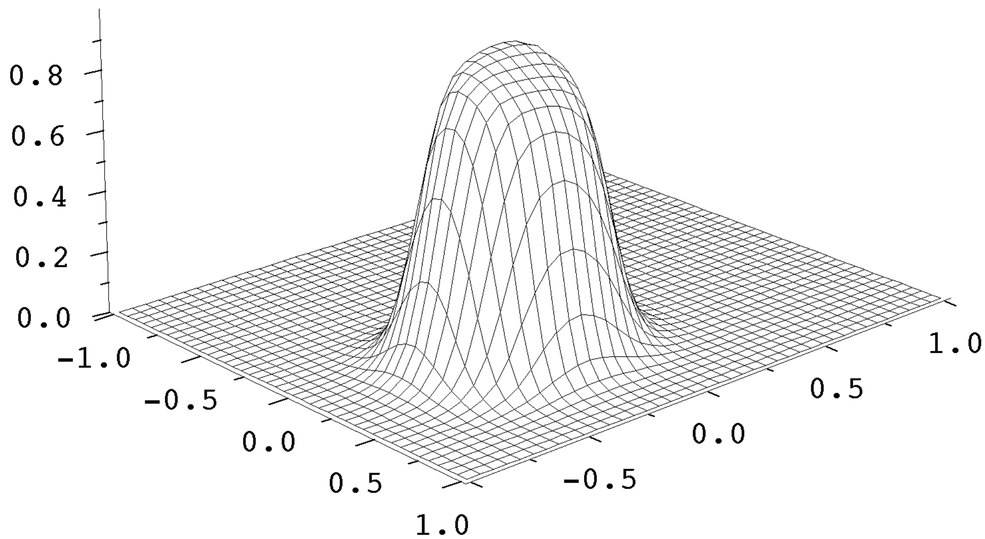

Figure 11: Surface plot of WENO5 shock capturing solution at $t = 1$ with $N_x = N_y = 40$.

TABLE 4: Numerical Accuracy for 2D solid body rotation.

$N_x = N_y$	N_t	E_{1-LST}	R_{1-LST}	$E_{\infty-LST}$	$R_{\infty-LST}$	E_{1-SC}	R_{1-SC}
20	130	5.85e-4		4.13e-3		2.01e-1	
40	415	2.00e-5	4.87	6.64e-4	2.64	1.09e-1	0.88
80	1310	2.15e-7	6.54	9.49e-6	6.13	6.20e-2	0.82
160	4160	7.56e-9	4.83	7.95e-7	3.58	3.53e-2	0.81

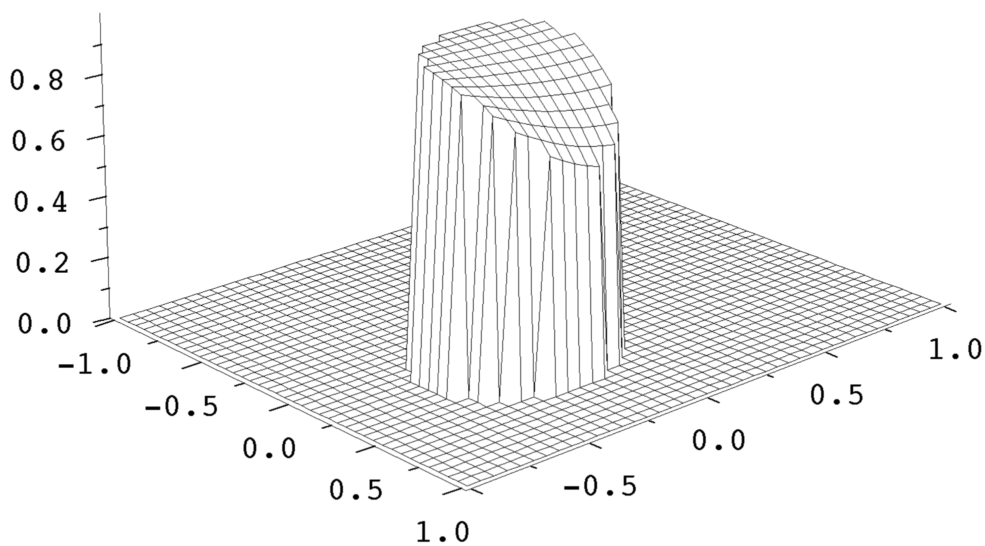

Figure 12: Surface plot of WENO5 level set tracking solution at $t = 1$ with $N_x = N_y = 40$.

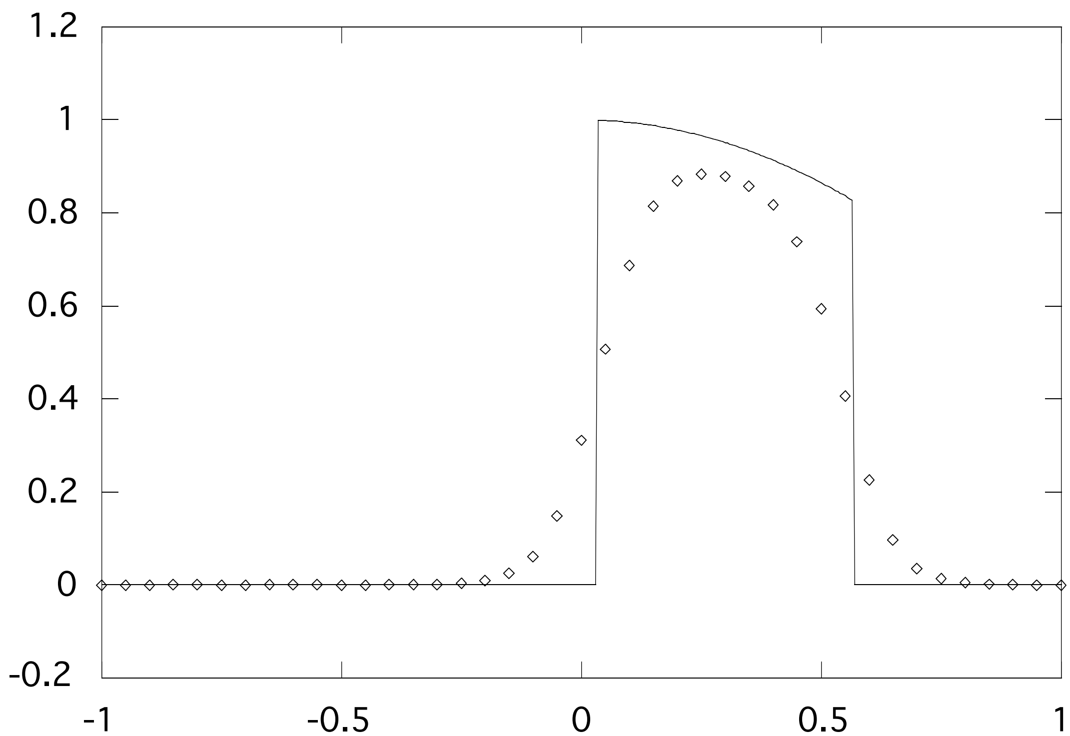

Figure 13: Plot of WENO5 shock capturing solution at $x = 0.5, t = 1$ with $N_x = N_y = 40$, $\diamond$, and exact solution, solid line.

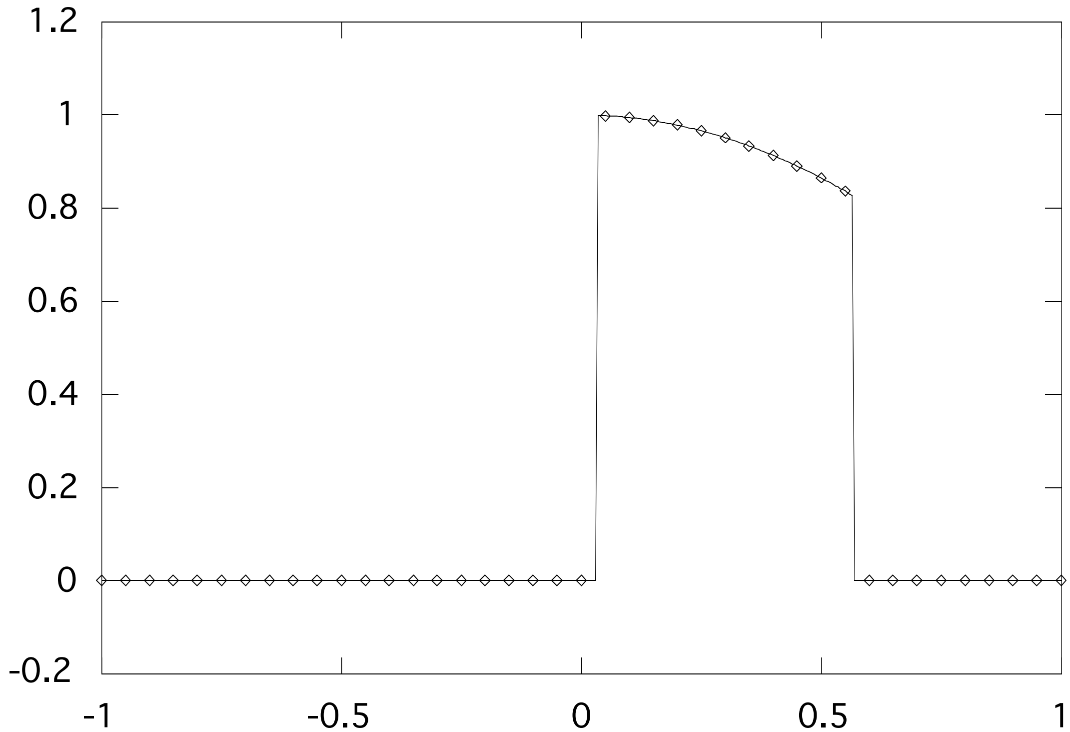

Figure 14: Plot of WENO5 level set tracking solution at $x = 0.5, t = 1$ with $N_x = N_y = 40$, $\diamond$, and exact solution, solid line.

4.5 Two Dimensional Burgers' Equation

Here, a nonlinear two dimensional test is conducted. The equation to be solved is the two dimensional Burgers' equation:

$$u_{,t} + (\frac{u^2}{2})_{,x} + (\frac{u^2}{2})_{,y} = 0, \tag{73}$$

with periodic boundary conditions at $x = 0$, $x = 1$, $y = 0$ and $y = 1$, subject to the initial conditions:

$$u = \begin{cases} \frac{1}{2}(1 + \cos(2\pi(x+a))), & \text{if } \frac{1}{3} < x + a < \frac{2}{3} \\ \frac{1}{2} + \sin(2\pi(x+a)), & \text{otherwise} \end{cases} \tag{74}$$

where

$$a = 0.1\sin(2\pi y). \tag{75}$$

Notice that this initial condition has discontinuities along the curves $x = \frac{1}{3} - 0.1\sin(2\pi y)$ and $x = \frac{2}{3} - 0.1\sin(2\pi y)$. See Figure 15.

This function is represented by equation (4) with

$$u_1 = \frac{1}{2} + \sin(2\pi(x+a)) \tag{76}$$

$$u_2 = \frac{1}{2}(1 + \cos(2\pi(x+a))) \tag{77}$$

$$\psi = \begin{cases} \frac{1}{3} - (x+a), & \text{if } x + a < \frac{1}{2} \\ x + a - \frac{2}{3}, & \text{otherwise} \end{cases} \tag{78}$$

At $t = 0.1$ with N_t timesteps, the error in the numerical solution using the discrete L_1 and L_∞ norms is measured, see Table 5. Notice that the level set tracking algorithm converges at fifth order in both the L_1 and L_∞ norms. Again, the standard shock capturing algorithm converges at first order in the L_1 norm. Figure 16 shows a surface plot of the solution of the WENO5 shock capturing algorithm, and Figure 17 shows the solution using the WENO5 level set tracking algorithm. Figure 18 show a line plot of the shock capturing solution at $t = 0.1$ at $y = 1/2$. Figure 19 shows a line plot of the level set tracking solution at $t = 0.1$ at $y = 1/2$.

TABLE 5: Numerical Accuracy for 2D Burgers' equation.

$N_x = N_y$	N_t	E_{1-LST}	R_{1-LST}	$E_{\infty-LST}$	$R_{\infty-LST}$	E_{1-SC}	R_{1-SC}
20	5	1.83e-3		8.15e-3		2.89e-2	
40	15	1.77e-4	3.37	1.55e-3	2.39	1.36e-2	1.09
80	50	9.72e-6	4.18	1.43e-4	3.44	6.34e-3	1.10
160	150	1.73e-7	5.81	5.41e-6	4.72	3.08e-3	1.04

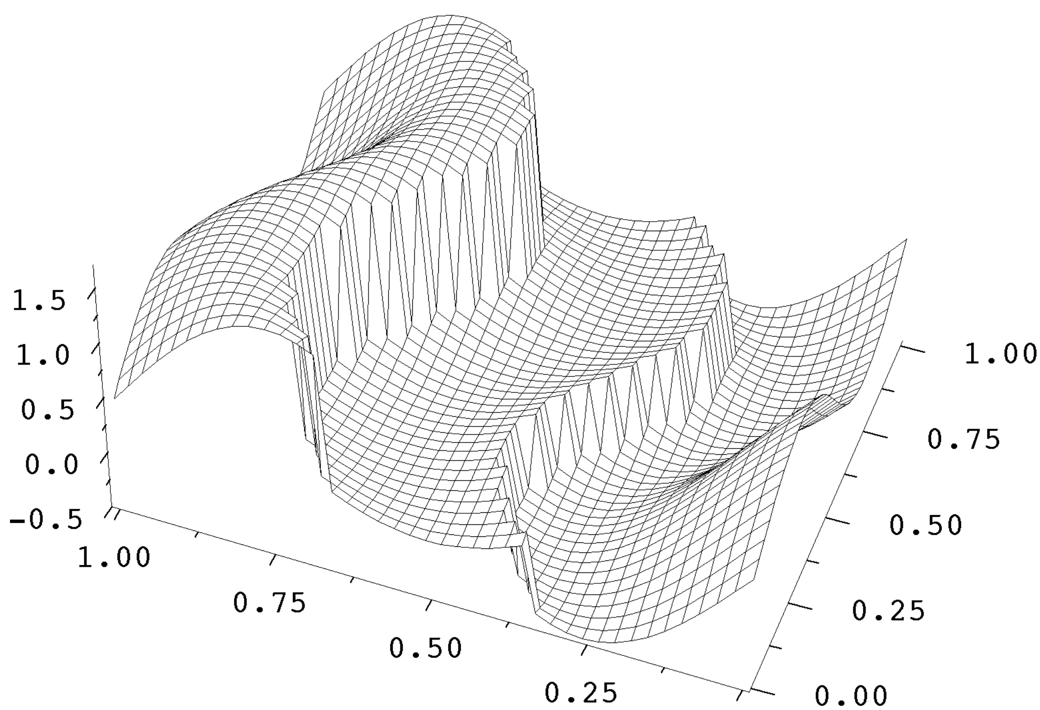

Figure 15: Surface Plot of initial conditions corresponding to equation (74).

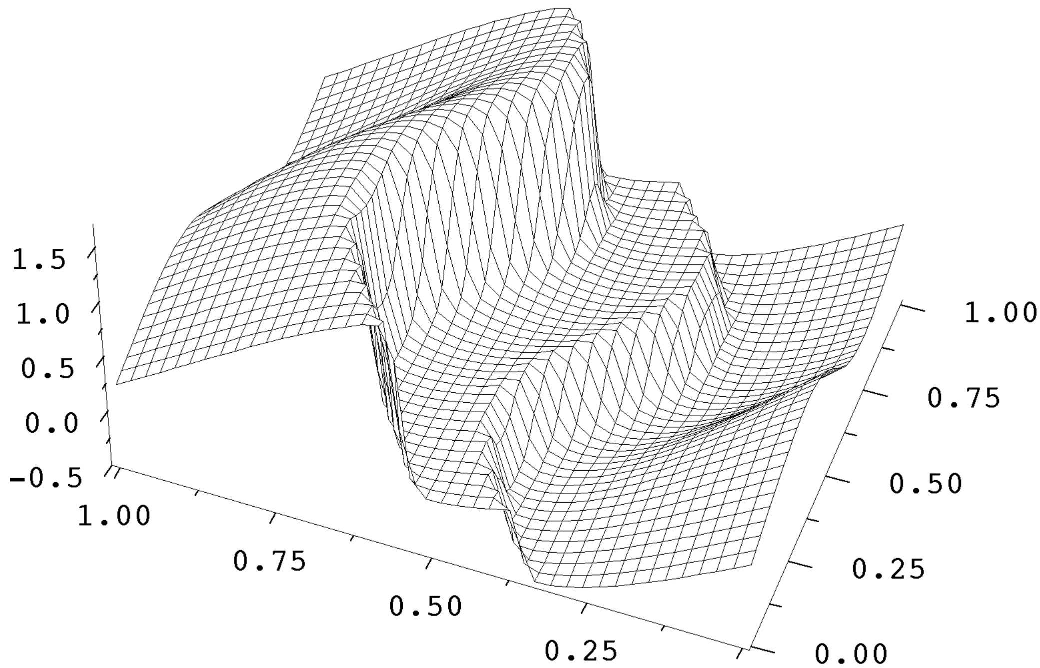

Figure 16: Surface plot of WENO5 shock capturing solution at $t = 0.1$ with $N_x = N_y = 40$.

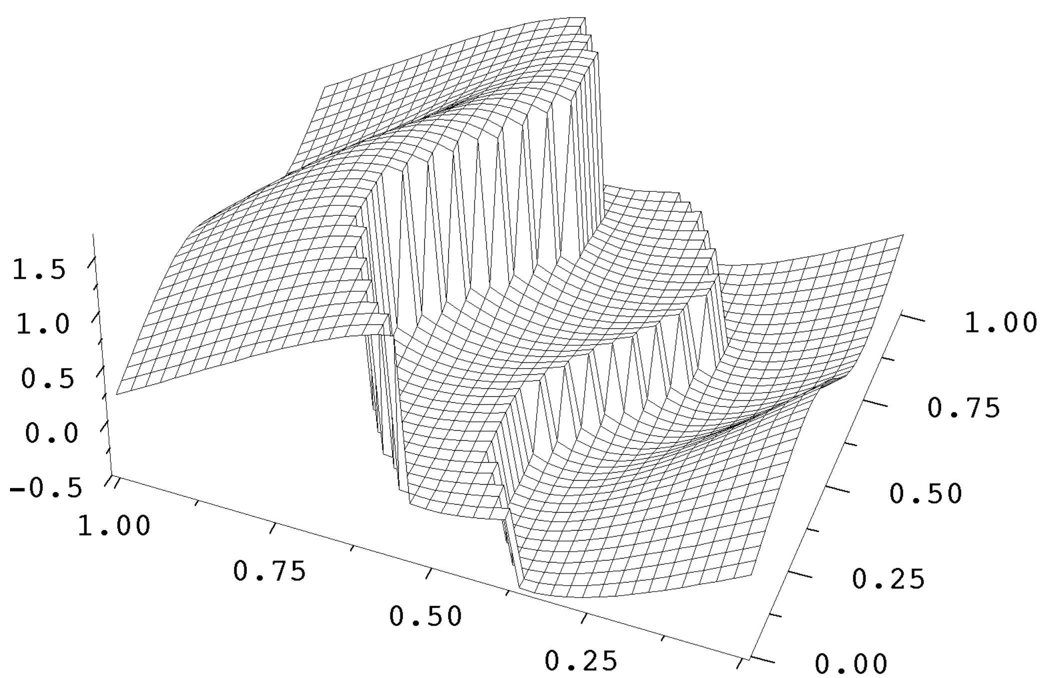

Figure 17: Surface plot of WENO5 level set tracking solution at $t = 0.1$ with $N_x = N_y = 40$.

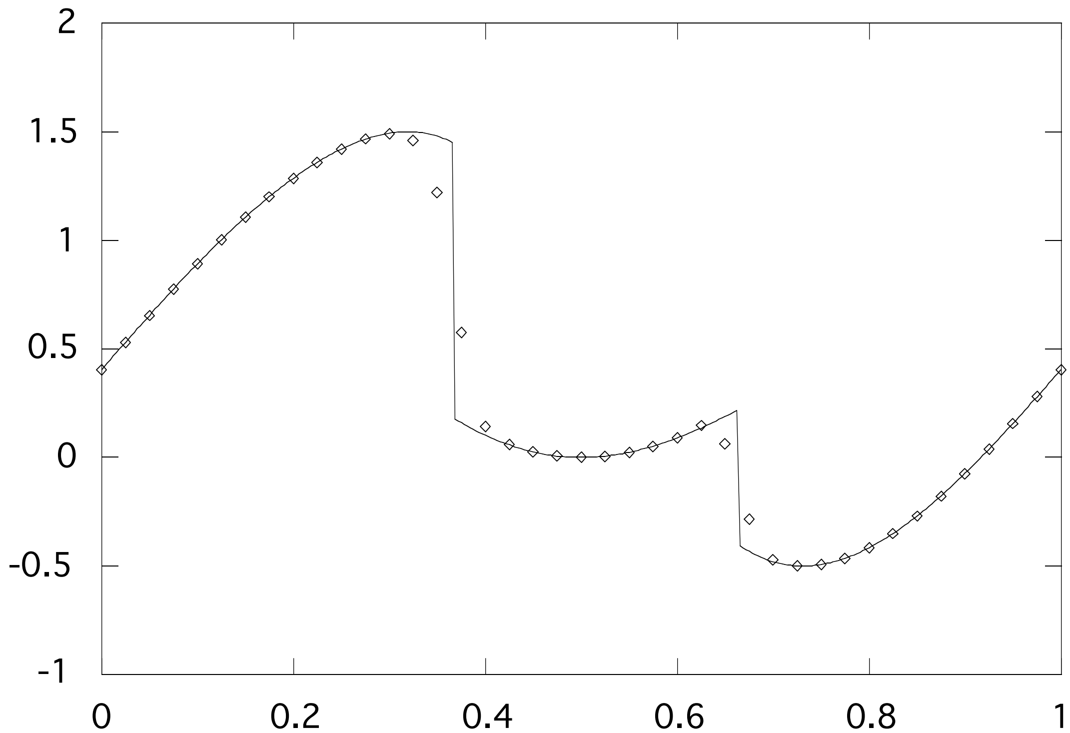

Figure 18: Plot of WENO5 shock capturing solution at $y = 0.5, t = 0.1$ with $N_x = N_y = 40$, $\diamond$, and converged level set tracking solution ($N_x = N_y = 320$), solid line.

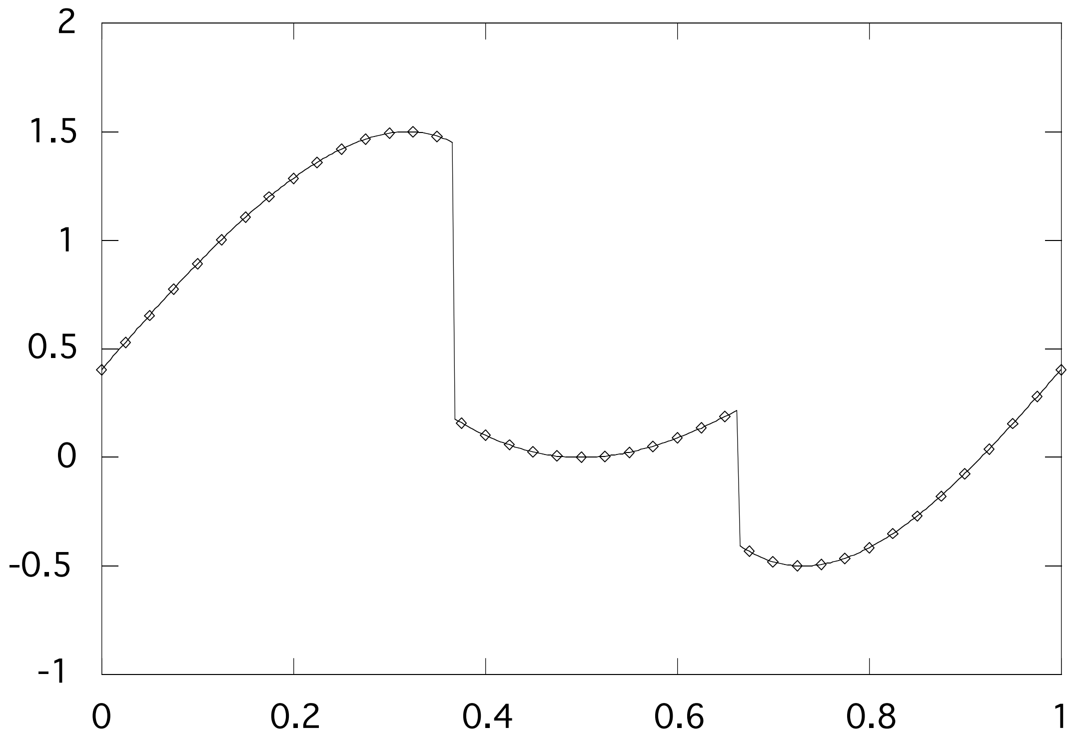

Figure 19: Plot of WENO5 level set tracking solution at $y = 0.5, t = 0.1$ with $N_x = N_y = 40$, $\diamond$, and converged level set tracking solution ($N_x = N_y = 320$), solid line.

5 Conclusions and Discussion

A simple level set algorithm for solving scalar hyperbolic conservation laws is presented. For scalar problems, high rates of convergence are demonstrated in linear and nonlinear problems, even near discontinuities and in multi-dimensions. Note that this scheme takes roughly 3 times as many numerical operations and memory as a standard capturing scheme, but the benefit greatly outweighs any extra computational time or storage. Also, this extrapolation free method can be applied to the original ghost fluid method [6], by initially extrapolating the density, etc. in a smooth fashion, and only projecting the ghost node states into the proper boundary condition states. Note that one can also use this algorithm for tracking discontinuous derivatives in Hamilton-Jacobi equations, with the modification that the "shock" speed will be a function of the derivatives of the solution states, since it has been shown in [15] and [16] that there is a relation between Hamilton-Jacobi equations and scalar conservation laws.

Work is currently being done to track multi-dimensional shock fronts in gas dynamics with a level set formulation. The largest difference is that in a system, there are characteristics that pass through the shock front, and so the boundary treatment must take this into account. Also, the two states will not be symmetric, as in the scalar case; one state, say $\vec{u}_2$, will be considered a shock state of the other, $\vec{u}_1$. A simple way to apply the appropriate ghost region state for $\vec{u}_2$ is to project this ghost region state into an appropriate shock state of the unshocked fluid, $\vec{u}_1$. This shock state is determined once one has an appropriate shock speed. The shock speed can be determined from a local Riemann problem. This can be determined exactly (eg. Godunov's method), or it can be determined from an approximate method (eg. Roe's method). The ghost region for $\vec{u}_1$ is not critical, since in the shock attached reference frame, $\vec{u}_1$ is supersonic. Again, one must ensure that both $\vec{u}_1$ and $\vec{u}_2$ are entropy satisfying states in their respective ghost regions. Preliminary results indicate that this method converges to the proper solution, with improved convergence and accuracy. Figure (20) shows the density field for a level set tracking algorithm, using a non-decomposition based Lax-Friedrichs scheme [22], [13], with a shock speed given from a Roe averaged $\bar{u}+\bar{c}$, corresponding to example 6 and Figure 14 (b) of [21]. Importantly, note there is no "intermediate" shock points near the lead shock at $x \approx 2.4$. Again, this work will be investigated in greater detail elsewhere [2].

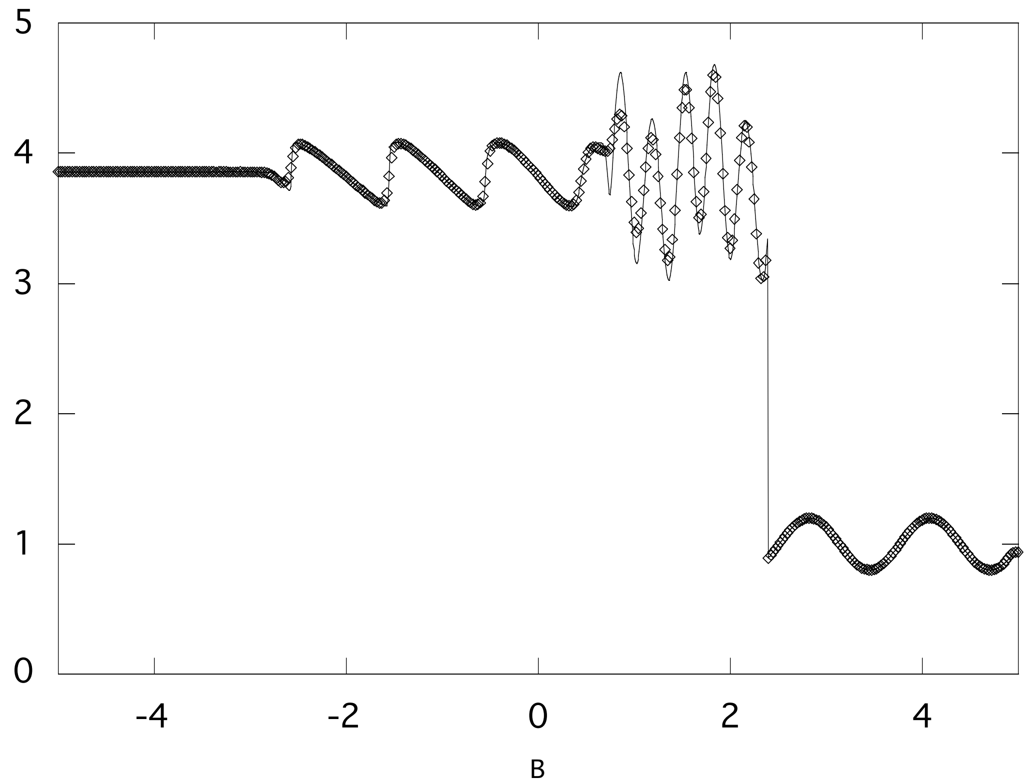

Figure 20: Plot of WENO5 level set tracking solution at $t = 1.8$ with $N_x = N_t = 400$, $\diamond$, and converged level set tracking solution ($N_x = N_t = 1600$), solid line.

Acknowledgments

The author would like to thank R. Fedkiw (UCLA), S. Osher (UCLA) and J. Bdzil (LANL) for reading through preliminary versions of this paper and making useful suggestions.

References

[1] M. Arora and P. L. Roe, "On Postshock oscillations Due to Shock Capturing Schemes in Unsteady Flows," *Journal of Computational Physics*, **130**, 25-40 (1997)

[2] T. D. Aslam, "A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws II: Systems of Equations," in preparation

[3] I.-L. Chern, J. Glimm, O. McBryan, B. Plohr and S. Yaniv, "Front Tracking for Gas Dynamics," *Journal of Computational Physics*, **62**, 83-110 (1986)

[4] R. Donat, "Studies on Error Propagation for Certain Nonlinear Approximations to Hyperbolic Equations: Discontinuities in Derivatives," *SIAM Journal on Numerical Analysis*, **31**, 3, 655-679, (1994)

[5] R. Donat and S. Osher, "Propagation of error into regions of smoothness for non-linear approximations to hyperbolic equations," *Computer Methods in Applied Mechanics and Engineering*, **80**, 59-64, (1990)

[6] R. P. Fedkiw, T. Aslam, B. Merriman and S. Osher, "A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)," submitted *Journal of Computational Physics*, (1998)

[7] A. Harten, "ENO Schemes with Subcell Resolution," *Journal of Computational Physics*, **82**,148-184 (1989)

[8] A. Harten and S. Osher, "Uniformly High-Order Accurate Nonoscillatory Schemes, I," *SIAM Journal on Numerical Analysis*, **24**, 2, 279-309, (1987)

[9] G.-S. Jiang and D. Peng, "Weighted ENO Schemes for Hamilton-Jacobi Equations," UCLA CAM Report 97-29, (1997) *SIAM Journal on Numerical Analysis*, to appear

[10] G.-S. Jiang and C.-W. Shu, "Efficient Implementation of Weighted ENO Schemes," *Journal of Computational Physics*, **126**, 202-228 (1996)

[11] R. J. Leveque, "Numerical Methods for Conservation Laws," Birkhauser Verlag, (Basel), (1992)

[12] R. J. LeVeque and K.-M. Shyue, "Two-Dimensional Front Tracking Based on High Resolution Wave Propagation Methods," *SIAM Journal on Scientific Computing*, **16**, 2, 348-377 (1995)

[13] X.-D. Liu, and S. Osher, "Convex ENO High Order Multi-Dimensional Schemes Without Field-by-Field Decomposition or Staggered Grids," UCLA CAM report 97-26 (1997), *Journal of Computational Physics*, to appear

[14] X.-D. Liu, S. Osher, T. Chan, "Weighted Essentially Non-oscillatory Schemes," *Journal of Computational Physics*, **115**, 200-212, (1994)

[15] S. Osher and J. A. Sethian, "Fronts Propagating with Curvature- Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *Journal of Computational Physics*, **79**, 12-49, (1988)

[16] S. Osher and C.-W. Shu, "High-Order essentially nonoscillatory schemes for Hamilton-Jacobi equations," *SIAM Journal on Numerical Analysis*, **28**, 4, 907-922, (1991)

[17] J. J. Quirk, "A Contribution to the Great Riemann Solver Debate," *International Journal for Numerical Methods in Fluids*, **18**, 555-574, (1994)

[18] T. W. Roberts, "The Behavior of Flux Difference Splitting Schemes near Slowly Moving Shock Waves," *Journal of Computational Physics*, **90**, 141-160 (1990)

[19] M. Sever (Mock), "Order of dissipation near rarefaction centers," Progress and Supercomputing in Computational Fluid Dynamics," Proceedings of U.S.-Isreal Workshop, Birkhauser, Boston, (1985)

[20] C.-W. Shu and S. Osher, "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, **77**, 439-471, (1988)

[21] C.-W. Shu and S. Osher, "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes II," *Journal of Computational Physics*, **83**, 32-78 (1989)

[22] S. Xu, T. Aslam, and D. S. Stewart, "High Resolution Numerical Simulation of Ideal and Non-ideal Compressible Reacting Flows with Embedded Internal Boundaries," *Combustion Theory and Modelling*, **1**, 1, 113-142 (1997)

[23] H. Yang, "An Artificial Compression Method for ENO Schemes: The Slope Modification Method," *Journal of Computational Physics*, **89**, 125-160 (1990)