Journal of Computational Mathematics, Vol.26, No.3, 2008, 336–346.

# NUMERICAL BOUNDARY CONDITIONS FOR THE FAST SWEEPING HIGH ORDER WENO METHODS FOR SOLVING THE EIKONAL EQUATION*

Ling Huang
*Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China*
*E-mail: huangl0@mail.ustc.edu.cn*
Chi-Wang Shu
*Division of Applied Mathematics, Brown University, Providence, RI 02912, USA*
*Email: shu@dam.brown.edu*
Mengping Zhang
*Department of Mathematics, University of Science and Technology of China, Hefei 230026, China*
*Email: mpzhang@ustc.edu.cn*

**Dedicated to Professor Junzhi Cui on the occasion of his 70th birthday**

**Abstract**

High order fast sweeping methods have been developed recently in the literature to solve static Hamilton-Jacobi equations efficiently. Comparing with the first order fast sweeping methods, the high order fast sweeping methods are more accurate, but they often require additional numerical boundary treatment for several grid points near the boundary because of the wider numerical stencil. It is particularly important to treat the points near the inflow boundary accurately, as the information would flow into the computational domain and would affect global accuracy. In the literature, the numerical solution at these boundary points are either fixed with the exact solution, which is not always feasible, or computed with a first order discretization, which could reduce the global accuracy. In this paper, we discuss two strategies to handle the inflow boundary conditions. One is based on the numerical solutions of a first order fast sweeping method with several different mesh sizes near the boundary and a Richardson extrapolation, the other is based on a Lax-Wendroff type procedure to repeatedly utilizing the PDE to write the normal spatial derivatives to the inflow boundary in terms of the tangential derivatives, thereby obtaining high order solution values at the grid points near the inflow boundary. We explore these two approaches using the fast sweeping high order WENO scheme in [18] for solving the static Eikonal equation as a representative example. Numerical examples are given to demonstrate the performance of these two approaches.

*Mathematics subject classification:* 65N06, 65N22.
*Key words:* Fast sweeping method, WENO scheme, Boundary condition.

## 1. Introduction

In this paper we are interested in the numerical solution of two dimensional static Hamilton-Jacobi equations

$$H(\phi_x, \phi_y) = f(x, y) \tag{1.1}$$

* Received February 29, 2008 / accepted March 14, 2008 /

which is defined on a domain $\Omega$ with suitable boundary conditions. Typically, the boundary condition for the solution $\phi$ is provided in the inflow part $\Gamma$ of the boundary. In particular, we will only study the so-called Eikonal equation in this paper as an example, that is, the Hamiltonian $H$ in (1.1) is given by

$$H(u,v)=\sqrt{u^2+v^2}. \tag{1.2}$$

Notice that the solution to (1.1) may not always be differentiable or unique, and we are interested in the viscosity solution [4] which is unique, Lipschitz continuous, but may not be everywhere differentiable.

Applications in which the Hamilton-Jacobi equation (1.1), in particular the Eikonal equation (1.1)-(1.2), appears are abundant, for example the level set method, image processing and computer vision, and control theory. Some of the recently developed pedestrian flow models [7, 16] also involve the static Eikonal equation.

Numerical discretization for (1.1) includes first order monotone schemes on structured meshes [5] and on unstructured meshes [1], high order essentially non-oscillatory (ENO) schemes on structured meshes [11, 12], high order weighted ENO (WENO) schemes on structured meshes [8], high order WENO schemes on unstructured meshes [17], and high order discontinuous Galerkin methods on unstructured meshes [3, 6], among many others. A review of the discretization techniques for the Hamilton-Jacobi equations can be found in [14].

For a time dependent Hamilton-Jacobi equation

$$\phi_t+H(\phi_x,\phi_y)=f(x,y), \tag{1.3}$$

an explicit time discretization, such as the total variation diminishing (TVD) time discretization in [15], is often used. Such discretization can also be used to obtain the steady state solution of (1.1), by marching in time until the difference of the numerical solution between successive time steps becomes negligibly small. This however may not be the most efficient approach to obtain the solution of (1.1). In recent years, the fast sweeping method has been developed as one of the efficient techniques for obtaining the steady state solution of (1.1). The original fast sweeping method [2, 19] is only for first order monotone schemes on structured meshes. For such first order schemes, there is no issue for numerical boundary conditions, since the first order upwind discretization will only need values from the physically given boundary condition on the inflow part of the domain boundary. Later, the fast sweeping method is generalized to some of the high order spatial discretizations. For example, in [18], the fast sweeping method is generalized to the high order WENO scheme of [8]; and in [10], it is generalized to the high order discontinuous Galerkin method of [3]. These high order fast sweeping methods are also used in the pedestrian flow simulations in [7, 16], which require repeated solution of a static Eikonal equation. The high order fast sweeping methods produce much more accurate solutions on coarser meshes when compared with the first order fast sweeping method. However, they do involve an additional difficulty associated with high order spatial discretizations, namely the necessity to treat numerical boundary conditions near the boundary. Our numerical experiments indicate that the main difficulty is near the inflow boundary, as simple extrapolation could take care of the outflow boundary since the information there would flow out of the computational domain. We will use the high order WENO scheme in [18] as a representative example to explain this difficulty. Because of the wider numerical stencil required for the high order WENO interpolation, the high order fast sweeping WENO method needs a suitable numerical boundary treatment for several grid points near the inflow boundary. In [18] and also several other papers on similar methods [7,10,16], the values of the numerical solution at these boundary points are either fixed with the exact solution, which is not always feasible, or computed with a first order discretization, which would reduce the global accuracy.

In this paper, we use the fast sweeping high order WENO scheme in [18] for solving the static Eikonal equation (1.1)-(1.2), as a representative example, to discuss two strategies to handle these numerical boundary conditions. In the first approach we use a first order fast sweeping method to produce numerical solutions with several different mesh sizes near the boundary, then we form a Richardson extrapolation to obtain suitable high order solution values at the grid points near the inflow boundary. This approach usually involves only a small additional computational cost because the numerical solution at the grid points near the inflow boundary can often be obtained with only local sweeping in the first order fast sweeping method. In the second approach we use a Lax-Wendroff type procedure, to repeatedly utilizing the PDE to write the normal spatial derivatives to the inflow boundary in terms of the tangential derivatives, which would then be readily available by the physical inflow boundary condition. With these normal spatial derivatives we can then obtain high order solution values at the grid points near the inflow boundary. This approach, when applicable, involves a negligibly small additional computational cost.

The rest of the paper is organized as follows. We describe the two approaches for the numerical boundary conditions in Section 2. In Section 3 we provide several numerical examples to demonstrate the performance of these two approaches. Concluding remarks are given in Section 4.

## 2. The Numerical Scheme and the Treatment of Boundary Conditions

In this section we first give a brief description of the third order fast sweeping WENO scheme [18] for solving the static Eikonal equation (1.1)-(1.2). We then describe two approaches for the numerical boundary conditions.

### 2.1. The third order fast sweeping WENO scheme for the Eikonal equation

We give a very brief description of the third order fast sweeping WENO scheme [18] for solving the static Eikonal equation (1.1)-(1.2). For more details, we refer to [8,18].

For simplicity, we assume that the computational domain $\Omega$ is a box $[0,1]^2$ which is covered by a tensor product mesh $(x_i, y_j)$ with $0 \le i \le I$ and $0 \le j \le J$. We assume without loss of generality that the mesh is uniform, $x_i = i\Delta x$, $y_j = j\Delta y$ and $\Delta x = \Delta y = h$. The approximation of the solution to the static Eikonal equation (1.1)-(1.2) at the location $(x_i, y_j)$ is denoted by $\phi_{i,j}$, which is obtained by a fast sweeping iterative procedure. For the convenience of the algorithm description below, we divide the set of mesh points $(x_i, y_j)$ into the following four categories:

- *Category I* contains the points at the inflow part of the domain boundary. The numerical solution $\phi_{i,j}$ in Category I is fixed at the prescribed physical boundary condition and does not change during the fast sweeping iteration.
- *Category II* contains the points at the outflow part of the domain boundary, where no physical boundary condition is given, and the ghost points outside the computational domain near the outflow boundary which are necessary for the wide stencil WENO in-

terpolation. The numerical solution $\phi_{i,j}$ in Category II is obtained by extrapolation of suitable accuracy, based on the numerical solution inside the computational domain.

- *Category III* contains the few points inside the computational domain and near the inflow boundary. These points cannot be updated by the WENO scheme because of its wide stencil. For the third order WENO scheme under consideration, any point which has a horizontal or vertical distance less than $3h$ from the inflow boundary belongs to this category. The strategy in [10, 18] to treat points in Category III is to fix the numerical solution $\phi_{i,j}$ as the exact solution of the PDE (1.1)-(1.2) and it does not change during the fast sweeping iteration. This of course is not always feasible. The strategy in [7, 16] is to fix the numerical solution $\phi_{i,j}$ as that of a first order fast sweeping solution and it does not change during the fast sweeping iteration. This could of course lead to a loss of local and hence global accuracy, since information will flow from this part of the boundary into the computational domain. The main purpose of this paper is to study two strategies to handle the points in Category III.
- *Category IV* contains all the remaining points, which are updated during the fast sweeping iterations until convergence.

The solution from the first-order Godunov fast sweeping method [19] is used as the initial guess for all the grid points in Category IV. Grid values in Categories I and III are fixed as appropriate, and before each iteration, grid values in Category II are obtained by suitable extrapolation.

The following Gauss-Seidel iterations with four alternating direction sweepings are then performed:

$$\begin{array}{llll} (1) & i=0:I, j=0:J; & (2) & i=I:0, j=0:J; \\ (3) & i=I:0, j=J:0; & (4) & i=0:I, j=J:0. \end{array} \tag{2.1}$$

When we loop to a point $(i,j)$, if it belongs to Category IV, the solution is updated as follows,

$$\phi_{i,j}^{new} = \begin{cases} \min(\phi_{i,j}^{xmin}, \phi_{i,j}^{ymin}) + f_{i,j}\, h, & \text{if } |\phi_{i,j}^{xmin} - \phi_{i,j}^{ymin}| \le f_{i,j}\, h, \\ \frac{1}{2}\Big(\phi_{i,j}^{xmin} + \phi_{i,j}^{ymin} + \Big(2f_{i,j}^2 h^2 - (\phi_{i,j}^{xmin} - \phi_{i,j}^{ymin})^2\Big)^{\frac{1}{2}},\Big) & \text{otherwise,} \end{cases} \tag{2.2}$$

where $f_{i,j} = f(x_i, y_j)$, and

$$\begin{cases} \phi_{i,j}^{xmin} = \min\big(\phi_{i,j} - h\,(\phi_x)_{i,j}^-,\ \ \phi_{i,j} + h\,(\phi_x)_{i,j}^+\big), \\ \phi_{i,j}^{ymin} = \min\big(\phi_{i,j} - h\,(\phi_y)_{i,j}^-,\ \ \phi_{i,j} + h\,(\phi_y)_{i,j}^+\big), \end{cases} \tag{2.3}$$

with

$$(\phi_x)_{i,j}^- = (1-w_-)\left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}\right) + w_-\left(\frac{3\phi_{i,j} - 4\phi_{i-1,j} + \phi_{i-2,j}}{2h}\right), \tag{2.4}$$

$$(\phi_x)_{i,j}^+ = (1-w_+)\left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}\right) + w_+\left(\frac{-3\phi_{i,j} + 4\phi_{i+1,j} - \phi_{i+2,j}}{2h}\right), \tag{2.5}$$

$$w_- = \frac{1}{1+2r_-^2}, \qquad r_- = \frac{\varepsilon + (\phi_{i,j} - 2\phi_{i-1,j} + \phi_{i-2,j})^2}{\varepsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}, \tag{2.6}$$

$$w_+ = \frac{1}{1+2r_+^2}, \qquad r_+ = \frac{\varepsilon + (\phi_{i,j} - 2\phi_{i+1,j} + \phi_{i+2,j})^2}{\varepsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}. \tag{2.7}$$

Here $\varepsilon$ is a small number in the WENO nonlinear weights. The definitions for $(\phi_y)_{i,j}^-$ and $(\phi_y)_{i,j}^+$ are of course analogous.

Convergence is declared if

$$\left\|\phi^{new}-\phi^{old}\right\| \le \delta, \tag{2.8}$$

where $\delta$ is a given convergence threshold value.

### 2.2. Boundary treatment strategy I: Richardson extrapolation

The first strategy that we propose to treat points in Category III is to obtain several first order accurate solutions with different mesh sizes, then use Richardson extrapolation to obtain accurate point values for those points in Category III. This is feasible without excessive computational cost because points in Category III are close to the inflow boundary, hence the first order fast sweeping iterations can be performed locally, greatly reducing the computational cost.

Richardson extrapolation is a well-known idea so we will only describe our application of this idea briefly. Assume $I_h$ is the numerical solution of the first order fast sweeping scheme with mesh size $h$ at the location $(x^*, y^*)$, which is a grid point in Category III. If we further assume

$$I_h - I = \alpha h + \beta h^2 + \mathcal{O}(h^3)$$

with constants $\alpha$ and $\beta$, where $I$ is the exact solution at the location $(x^*, y^*)$, which is reasonable when the exact solution is smooth, then clearly

$$\tilde{I}_h = \frac{1}{3} I_h - 2 I_{h/2} + \frac{8}{3} I_{h/4} \tag{2.9}$$

would be a third order approximation to $I$:

$$\tilde{I}_h - I = \mathcal{O}(h^3).$$

This boundary treatment strategy is suitable for most types of inflow boundaries, including the source boundary consisting of a single point. The efficiency of this strategy however depends on how fast we can compute the first order approximations $I_h$, $I_{h/2}$ and $I_{h/4}$ for all grid points inside Category III. When the characteristics from the inflow boundary do not intersect with each other, such first order fast sweeping computation can be performed locally and is very fast. When the characteristics from the inflow boundary do intersect with each other, the efficiency of this strategy would decrease. Fortunately, in this case the inflow boundary would not be a single point, hence the second strategy described in next subsection would usually be applicable.

### 2.3. Boundary treatment strategy II: a Lax-Wendroff type procedure

The original Lax-Wendroff scheme [9] uses an important idea of converting the time derivatives to spatial derivatives, by repeatedly using the PDE. We propose to use the same idea to obtain high order approximations to the solution values for the points in Category III.

To fix the ideas, let us assume that the left boundary

$$\Gamma = \{(x, y) : x = 0,\ 0 \le y \le 1\} \tag{2.10}$$

of the computational domain $[0, 1]^2$ is the inflow boundary, on which the solution is given as

$$\phi(0, y) = g(y), \qquad 0 \le y \le 1.$$

We would like to obtain a high order approximation to the solution value $\phi_{i,j} \approx \phi(x_i, y_j)$ for $i = 1, 2$ and a fixed $j$, which corresponds to a point $(x_i, y_j)$ in Category III. A simple Taylor expansion gives, for $i = 1, 2$,

$$\phi(x_i, y_j) = \phi(0, y_j) + ih\,\phi_x(0, y_j) + \frac{(ih)^2}{2}\phi_{xx}(0, y_j) + \mathcal{O}(h^3)$$

hence our desired approximation for the third order WENO scheme is

$$\phi_{i,j} = \phi(0, y_j) + ih\,\phi_x(0, y_j) + \frac{(ih)^2}{2}\phi_{xx}(0, y_j).$$

We already have $\phi(0, y_j) = g(y_j)$. The PDE (1.1), evaluated at the point $(0, y_j)$, becomes

$$H(\phi_x(0, y_j), g'(y_j)) = f(0, y_j) \tag{2.11}$$

in which the only unknown quantity is $\phi_x(0, y_j)$. Solving this (usually nonlinear) equation should give us $\phi_x(0, y_j)$. There might be more than one root, in which case we should choose the root so that

$$\partial_u H(\phi_x(0, y_j), g'(y_j)) > 0 \tag{2.12}$$

where $\partial_u$ refers to the partial derivative with respect to the first argument in $H(u, v)$. The condition (2.12) guarantees that the boundary $\Gamma$ in (2.10) is an inflow boundary. If the condition (2.12) still cannot pin down a root, then we would choose the root which is closest to the value from the first order fast sweeping solution at the same grid point. To obtain $\phi_{xx}(0, y_j)$, we first take the derivative with respect to $y$ on the original PDE (1.1), and then evaluate it at the point $(0, y_j)$, which yields

$$\partial_u H(\phi_x(0, y_j), g'(y_j))\phi_{xy}(0, y_j) + \partial_v H(\phi_x(0, y_j), g'(y_j))g''(y_j) = f_y(0, y_j), \tag{2.13}$$

where $\partial_u$ and $\partial_v$ refer to the partial derivatives with respect to the first and second arguments in $H(u, v)$, respectively. In this equation the only unknown quantity is $\phi_{xy}(0, y_j)$, hence we obtain easily its value, thanks to (2.12). We then take the derivative with respect to $x$ on the original PDE (1.1), and evaluate it at the point $(0, y_j)$ to obtain

$$\partial_u H(\phi_x(0, y_j), g'(y_j))\phi_{xx}(0, y_j) + \partial_v H(\phi_x(0, y_j), g'(y_j))\phi_{xy}(0, y_j) = f_x(0, y_j),$$

This time, the only unknown quantity is $\phi_{xx}(0, y_j)$, which we can obtain readily from this equality.

It is clear that this procedure can be carried out to any desired order of accuracy. Also, the inflow boundary $\Gamma$ in (2.10) can be any piece of a smooth curve: we only need to change the $x$ and $y$ partial derivatives to normal and tangential derivatives with respect to $\Gamma$. However, for this approach to work, $\Gamma$ can not consist of a single point.

## 3. Numerical Examples

In this section, we demonstrate the performance of the two approaches for treating the inflow boundary conditions described in Section 2 through a few two dimensional numerical examples. The third order fast sweeping WENO method [18], outlined in Section 2.1, is used. In our computation, the threshold value at which iteration stops is taken to be $\delta = 10^{-11}$. The small number in the WENO nonlinear weights $\varepsilon$ is taken as $10^{-6}$ unless otherwise stated.

Table 3.1: Example 1. Richardson extrapolation for the inflow boundary. $N$ is the number of mesh points in each direction.

| N | $L^1$ error | order | $L^\infty$ | order | iteration number |
|---|---|---|---|---|---|
| 40 | 8.00E-04 | | 6.36E-04 | | 40 |
| 80 | 5.92E-05 | 3.76 | 3.33E-05 | 4.25 | 30 |
| 160 | 3.64E-06 | 4.02 | 1.54E-06 | 4.44 | 38 |
| 320 | 4.00E-07 | 3.19 | 1.58E-07 | 3.28 | 50 |
| 640 | 4.98E-08 | 3.00 | 2.02E-08 | 2.97 | 81 |

**Example 1.**

We solve the Eikonal equation (1.1)-(1.2) with

$$f(x,y)=\frac{\pi}{2}\sqrt{\sin^2\left(\frac{\pi}{2}x\right)+\sin^2\left(\frac{\pi}{2}y\right)}.$$

The inflow boundary $\Gamma$ is the single point (0,0). The computational domain is $[-1,1]^2$. The exact solution for this problem is

$$\phi(x,y)=-\cos\left(\frac{\pi}{2}x\right)-\cos\left(\frac{\pi}{2}y\right).$$

Since the inflow boundary $\Gamma$ consists of a single point, the second strategy described in Section 2.3 does not apply. We use the first strategy described in Section 2.2 to handle the inflow boundary condition. Namely, in the small box $[-2h,\, 2h]^2$, we apply the first order fast sweeping method [19] with three different mesh sizes $h$, $h/2$ and $h/4$, and then use the Richardson extrapolation (2.9) to obtain a third order approximation to the grid values $\phi_{i,j}$ in this small box, which are then fixed as the initial values during the third order fast sweeping WENO process. Notice that this process has very little computational cost since the box $[-2h,\, 2h]^2$ is very small. For the outflow boundary, which is the boundary of the box $[-1,1]^2$, we take a simple third order extrapolation to provide solution values in the ghost points outside the computational domain. The results are given in Table 3.1. We can see clearly that the scheme with the numerical boundary treatment gives the correct order of accuracy.

**Example 2 (shape-from-shading).**

We solve the Eikonal equation (1.1)-(1.2) with

$$\text{case (a):}\quad f(x,y)=\sqrt{(1-|x|)^2+(1-|y|)^2}; \tag{3.1}$$

$$\text{case (b):}\quad f(x,y)=2\sqrt{y^2(1-x^2)^2+x^2(1-y^2)^2}. \tag{3.2}$$

The computational domain $\Omega=[-1,1]^2$. The inflow boundary for this example is the whole boundary of the box $[-1,1]^2$, namely $\Gamma=\{(x,y):\ |x|=1 \text{ or } |y|=1\}$. The boundary condition $\phi(x,y)=0$ is prescribed on $\Gamma$. For case (b), an additional boundary condition $\phi(0,0)=1$ is also prescribed at the center, see [13]. The exact solutions for these two cases are given by

$$\text{case (a):}\quad \phi(x,y)=(1-|x|)(1-|y|); \tag{3.3}$$

$$\text{case (b):}\quad \phi(x,y)=(1-x^2)(1-y^2). \tag{3.4}$$

For this example only, we set the parameter $\varepsilon$ in the nonlinear WENO weights (2.6)-(2.7) as $\varepsilon=10^{-6}h^2$. This smaller choice of $\varepsilon$ seems to make the adjustment of the nonlinear weights better near the center singularity of the solution, especially for case (b).

Table 3.2: Example 2. Richardson extrapolation for the inflow boundary. $N$ is the number of mesh points in each direction.

| N | $L^1$ error | order | iter | $L^1$ error | order | iter |
|---|---|---|---|---|---|---|
| | case (a) | | | case (b) | | |
| 80 | 3.45E-06 | | 21 | 3.38E-05 | | 35 |
| 160 | 3.04E-07 | 3.51 | 28 | 3.44E-06 | 3.30 | 50 |
| 320 | 2.58E-08 | 3.56 | 46 | 2.23E-07 | 3.95 | 81 |
| 640 | 2.14E-09 | 3.59 | 81 | 9.25E-09 | 4.59 | 149 |

Table 3.3: Example 2. Lax-Wendroff type procedure for the inflow boundary. $N$ is the number of mesh points in each direction.

| N | $L^1$ error | $L^\infty$ | iter | $L^1$ error | $L^\infty$ | iter |
|---|---|---|---|---|---|---|
| | case (a) | | | case (b) | | |
| 80 | 2.06E-14 | 9.26E-13 | 1 | 8.64E-07 | 9.99E-04 | 35 |
| 160 | 1.58E-14 | 2.34E-12 | 1 | 5.25E-08 | 2.34E-04 | 47 |
| 320 | 1.09E-14 | 5.49E-12 | 1 | 4.43E-15 | 4.26E-12 | 62 |
| 640 | 1.01E-14 | 1.08E-11 | 2 | 1.41E-15 | 1.14E-12 | 90 |

For this example, we can apply both of the strategies in Sections 2.2 and 2.3. We first apply the Richardson extrapolation strategy of Section 2.2 for the points inside the computational domain which are of distance at most $2h$ away from the inflow boundary $\Gamma$, using the results of the first order fast sweeping method with three different mesh sizes $h$, $h/2$ and $h/4$. For case (b), even though an additional point $\phi(0,0) = 1$ is prescribed at the center, we do not take any special treatment for points near the center. The results are given in Table 3.2. We can again see clearly that the scheme with this numerical boundary treatment gives the correct order of accuracy.

Next, we apply the Lax-Wendroff type procedure of Section 2.2 to obtain third order approximations to the values of the numerical solution corresponding to the points inside the computational domain which are of distance at most $2h$ away from the inflow boundary $\Gamma$. Again, for case (b), no special treatment has been done for points near the center. The results are given in Table 3.3. This time, since the solution is a polynomial of degree lower than the order of the scheme, we are able to obtain the exact solution with only round-off errors, as the Lax-Wendroff type procedure of Section 2.2 is able to prescribe the values to the points inside the computational domain which are of distance at most $2h$ away from the inflow boundary $\Gamma$ *exactly* by Taylor expansion[1)] . We remark that the first strategy of using the Richardson extrapolation in Section 2.2 is *not* able to provide the solution values to the points inside the computational domain which are of distance at most $2h$ away from the inflow boundary $\Gamma$ exactly, but only to the designed third order accuracy, hence the final fast sweeping WENO results in Table 3.2 are also only high order accurate but not exact to round-off errors.

[1)] We do note that for the coarser meshes in case (b) the solution has not settled to round-off errors. Our experiments show that this is related to the treatment of the center point: if we fix the numerical solution in the small box $[-2h, 2h]^2$ rather than just at the center point (0,0) by the exact solution, round-off error can be achieved. We will not further explore this difficulty as it is not related to the theme of this paper.

Table 3.4: Example 3. Lax-Wendroff type procedure for the inflow boundary. $N$ is the number of mesh points in each direction. The errors are measured in the computational domain but outside the box $[-0.15, 0.15]^2$.

| N | $L^1$ error | order | $L^\infty$ | order | iteration number |
|---|---|---|---|---|---|
| 80 | 0.573E-05 | | 0.129E-03 | | 25 |
| 160 | 0.122E-05 | 2.23 | 0.407E-05 | 4.98 | 32 |
| 320 | 0.191E-06 | 2.68 | 0.122E-05 | 1.74 | 46 |
| 640 | 0.246E-07 | 2.95 | 0.161E-06 | 2.92 | 62 |

Table 3.5: Example 4. Lax-Wendroff type procedure for the inflow boundary. $N$ is the number of mesh points in each direction. The errors are measured in the computational domain but outside the boxes $[-1.15,\ -0.85] \times [-0.15,\ 0.15]$, $[\sqrt{1.5} - 0.15,\ \sqrt{1.5} + 0.15] \times [-0.15, 0.15]$ and $[\sqrt{0.375} - 0.65,\ \sqrt{0.375} - 0.35] \times [-3,\ 3]$.

| N | $L^1$ error | order | $L^\infty$ | order | iteration number |
|---|---|---|---|---|---|
| 80 | 0.569E-02 | | 0.274E-02 | | 38 |
| 160 | 0.346E-03 | 4.04 | 0.766E-03 | 1.84 | 47 |
| 320 | 0.240E-04 | 3.85 | 0.294E-04 | 4.71 | 47 |
| 640 | 0.470E-05 | 2.35 | 0.336E-05 | 3.13 | 67 |

**Example 3.**

We solve the Eikonal equation (1.1)-(1.2) with $f(x, y) = 1$. The computational domain is $[-1, 1]^2$, and the inflow boundary $\Gamma$ is the unit circle of center (0,0) and radius 0.5, that is

$$\Gamma = \left\{(x, y):\ x^2 + y^2 = 0.25\right\}.$$

The boundary condition $\phi(x, y) = 0$ is prescribed on $\Gamma$. The exact solution for this problem is the distance function to the circle $\Gamma$. This exact solution has a singularity at the center of the circle to which the characteristics converge, hence we exclude the box $[-0.15, 0.15]^2$ when measuring the errors. For this problem, it is not easy to apply the Richardson extrapolation strategy in Section 2.2, since we use rectangular meshes and the inflow boundary $\Gamma$ is not on grid points. However, the Lax-Wendroff type procedure in Section 2.3 can be easily used to obtain the values of the numerical solution corresponding to the points inside the computational domain which have a horizontal or vertical distance less than $3h$ from the inflow boundary $\Gamma$. For the outflow boundary, which is the boundary of the box $[-1, 1]^2$, we take a simple third order extrapolation to provide solution values in the ghost points outside the computational domain. The results are given in Table 3.4. We can see happily again that the scheme with this numerical boundary treatment gives the correct order of accuracy away from the singularity at the center.

**Example 4.**

We solve the Eikonal equation (1.1)-(1.2) with $f(x, y) = 1$. The computational domain is $[-3, 3]^2$, and the inflow boundary $\Gamma$ consists of two circles of equal radius 0.5 with centers located at $(-1, 0)$ and $(\sqrt{1.5}, 0)$, respectively, that is

$$\Gamma = \left\{(x, y):\ (x + 1)^2 + y^2 = 0.25 \text{ or } (x - \sqrt{1.5})^2 + y^2 = 0.25\right\}.$$

The boundary condition $\phi(x, y) = 0$ is prescribed on $\Gamma$. The exact solution for this problem is the distance function to $\Gamma$. The exact solution for this problem is the distance function to the circle $\Gamma$. This exact solution has singularities at the centers of the circles and on the line that has the same distance to the two circles, on which the characteristics converge, hence we exclude the boxes $[-1.15, -0.85] \times [-0.15, 0.15]$, $[\sqrt{1.5} - 0.15, \sqrt{1.5} + 0.15] \times [-0.15, 0.15]$ and $[\sqrt{0.375} - 0.65, \sqrt{0.375} - 0.35] \times [-3, 3]$ when measuring the errors. Again, for this problem, it is not easy to apply the Richardson extrapolation strategy in Section 2.2, since we use rectangular meshes and the inflow boundary $\Gamma$ is not on grid points. However, the Lax-Wendroff type procedure in Section 2.3 can again be easily used to obtain the values of the numerical solution corresponding to the points inside the computational domain which have a horizontal or vertical distance less than $3h$ from the inflow boundary $\Gamma$. For the outflow boundary, which is the boundary of the box $[-3, 3]^2$, we take a simple third order extrapolation to provide solution values in the ghost points outside the computational domain. The results are given in Table 3.5. We can see happily again that the scheme with this numerical boundary treatment gives the correct order of accuracy away from the singularities.

## 4. Concluding Remarks

In this paper we have discussed two strategies to handle the inflow boundary conditions for high order fast sweeping methods for solving static Hamilton-Jacobi equations. The first method is based on Richardson extrapolation and a local application of a first order fast sweeping method with several different mesh sizes. The second method is based on a Lax-Wendroff type procedure to use repeatedly the PDE to obtain a high order Taylor expansion solution for grid points near the inflow boundary. Numerical examples are provided to demonstrate that both strategies work well and can provide the designed high order accuracy. The second method involves smaller extra computational cost and usually gives more accurate results, hence it is preferred if the inflow boundary allows (i.e., if the inflow boundary is not a single point or a set of isolated points). Even though the boundary treatments are only discussed in the context of the fast sweeping method, they (especially the second approach relying on the Lax-Wendroff-type procedure) are actually quite general, and can be applied to various high order numerical schemes for solving both static and time dependent PDEs. We will explore these issues in more detail in future work.

**Acknowledgments.** Research supported by NSFC grant 10671190. Additional support for the second author is provided by NSF grant DMS-0510345.

## References

[1] R. Abgrall, Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes, *Commun. Pur. Appl. Math.,* **49** (1996), 1339-1373.

[2] M. Boué and P. Dupuis, Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, *SIAM J. Numer. Anal.,* **36** (1999), 667-695.

[3] Y. Cheng and C.-W. Shu, A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations, *J. Comput. Phys.,* **223** (2007), 398-415.

[4] M. Crandall and P.L. Lions, Viscosity solutions of Hamilton-Jacobi equations, *Transactions of American Mathematical Society,* **277** (1983), 1-42.

[5] M. Crandall and P.L. Lions, Monotone difference approximations for scalar conservation laws, *Math. Comput.,* **34** (1984), 1-19.

[6] C. Hu and C.-W. Shu, A discontinuous Galerkin finite element method for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.,* **21** (1999), 666-690.

[7] L. Huang, S.C. Wong, M. Zhang, C.-W. Shu and W.H.K. Lam, Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm, *Transportation Research Part B,* submitted.

[8] G. Jiang and D.P. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.,* **21** (2000), 2126-2143.

[9] P.D. Lax and B. Wendroff, Systems of conservation laws, *Commun. Pur. Appl. Math.,* **13** (1960), 217-237.

[10] F. Li, C.-W. Shu, Y.-T. Zhang and H. Zhao, A second order discontinuous Galerkin fast sweeping method for Eikonal equations, *J. Comput. Phys.,* submitted.

[11] S. Osher and J. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.,* **79** (1988), 12-49.

[12] S. Osher and C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations, *SIAM J. Numer. Anal.,* **28** (1991), 907-922.

[13] E. Rouy and A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM J. Numer. Anal.,* **29** (1992), 867-884.

[14] C.-W. Shu, High order numerical methods for time dependent Hamilton-Jacobi equations, in Mathematics and Computation in Imaging Science and Information Processing, S.S. Goh, A. Ron and Z. Shen, Editors, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, volume 11, World Scientific Press, Singapore, 2007, pp.47-91.

[15] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.,* **77** (1988), 439-471.

[16] Y. Xia, S.C. Wong, M. Zhang, C.-W. Shu and W.H.K. Lam, An efficient discontinuous Galerkin method on triangular meshes for a pedestrian flow model, *Int. J. Numer. Meth. Eng.,* to appear.

[17] Y.-T. Zhang and C.-W. Shu, High order WENO schemes for Hamilton-Jacobi equations on triangular meshes, *SIAM J. Sci. Comput.,* **24** (2003), 1005-1030.

[18] Y.-T. Zhang, H.-K. Zhao and J. Qian, High order fast sweeping methods for static Hamilton-Jacobi equations, *J. Sci. Comput.,* **29** (2006), 25-56.

[19] H.-K. Zhao, A fast sweeping method for Eikonal equations, *Math. Comput.,* **74** (2005), 603-627.