Commun. Math. Res.
doi: 10.4208/cmr.2022-0015

Vol. **38**, No. 3, pp. 333-350
August 2022

Inverse Lax-Wendroff Boundary Treatment: a Survey

Chi-Wang Shu*

Division of Applied Mathematics, Brown University, Providence, RI 02912, USA.

Received 11 April 2022; Accepted 24 April 2022

Abstract. The inverse Lax-Wendroff (ILW) procedure is a numerical boundary treatment technique, which allows finite difference schemes and other schemes to achieve stability and high order accuracy when using cartesian meshes to solve boundary value problems defined on complex computational domain. In this short survey we summarize the main ingredients of the ILW procedure, discuss its applicability and stability properties, and provide possible directions of its future development.

AMS subject classifications: 65M06, 65M12

Key words: Inverse Lax-Wendroff procedure, boundary treatment, high order accuracy, stability, complex geometry.

1 Introduction

Finite difference methods are widely used to solve partial differential equations (PDEs). For example, to solve a hyperbolic equation

$$u_t + u_x = 0, \quad 0 \le x \le 1 \tag{1.1}$$

with the initial condition $u(x,0) = u^0(x)$ and the boundary condition

$$u(0,t) = g(t), \tag{1.2}$$

*Corresponding author. *Email address:* chi-wang_shu@brown.edu (C.-W. Shu)

a finite difference scheme approximates (1.1) on a spatial grid

$$0=x_0<x_1<\cdots<x_N=1 \tag{1.3}$$

and, for simplicity, we assume a uniform grid with the mesh size $\Delta x=x_{j+1}-x_j$. For example, we could use the following finite difference scheme:

$$u_j^{n+1}=au_{j-2}^n+bu_{j-1}^n+cu_j^n+du_{j+1}^n \tag{1.4}$$

with suitably chosen constants a,b,c and d (which depend on $\lambda=\frac{\Delta t}{\Delta x}$), approximating the PDE (1.1) to third order accuracy. Here and below, u_j^n is the numerical approximation at the grid point $x=x_j$ and at time $t=t^n$, and we assume, for simplicity, a constant time step size $\Delta t=t^{n+1}-t^n$.

The first difficulty associated with the boundary conditions of schemes such as (1.4) is the wide stencil, e.g., for the scheme (1.4) the stencil consists of four points $\{x_{j-2},x_{j-1},x_j,x_{j+1}\}$. Notice that the scheme (1.4) cannot be used to compute u_1^{n+1} and u_N^{n+1}, if we only have the information of u_j^n for $0\leq j\leq N$. We would need to either define the "ghost point" values u_{-1}^n and u_{N+1}^n and then use the scheme (1.4), or we could use a different scheme than (1.4) for the computation of u_1^{n+1} and u_N^{n+1}. In either case, we must analyze the stability and accuracy of the resulting approximations.

For a higher order finite difference scheme, the stencil is wider, and hence the number of such "abnormal" points near the boundary will be larger, causing more complications in either of the two approaches above.

The second difficulty associated with the boundary conditions of schemes such as (1.4) is the possibility that the boundary of the computational domain may not coincide with the grid points. For example, instead of the grid points defined in (1.3), we could imagine that the first grid point x_0 is not at $x=0$, for example $x_0=0.4\Delta x$. Such a configuration will make it difficult to apply the given boundary condition (1.2) which is defined at $x=0$. Of course, one might argue that such choice of grid points seems artificial. But even in one space dimension, such scenarios cannot be avoided if we are computing a moving (in time) domain with a fixed spatial grid. In two or higher dimensions, such scenarios will always happen if we are attempting to solve a PDE defined in a domain with complex geometry using cartesian meshes.

One of the major problems associated with the second difficulty is the possible appearance of small cells near the boundary, in the sense that the distance from the first grid point x_0 and the physical boundary $x=0$ is very small in comparison with the mesh size Δx. For many explicit schemes, the ratio of time step size over