

High order computational scheme for a dynamic continuum model for bi-directional pedestrian flows

Tao Xiong¹, Mengping Zhang², Chi-Wang Shu³, S.C. Wong⁴ and Peng Zhang⁵

Abstract

In this paper, we present a high-order weighted essentially non-oscillatory (WENO) scheme, coupled with a high-order fast sweeping method, for solving a dynamic continuum model for bi-directional pedestrian flows. We first review the dynamic continuum model for bi-directional pedestrian flows. This model is composed of a coupled system of a conservation law and an Eikonal equation. Then we present the first-order Lax-Friedrichs difference scheme with first order Euler forward time discretization, the third order WENO scheme with third order total variation diminishing (TVD) Runge-Kutta time discretization, and the fast sweeping method, and demonstrate how to apply them to the model under study. We present a comparison of the numerical results of the model from the first order and high-order methods, and conclude that the high-order methods are more efficient than the first order one, and they both converge to the same solution of the physical model.

keywords: dynamic continuum model, bi-directional flow, conservation law, Eikonal equation, WENO method, fast sweeping method.

1 Introduction

The continuum model for steady-state user equilibrium problems, with route choice behavior considered, has been developed [3, 25] which has a wide range of applications to, for example, the multi-commodity cost-flow relationship [24, 28], market share determination [27], elastic market externalities [31], the cordon-based congestion pricing problem [6], the combined multi-class, distribution and assignment model [5], the combined discrete and continuum model [23], and the housing problem [4]. For the dynamic macroscopic modeling of pedestrian flow problems, Hughes [12] provided a systematic framework without explicitly considering the user equilibrium concept and without providing a numerical solution procedure. To address this issue, a predictive user equilibrium model has been developed for describing the dynamic route choice behavior of pedestrians in which pedestrians were assumed to have perfect information to make their route choice decisions over time [7, 8, 9]. In contrast to the predictive user equilibrium model, the reactive dynamic user equilibrium model [1] describes the movement of pedestrians who do not have predictive information when they are changing their route and pedestrians have to rely on the instantaneous information available to them and make their choices in a reactive manner to minimize the walking cost to their destination. See [21] for the difference between predictive and reactive dynamic user equilibrium principles.

In [10], a pedestrian flow model based on the reactive dynamic user equilibrium principle is developed, where pedestrian density is governed by the scalar two-dimensional conservation law and the flow flux is implicitly dependent on the speed through an Eikonal equation. The weighted essentially non-oscillatory (WENO) scheme for the conservation law coupled with the fast sweeping method for the Eikonal equation on rectangular meshes has been designed to solve that model. In [29], another efficient method, the discontinuous Galerkin (DG) method for the conservation law coupled with the fast sweeping method for the Eikonal equation, which works nicely on triangular meshes, has been developed for the model.

¹Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: jingt@mail.ustc.edu.cn.

²Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: mpzhang@ustc.edu.cn.

³Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu.

⁴Department of Civil Engineering, The University of Hong Kong, Hong Kong, China. E-mail: hhccwsc@hku.hk.

⁵Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai 200072, P.R. China, E-mail: pzhang@mail.shu.edu.cn

The investigation of the flux or density distribution and the impact factor of the speed effect and path choices between two interactive streams of pedestrians, i.e., bi-directional pedestrian flows, has been performed in [11, 15]. The model for bi-directional pedestrian flows consists of a two-dimensional (2D) scalar hyperbolic conservation law equation coupled with an Eikonal equation for each group, which is extended from a single pedestrian type reactive dynamic user equilibrium model [10]. In [11], the model for two pedestrian flows walking in opposite directions in a continuous walking facility has been developed and the look-ahead behavior in the model induces a viscosity effect on movement patterns, hence the source term in the Eikonal equation is dependent on the solution of the equation and cannot be seen as known input. The first order finite difference scheme for the conservation law coupled with the pseudo time-marching approach for the Eikonal equation on rectangular meshes has been designed to solve the model. In [15], two pedestrian flows walking in crossing paths have been considered but without the look-ahead behavior, hence the source term in the Eikonal equation is only dependent on the position and time, and supplied as a known input. The finite volume method for the conservation law coupled with the fast marching method for the Eikonal equation on triangular meshes has been designed to solve the model. We remark that both of these works used only low order methods.

In this paper, we design a high-order computational scheme for the model in [15], that is the third order weighted essentially non-oscillatory (WENO) scheme for the conservation law with third order total variation diminishing (TVD) time discretization, coupled with the third order fast sweeping method for the Eikonal equation on rectangular meshes. We remark that for each component of the current model (conservation law, Eikonal equation), many algorithms (e.g. [22, 17, 13]) have been developed with different advantages and limitations. It is not the purpose of this paper to perform a comprehensive comparison of all existing methods. Rather, we concentrate on a specific combination of third order methods for different components of the model and study its accuracy and efficiency in relation to the first order solver. For a high order method with complicated boundaries, appropriate numerical boundary settings seems to be very important. It is also necessary to pay special attention to the fast sweeping procedure in order to ensure its convergence and non-oscillatory property. We will compare the results of the high order scheme with a first order Lax-Friedrichs difference scheme for the conservation law coupled with a first order fast sweeping method for the Eikonal equation on rectangular meshes, and demonstrate that with convergent solutions, the high-order scheme are more efficient than the low-order one. We remark that high order accurate numerical methods can often save both computational time and memory in achieving comparable precision than lower order methods. This is especially the case for problems containing complex solutions. We have included a numerical example with a simplified setup and known exact solutions, to verify the advantage of using the higher order scheme to reach higher precision with less CPU time.

The rest of the paper is organized as follows. In Section 2, the model formulation of bi-directional pedestrian flows is described. The computational schemes for solving these models are given in Section 3. Numerical examples are given in Section 4, including one with a simplified setup and with known exact solutions to demonstrate the convergence and efficiency of the high order schemes. Section 5 gives concluding remarks.

2 Model Formulation

We use the following nomenclature:

- (1) Ω represents a 2D continuous walking facility (in m^2).
- (2) Group c represents the c -th type of pedestrians marching toward the c -th destination of the walking facility.
- (3) Γ is the boundary of Ω (in m), Γ_o^c the original segment from which Group c enters into the walking facility (in m), Γ_d^c the destination segment from which Group c leaves the walking facility (in m), and Γ_h^c the wall segment from which nobody in Group c is allowed to enter or leave the walking facility (in m).
- (4) T is the time horizon (in s).
- (5) $q^c(x, y, t)$ expresses the number of pedestrians who cross a unit width of Γ_o^c and describes the time-varying demand of Group c (in $ped/m/s$).
- (6) $\rho^c(x, y, t)$ is a time-varying function that denotes the pedestrian density of Group c (in ped/m^2).
- (7) $\mathbf{F}^c(x, y, t) := (f_1^c(x, y, t), f_2^c(x, y, t))$ represents the flow vector for Group c , where $f_1^c(x, y, t)$ is the

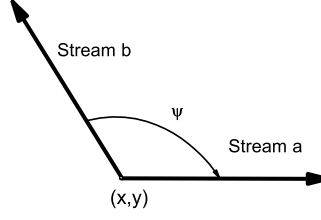


Figure 1: Angle Ψ between Streams a and b .

flow flux in the x -direction and $f_2^c(x, y, t)$ is the flow flux in the y -direction (in $ped/m/s$). By default, $\mathbf{F}^c(x, y, t) \cdot \mathbf{n} = 0$, where \mathbf{n} denotes the unit outward normal of Γ_h^c .

(8) $v^c(x, y, t)$ is the local walking speed of Group c (in m/s).

(9) $\tau^c(x, y, t)$ is the local cost per unit distance of movement experienced by a Group c pedestrian at location (x, y) and time t (in $/m$). If this cost is represented by the travel time as the reciprocal of speed, then the unit becomes s/m .

(10) $\Phi^c(x, y, t)$ is the instantaneous walking time potential of Group c from location (x, y) to Γ_d^c , at which $\Phi^c = 0$ (in s). This measures the total walking time along the instantaneous shortest path between location (x, y) and destination based on the traffic condition at snapshot moment t .

Finally, $\Gamma = \Gamma_o^c \cup \Gamma_d^c \cup \Gamma_h^c$, $c \in \{a, b\}$, $(x, y) \in \Omega$ and $t \in T$.

2.1 Flow Conservation Equation

The density, velocity and flow of pedestrians follow the physical principle of mass conservation, which is given by

$$\rho_t^c(x, y, t) + \nabla \cdot \mathbf{F}^c(x, y, t) = 0, \quad (1)$$

where $\rho_t^c(x, y, t) = \frac{\partial \rho^c(x, y, t)}{\partial t}$, and $\nabla \cdot \mathbf{F}^c(x, y, t) = \frac{\partial f_1^c(x, y, t)}{\partial x} + \frac{\partial f_2^c(x, y, t)}{\partial y}$ for commodity $c \in \{a, b\}$.

2.2 Walking Speed Function

The local walking speeds of the two groups depend on the densities and traveling directions of the two pedestrians in the walking facility and are determined by

$$v^a(\tilde{\rho}, \Psi) = v_f \exp(-\alpha(\rho^a + \rho^b)^2) \exp(-\beta(1 - \cos(\Psi))(\rho^b)^2), \quad (2)$$

$$v^b(\tilde{\rho}, \Psi) = v_f \exp(-\alpha(\rho^a + \rho^b)^2) \exp(-\beta(1 - \cos(\Psi))(\rho^a)^2), \quad (3)$$

where $\tilde{\rho} := \{\rho^a(x, y, t), \rho^b(x, y, t)\}$. Here, v_f is the free-flow walking speed of pedestrians, α, β , are the model parameters, and $\Psi := \Psi(x, y, t)$ is the intersecting angle between the two pedestrian streams of Group a and Group b at location (x, y) , as shown in Figure 1 (see [26] for the empirical evidence of this functional form for bi-directional pedestrian flow problems).

2.3 Local Cost Function

The local cost τ^c , as a function of location and time, is assumed to depend on walking speed alone and can be specified as

$$\tau^c(x, y, t) : = \tau(v^c(\tilde{\rho}, \Psi)) = \frac{1}{v^c(\tilde{\rho}, \Psi)}, \quad (4)$$

which describes the walking time per unit distance of movement incurred by a pedestrian in Group c in the walking facility.

2.4 Flow Definition

For each Group $c \in \{a, b\}$, along the direction of the flow vector (or movement), flow intensity, which is determined as the norm of the pedestrian flow, is equal to the product of speed and density,

$$\| \mathbf{F}^c(x, y, t) \| = v^c(\tilde{\rho}, \Psi) \rho^c(x, y, t), \quad (5)$$

where $\| \mathbf{F}^c(x, y, t) \| = (f_1^c(x, y, t)^2 + f_2^c(x, y, t)^2)^{\frac{1}{2}}$.

2.5 Path Choice Constraints

It is assumed that a pedestrian in each group at location $(x, y) \in \Omega$ walks in a route choice to minimize his or her travel cost to the destination, based on the instantaneous travel cost information that is available at the time of making a decision [10, 15]. Then, we have

$$\tau^c(x, y, t) \frac{\mathbf{F}^c(x, y, t)}{\| \mathbf{F}^c(x, y, t) \|} + \nabla \Phi^c(x, y, t) = 0, \quad (6)$$

where $\nabla \Phi^c = (\frac{\partial \Phi^c}{\partial x}, \frac{\partial \Phi^c}{\partial y})$. From (4) - (6), we have

$$\begin{aligned} \mathbf{F}^c(x, y, t) &= -\frac{v^c(\tilde{\rho}, \Psi) \rho^c(x, y, t)}{\tau^c(x, y, t)} \nabla \Phi^c(x, y, t) \\ &= -\rho^c(x, y, t) (v^c(\tilde{\rho}, \Psi))^2 \nabla \Phi^c(x, y, t), \end{aligned} \quad (7)$$

and

$$\| \nabla \Phi^c(x, y, t) \| = \tau^c(x, y, t). \quad (8)$$

From (7), we have

$$\mathbf{F}^c(x, y, t) // (-\nabla \Phi^c(x, y, t)), \quad (9)$$

with $\rho^c(x, y, t) \neq 0$, which means that the two vectors, $\mathbf{F}^c(x, y, t)$ and $\nabla \Phi^c(x, y, t)$, are parallel but pointing in the opposite direction.

From (6) - (9), for any used path p and for an unused path \tilde{p} , we have the following property

$$\bar{\tau}_p^c(x, y, t) = \int_p \tau^c(x, y, t) ds = \Phi^c(x, y, t), \quad (10)$$

$$\bar{\tau}_{\tilde{p}}^c(x, y, t) = \int_{\tilde{p}} \tau^c(x, y, t) ds \geq \Phi^c(x, y, t) \quad (11)$$

which means that the total instantaneous travel cost at time $t \in T$ is independent of the used paths, and for any unused paths, the total instantaneous travel cost is greater than or equal to that of the used paths. In this way, the model guarantees that pedestrians choose their paths in the walking facility in a user-optimal manner with respect to the instantaneous travel information.

The model can now be formulated as the following set of differential equations.

$$\begin{cases} \rho_t^c(x, y, t) + \nabla \cdot \mathbf{F}^c(x, y, t) &= 0, \\ \mathbf{F}^c(x, y, t) &= -\rho^c(x, y, t) (v^c(\tilde{\rho}, \Psi))^2 \nabla \Phi^c(x, y, t), \\ \| \nabla \Phi^c(x, y, t) \| &= \tau^c(x, y, t), \end{cases} \quad (12)$$

where $\tau^c(x, y, t)$ are determined by equations (2), (3) and (4), respectively. The angle Ψ in (2) can be obtained by two potential functions due to the formula (7), and we have

$$\cos(\Psi) = \frac{\nabla \Phi^a(x, y, t)}{\| \nabla \Phi^a(x, y, t) \|} \cdot \frac{\nabla \Phi^b(x, y, t)}{\| \nabla \Phi^b(x, y, t) \|}.$$

The system (12) is subject to the following initial boundary conditions:

$$\mathbf{F}^c(x, y, t) \cdot \mathbf{n}(x, y) = q^c(x, y, t), \forall (x, y) \in \Gamma_o^c, \quad (13)$$

$$\rho^c(x, y, 0) = \rho_0^c(x, y), \quad \forall (x, y) \in \Omega, \quad (14)$$

$$\Phi^c(x, y, t) = 0, \quad \forall (x, y) \in \Gamma_d^c, \quad (15)$$

where $\mathbf{n}(x, y)$ is a unit outer normal vector that points out of the domain boundary, and $\rho_0^c(x, y)$ is the initial pedestrian density for group c .

3 Computational Schemes

In this section, we describe an efficient method for solving the model of system (12), which includes the third order weighted essentially non-oscillatory (WENO) scheme for the conservation law equation (1), the fast sweeping method based on the third order WENO scheme for the Eikonal equation (8) and the third order TVD Runge-Kutta time discretization for the coupled system of equations. The method is similar to that in [10], however, additional difficulties arise for the bi-directional flow considered in this work, especially for boundary treatments and for convergence of the fast sweeping iterations, which we will address. In order to highlight the high order method, we will compare the results of the third order method for all equations with the first order method for all equations. The first order method consists of the first order Lax-Friedrichs difference scheme for the conservation law equation, the first order fast sweeping method for the Eikonal equation and the Euler forward time discretization for the coupled system of equations, which will also be described here.

3.1 Schemes for the Conservation Law

For the 2D mass conservation equation (1) of the system, the semi-discrete scheme approximates the point values $\rho_{i,j}^c \approx \rho^c(x_i, y_j, t)$ through a conservative difference formula as follows. To simplify notation, we drop the superscript c where there is no confusion

$$\frac{d}{dt}\rho_{i,j} = -\frac{1}{\Delta x}((\hat{f}_1)_{i+\frac{1}{2},j} - (\hat{f}_1)_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}((\hat{f}_2)_{i,j+\frac{1}{2}} - (\hat{f}_2)_{i,j-\frac{1}{2}}) \quad (16)$$

where Δx and Δy are the mesh sizes in x and y , respectively, which are assumed to be uniform $\Delta x = \Delta y = h$ for simplicity. $(\hat{f}_1)_{i+\frac{1}{2},j}$ and $(\hat{f}_2)_{i,j+\frac{1}{2}}$ are the numerical fluxes in the x and y directions, respectively. We describe the details of the definition for the x flux $(\hat{f}_1)_{i+\frac{1}{2},j}$ below. The definition for the y flux $(\hat{f}_2)_{i,j+\frac{1}{2}}$ is analogous. When we are computing the x flux $(\hat{f}_1)_{i+\frac{1}{2},j}$, the y index j is fixed. Again, for simplicity of notation, we drop the y index j below.

3.1.1 The first order Lax-Friedrichs difference scheme

The first order Lax-Friedrichs difference scheme to approximate the numerical flux $(\hat{f}_1)_{i+\frac{1}{2}}$ in (16) can be defined as

$$(\hat{f}_1)_{i+\frac{1}{2}} = \frac{1}{2}((f_1)_i + (f_1)_{i+1} - \alpha(\rho_{i+1} - \rho_i)) \quad (17)$$

where $\alpha = |f_1'(\rho)|$.

3.1.2 The third order WENO scheme

We summarize the third-order WENO scheme in this subsection [16, 14, 18]. The basic idea of the WENO scheme is the locally adaptive choice of the approximation stencil, so that high order accuracy is achieved in smooth regions and discontinuities are resolved in a sharp and non-oscillatory fashion. Here we use the third order WENO scheme of [14] to approximate the numerical flux $(\hat{f}_1)_{i+\frac{1}{2}}$ in (16).

We first assume $f_1'(\rho) \geq 0$, namely, a positive wind direction, to simplify the description. The general case is commented upon later. In this case, the numerical flux $(\hat{f}_1)_{i+\frac{1}{2}}$ is obtained through a one-point upwind-biased stencil containing $s_l = f_1(x_l, y_j, t)$ for $l = i-1, i, i+1$. The numerical flux $(\hat{f}_1)_{i+\frac{1}{2}}$ is given by the convex combination of two second-order fluxes as

$$(\hat{f}_1)_{i+\frac{1}{2}} = \omega_0 \hat{s}_{i+\frac{1}{2}}^{(0)} + \omega_1 \hat{s}_{i+\frac{1}{2}}^{(1)} \quad (18)$$

where the two second-order numerical fluxes are based on two different sub-stencils given by

$$\hat{s}_{i+\frac{1}{2}}^{(0)} = \frac{1}{2}s_i + \frac{1}{2}s_{i+1} \quad (19)$$

$$\hat{s}_{i+\frac{1}{2}}^{(1)} = -\frac{1}{2}s_{i-1} + \frac{3}{2}s_i \quad (20)$$

and the nonlinear weights ω_0 and ω_1 in (18) are defined by

$$\omega_0 = \frac{\alpha_0}{\alpha_0 + \alpha_1}, \quad \omega_1 = \frac{\alpha_1}{\alpha_0 + \alpha_1} \quad (21)$$

where

$$\alpha_0 = \frac{d_0}{(\epsilon + \beta_0)^2}, \quad \alpha_1 = \frac{d_1}{(\epsilon + \beta_1)^2} \quad (22)$$

ϵ is a parameter to prevent the denominator from becoming 0, and is taken as 10^{-8} in our simulation. d_0 and d_1 are the linear weights given by

$$d_0 = \frac{2}{3}, \quad d_1 = \frac{1}{3}$$

and the smoothness indicators β_0 and β_1 , which measure the smoothness of the approximation in the substencils, are given by

$$\beta_0 = (s_{i+1} - s_i)^2, \quad \beta_1 = (s_i - s_{i-1})^2.$$

For the general case where $f'_1(\rho)$ may change sign, first we form a flux splitting $f_1(\rho) = f_1^+(\rho) + f_1^-(\rho)$, such that $\frac{d}{dt}f_1^+(\rho) \geq 0$ and $\frac{d}{dt}f_1^-(\rho) \leq 0$. We then perform the procedure above to obtain the numerical flux corresponding to $f_1^+(\rho)$, and use a mirror symmetric procedure, with respect to the location $x_{i+\frac{1}{2}}$ to obtain the numerical flux corresponding to $f_1^-(\rho)$. These two numerical fluxes are then combined to form the final numerical flux. Several flux splittings can be used in practice. We use the local Lax-Friedrichs (LLF) splitting here, which is given by

$$f_1^\pm(\rho) = \frac{1}{2}(f_1(\rho) \pm \alpha(\rho)\rho), \quad \text{where } \alpha(\rho) = \max_{\rho \in [\rho^-, \rho^+]} |f'_1(\rho)|.$$

Here, the range to take the maximum value for computing $\alpha(\rho)$ depends on the stencil: if the stencil $S = \{i-2, i-1, i, i+1, i+2\}$ is used, then

$$\rho^- = \min\{\rho_{i-2}, \rho_{i-1}, \rho_i, \rho_{i+1}, \rho_{i+2}\}, \quad \rho^+ = \max\{\rho_{i-2}, \rho_{i-1}, \rho_i, \rho_{i+1}, \rho_{i+2}\}.$$

However, in solving the system (12), $f_1(\rho)$ is not an explicit function of ρ (see the second equation of (12)). If we assume that the velocity v and the potential ϕ are not dependent on ρ , then we have $f'_1(\rho) = -v^2\phi_x$, and

$$\alpha(\rho) = \max_{l \in \{i-2, i-1, i, i+1, i+2\}} |-v_l^2(\phi_x)_l|.$$

In our computation, we use

$$\alpha(\rho) = 3 \max_{l \in \{i-2, i-1, i, i+1, i+2\}} |-v_l^2(\phi_x)_l|$$

which seems to be able to better control spurious oscillations.

3.2 The Fast Sweeping Method for the Eikonal Equation

The Eikonal equation is a special case of steady-state Hamilton-Jacobi equations. Solutions to Hamilton-Jacobi equations are usually continuous but not everywhere differentiable and are usually not unique. The viscosity solution [2] is the physically relevant solution and numerical solutions for approximating the viscosity solutions of the Hamilton-Jacobi equations follow the lines similar to those for solving conservation laws. Thus the WENO schemes for conservation laws are extended to WENO schemes for Hamilton-Jacobi equations [13, 19]. For the steady-state equation (8), if we use the time-dependent WENO scheme of [13, 19], we would need to introduce a pseudo-time and then march to a steady state for each fixed t . The recently developed fast sweeping method [33] is much faster in computer time than the pseudo-time marching. We use the fast sweeping method to solve the Eikonal equation (8) here. The first and third order fast sweeping methods can be found in [32] and [33], respectively. Similarly, we drop the superscript c for simplicity of description.

The fast sweeping method starts with the following initialization. Based on the boundary condition $\phi(x, y) = 0$ for $(x, y) \in \Gamma_d$, we assign the exact boundary values on Γ_d . For the first order fast sweeping

method, large values (for example 10^6) is assigned as the initial guess at all other grid points. For the third-order fast sweeping method, the solution from the first-order fast sweeping method is used as the initial guess at all other grid points.

The following Gauss-Seidel iterations with four alternating direction sweepings are performed after initialization

$$\begin{aligned} (1) \quad & i = 1 : N_x, \quad j = 1 : N_y; & (2) \quad & i = N_x : 1, \quad j = 1 : N_y; \\ (3) \quad & i = N_x : 1, \quad j = N_y : 1; & (4) \quad & i = 1 : N_x, \quad j = N_y : 1 \end{aligned} \quad (23)$$

where (i, j) is the grid index pair in (x, y) and N_x and N_y are the number of grid points in x and y , respectively. When we loop to a point (i, j) , the solution is updated as follows, using the Godunov Hamiltonian

$$\phi_{i,j}^{new} = \begin{cases} \min(\phi_{i,j}^{xmin}, \phi_{i,j}^{ymain}) + \tau_{i,j} h, & \text{if } |\phi_{i,j}^{xmin} - \phi_{i,j}^{ymain}| \leq \tau_{i,j} h, \\ \frac{\phi_{i,j}^{xmin} + \phi_{i,j}^{ymain} + (2\tau_{i,j}^2 h^2 - (\phi_{i,j}^{xmin} - \phi_{i,j}^{ymain})^2)^{\frac{1}{2}}}{2}, & \text{otherwise} \end{cases} \quad (24)$$

where $\tau_{i,j} = \tau(x_i, y_j, t)$.

For a first-order fast sweeping method, we define $\phi_{i,j}^{xmin}$ and $\phi_{i,j}^{ymain}$ in (24) as

$$\begin{cases} \phi_{i,j}^{xmin} = \min(\phi_{i-1,j}, \phi_{i+1,j}) \\ \phi_{i,j}^{ymain} = \min(\phi_{i,j-1}, \phi_{i,j+1}) \end{cases} \quad (25)$$

For a third-order fast sweeping method, we define $\phi_{i,j}^{xmin}$ and $\phi_{i,j}^{ymain}$ in (24) as

$$\begin{cases} \phi_{i,j}^{xmin} = \min(\phi_{i,j}^{old} - h(\phi_x)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_x)_{i,j}^+) \\ \phi_{i,j}^{ymain} = \min(\phi_{i,j}^{old} - h(\phi_y)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_y)_{i,j}^+) \end{cases} \quad (26)$$

with

$$(\phi_x)_{i,j}^- = (1 - w_-) \left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \right) + w_- \left(\frac{3\phi_{i,j} - 4\phi_{i-1,j} + \phi_{i-2,j}}{2h} \right), \quad (27)$$

$$(\phi_x)_{i,j}^+ = (1 - w_+) \left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \right) + w_+ \left(\frac{-3\phi_{i,j} + 4\phi_{i+1,j} - \phi_{i+2,j}}{2h} \right), \quad (28)$$

$$w_- = \frac{1}{1 + 2r_-^2}, \quad r_- = \frac{\epsilon + (\phi_{i,j} - 2\phi_{i-1,j} + \phi_{i-2,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}, \quad (29)$$

$$w_+ = \frac{1}{1 + 2r_+^2}, \quad r_+ = \frac{\epsilon + (\phi_{i,j} - 2\phi_{i+1,j} + \phi_{i+2,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}. \quad (30)$$

The nonlinear weights are the same as those in (21), as well as the meaning of ϵ , only that they are written in a different form as in [13] for easier implementation. The definitions for $(\phi_y)_{i,j}^-$ and $(\phi_y)_{i,j}^+$ are of course analogous.

Convergence is declared if

$$\|\phi^{new} - \phi^{old}\| \leq \delta, \quad (31)$$

where δ is a given convergence threshold value. $\delta = 10^{-11}$ and L^1 norm are used in our computation.

We can also obtain ϕ_x from solving (8). For the first-order fast sweeping method, $(\phi_x)_{i,j}$ is computed according to (25)

$$\begin{cases} (\phi_x)_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h}, & \text{if } \phi_{i,j}^{xmin} = \phi_{i-1,j} \\ (\phi_x)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h}, & \text{if } \phi_{i,j}^{xmin} = \phi_{i+1,j} \end{cases} \quad (32)$$

and for third-order fast sweeping method, $(\phi_x)_{i,j}$ is computed from (27) and (28) according to (26), that is

$$\begin{cases} (\phi_x)_{i,j} = (\phi_x)_{i,j}^-, & \text{if } \phi_{i,j}^{xmin} = \phi_{i,j}^{old} - h(\phi_x)_{i,j}^- \\ (\phi_x)_{i,j} = (\phi_x)_{i,j}^+, & \text{if } \phi_{i,j}^{xmin} = \phi_{i,j}^{old} + h(\phi_x)_{i,j}^+ \end{cases} \quad (33)$$

the computation for $(\phi_y)_{i,j}$ is analogous.

For the third order fast sweeping method, the nonlinear weights in (27) and (28) may cause non-convergence for (31) in some time steps for solving the system (12) of our model. For such cases, the iteration does converge if linear weights are used instead, however this would bring oscillatory results without the advantage of choosing smooth stencils from the application of nonlinear weights, especially around the boundary where discontinuities exist. We have therefore adopted the following strategy. First, we use the nonlinear weights to iterate. After the iteration error stagnates for 50 consecutive steps at the same error level, for example at 10^{-8} , we switch to fixed values for the weights taken as the average values of the nonlinear weights in those 50 iteration steps. These fixed weights are then used in (27) and (28) instead of the nonlinear weights (29) and (30) afterwards, which leads to very fast convergence according to (31).

3.3 The Third-order TVD Runge-Kutta Time Discretization

Finally, the semi-discrete scheme (16) must also be discretized in time. For the first order method, an Euler forward time discretization is used. For the third order WENO method, we use the third-order total-variation-diminishing (TVD) Runge-Kutta method [20, 18], which is the convex combination of three Euler forward time discretization steps and can maintain the stability of the spatial discretization. If we denote the right-hand side of the ordinary differential equation system (16) by $L(\rho)_{i,j}$, a first order Euler forward time stepping is

$$\rho^{n+1} = \rho^n + \Delta t L(\rho^n) \quad (34)$$

and a third-order TVD Runge-Kutta method is given by

$$\rho^{(1)} = \rho^n + \Delta t L(\rho^n) \quad (35)$$

$$\rho^{(2)} = \frac{3}{4}\rho^n + \frac{1}{4}(\rho^{(1)} + \Delta t L(\rho^{(1)})) \quad (36)$$

$$\rho^{n+1} = \frac{1}{3}\rho^n + \frac{2}{3}(\rho^{(2)} + \Delta t L(\rho^{(2)})). \quad (37)$$

Here, the time step Δt needs to satisfy the Courant-Friedrichs-Lewy (CFL) condition. We take the CFL coefficient to be 0.1 in our computation.

3.4 Solution Procedure

Starting from the density ρ^n at time level n , we obtain the density ρ^{n+1} by the following steps.

1. Obtain the cost function $\tau^c(x, y, t)$ by formula (2)-(4);
2. Solve the Eikonal equation (8) using the fast sweeping method to obtain ϕ^c and $\nabla\phi^c$;
3. Obtain the flux \mathbf{F}^c by using the formula (7); and
4. Use the first-order Lax-Friedrichs difference scheme with an Euler forward time discretization (34) or the third-order LLF WENO scheme with the third-order TVD Runge-Kutta time discretization (35)-(37) to obtain ρ^{n+1} by solving the conservation law (1).

The procedure repeats until it marches to the end of the analysis period.

4 Numerical Experiment

In this section, we first design a numerical example to model (12) to demonstrate the convergence and efficiency of the high order scheme. We then provide another numerical example with a simplified setup to model (12) which has a known exact solution, to demonstrate quantitatively the advantage of higher order schemes in reaching higher precision with less CPU time than the lower order scheme.

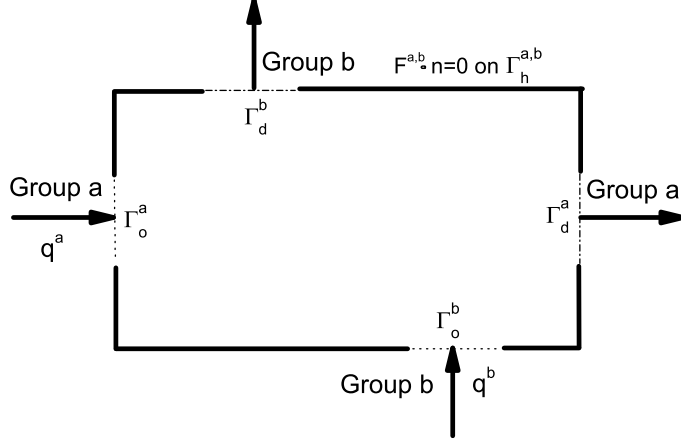


Figure 2: Overview of the railway platform.

4.1 Example 1

We consider a railway platform that is $100m$ by $50m$ in size, as shown in Figure 2. Group c enters the platform from segment Γ_o^c with a width of $20m$ and leaves the platform from segment Γ_d^c with a width of $20m$.

The time horizontal is set to be $240s$. Initially, the platform is empty. The inlet flow rate, $q^c(x, y, t)$, on Γ_o^c represents time-varying demand and is given as follows.

$$q^a(0, y, t) = \begin{cases} \frac{t}{30} \times m_a, & t \in [0, 30], \\ m_a, & t \in [30, 90], \\ -m_a \times \frac{t-120}{30}, & t \in [90, 120], \\ 0, & t \in [120, 300]. \end{cases}, \quad \forall y \in [15, 35],$$

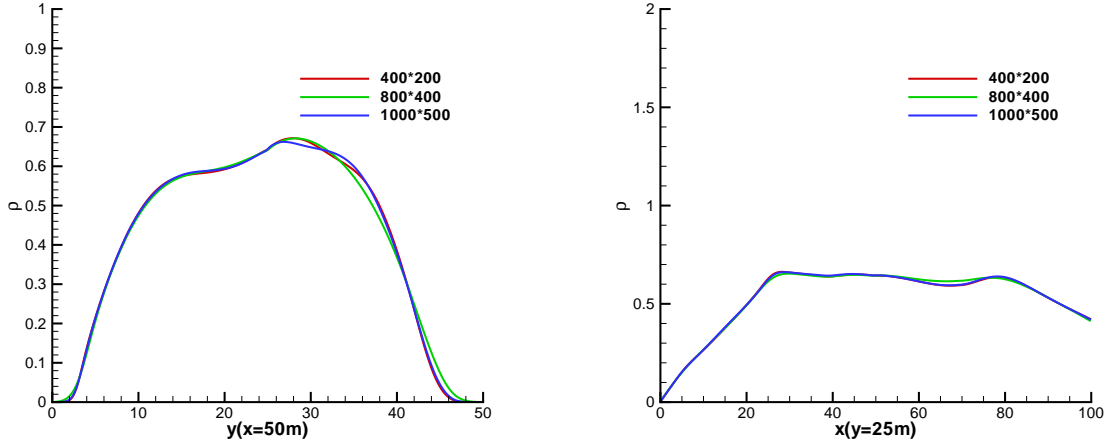
and

$$q^b(x, 0, t) = \begin{cases} \frac{t}{30} \times m_b, & t \in [0, 30], \\ m_b, & t \in [30, 90], \\ -m_b \times \frac{t-120}{30}, & t \in [90, 120], \\ 0, & t \in [120, 300]. \end{cases}, \quad \forall x \in [65, 85],$$

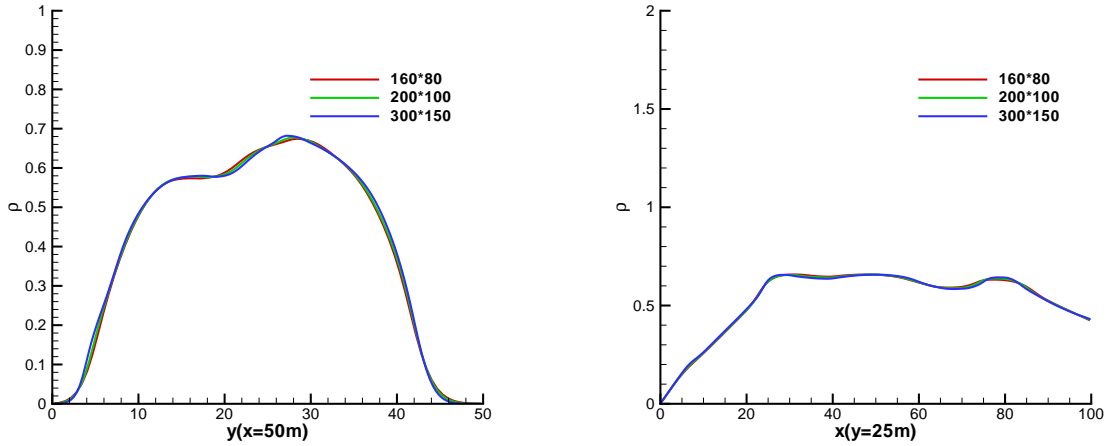
where m_a and m_b denote, respectively, the peaks of the inlet flow rate at the two entrances and we take $m_a = 1.0$ and $m_b = 0.5$ for the calculation. Then, the flow vector $\mathbf{F}^a = (q^a, 0)$ on Γ_o^a and $\mathbf{F}^b = (0, q^b)$ on Γ_o^b . The free-flow walking speed v_f is taken as $1.034m/s$ and two model parameters, α and β in (2) and (3), are taken as 0.075 and 0.019 , respectively (see [26]).

The numerical experiments on rectangular meshes are tested. The first order Lax-Friedrichs difference scheme for the conservation law coupled with the first order fast sweeping method (FSM) is denoted by “b11”, and the third order WENO scheme for the conservation law coupled with the third order FSM is denoted by “b33”.

For the high order scheme of this model, we have to assign values on ghost points, which are the grid points of not more than $2h$ away from the boundary outside the computational domain. There are several variables (more than three) in the computation and only three of them need ghost point values for the scheme: density ρ , flux $\mathbf{F} = (f_1, f_2)$ and potential ϕ . The boundary Γ is composed of the inlet Γ_o , the outlet Γ_d and the wall Γ_h , that is $\Gamma = \Gamma_o \cup \Gamma_d \cup \Gamma_h$. For the density ρ , the third order extrapolation is used on Γ_o and Γ_d , and $\rho = 0$ on Γ_h . For the flux $F = (f_1, f_2)$, on Γ_o it takes the exact values, the third order extrapolation is used on Γ_d , and we take 0 on Γ_h . For the potential ϕ , $\phi = 0$ on



(a) $t = 120$ s



(b) $t = 120$ s

Figure 3: Cuts for ρ of Group a at $t = 60s, 90s$ and $120s$ with three different mesh sizes . Left: cuts at $x = 50m$; Right: cuts at $y = 25m$. (a) for “b11”, (b) for “b33”.

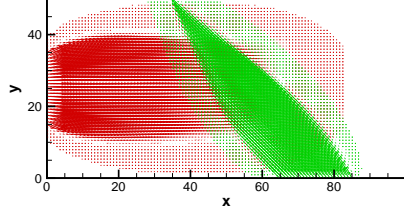
Γ_d is the exact solution, the third order extrapolation is used on Γ_o , and equal value of ϕ from the first point inside the computational domain is set on the wall Γ_h .

4.1.1 Convergence of the Low-order and High-order Numerical Results

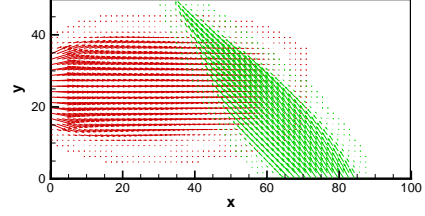
In this subsection, we show that the low-order scheme and the high-order scheme are both convergent, where the low-order scheme is the first-order Lax-Friedrichs scheme for the conservation law coupled with the first-order fast sweeping method for the Eikonal equation and the high-order scheme is the third order WENO scheme for the conservation law coupled with the third order fast sweeping method for the Eikonal equation. The results are consistent with those in [15].

We compare the density curves of Group a along the center line, $x = 50m$ and $y = 25m$, of the computational domain at several different mesh sizes for “b11” and “b33” in Figure 3, which could convince us the convergence for both of the schemes.

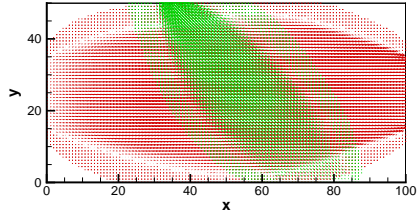
In Figure 4, we depict the flow vector $F = (f_1, f_2)$ for “b11” at the mesh size 800×400 and for “b33” at the mesh size 200×100 . High-density values would be seen near the two exits as the pedestrians arrive there and have to decrease their speed as they line up to walk through the relatively narrow exits.



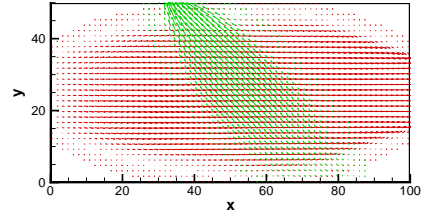
(a) $t = 60s$



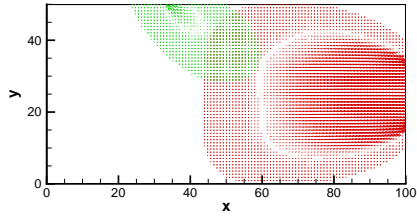
(b) $t = 60s$



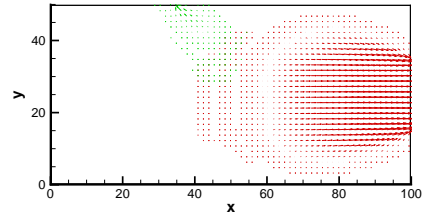
(c) $t = 120s$



(d) $t = 120s$



(e) $t = 180s$



(f) $t = 180s$

Figure 4: Flow vector plots. Left: "b11" at the mesh size 800×400 ; Right: "b33" at the mesh size 200×100 . Red: Group a ; Green: Group b .

4.1.2 Comparison of the Low-order and High-order Numerical Results

In this subsection, we show that the high-order method can obtain almost the same results at coarser meshes and with less computational time.

We compare the contours of densities for Group *a* and Group *b*, at the mesh size 800×400 for “b11” and the mesh size 200×100 for “b33” in Figure 5. We find that “b33” at the mesh size 200×100 can bring almost the same resolution (in cuts and contours) as “b11” at the mesh size 800×400 . If we refine the meshes for both “b11” and “b33”, the results in Figure 6 are almost the same as these in Figure 5, therefore we may be convinced that the results at the mesh size 800×400 for “b11” and the mesh size 200×100 for “b33” are the converged physical solution. Of course, the CPU cost per grid point is different for the first order and third order schemes. It is more meaningful to compare the total CPU time. In our implementation on a PC, the total CPU time for “b33” at the mesh size of 200×100 is 128,099 seconds, which is less than 130,358 seconds for “b11” at the mesh size of 800×400 . The saving of CPU time for this case is not significant, however we do save computer memory in using a much coarser mesh for the higher order scheme. The saving of CPU time is more significant for solutions with more complicated structure, or if we need higher precision of the numerical solution, see the next example, and also [30] for more examples.

4.2 Example 2

In this subsection, we design a numerical example, with a simplified setup for the model (12), which has known exact solutions, to quantitatively demonstrate the advantage of using the higher order scheme to reach higher precision with less CPU time than the lower order scheme.

We consider a single group of the model (12), which can be reformulated as:

$$\begin{cases} \rho_t(x, y, t) + \nabla \cdot \mathbf{F}(x, y, t) &= s(x, y, t), \\ \mathbf{F}(x, y, t) &= -\rho(x, y, t)v(\rho)^2 \nabla \Phi(x, y, t), \\ \|\nabla \Phi(x, y, t)\| &= \frac{1}{v(\rho)}, \end{cases} \quad (38)$$

where $s(x, y, t)$ is a source term, chosen so that an explicit exact solution is available, and the velocity $v(\rho) = v_f \exp(-\alpha \rho^2)$.

We solve the system (38) on a domain of $\Omega = [-2, 0] \times [-1, 1]$, subject to the following initial boundary conditions:

$$\mathbf{F}(x, y, t) \cdot \mathbf{n}(x, y) = q(x, y, t), \quad \forall (x, y) \in \Gamma_o, \quad (39)$$

$$\rho(x, y, 0) = \rho_0(x, y), \quad \forall (x, y) \in \Omega, \quad (40)$$

$$\Phi(x, y, t) = 0, \quad \forall (x, y) \in \Gamma_d, \quad (41)$$

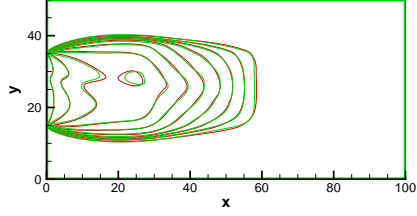
with the inlet segment $\Gamma_o = \{-2\} \times [-1, 1]$, and the outlet segment $\Gamma_d = \{0\} \times [-1, 1]$.

We denote

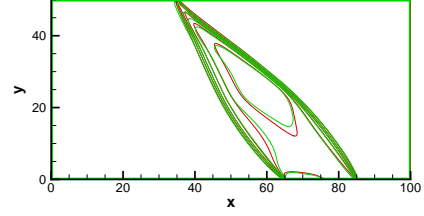
$$\begin{aligned} s_1 &= 3e^{-r \sin t} \\ s_2 &= 12 - 3y + y^3 \\ s_3 &= 9x^2(1 - y^2)^2 \\ v_{log} &= \log \left[\frac{s_1}{c_f v_f \sqrt{s_3 + s_2^2}} \right] \end{aligned}$$

where r, c_f, v_f are given parameters, to write the exact solution as

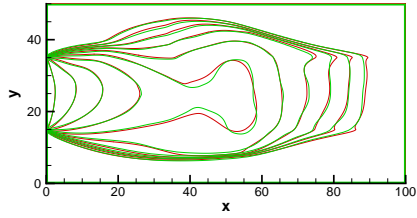
$$\begin{aligned} \Phi_e(x, y, t) &= c_f e^{r \sin t} x \left(-4 + y - \frac{y^3}{3} \right) \\ \rho_e(x, y, t) &= \sqrt{-\frac{v_{log}}{\alpha}} \\ f_{1e}(x, y, t) &= \frac{s_1 s_2 \rho_e}{c_f (s_3 + s_2^2)} \\ f_{2e}(x, y, t) &= \frac{3s_1 x (-1 + y^2) \rho_e}{c_f (s_3 + s_2^2)} \end{aligned}$$



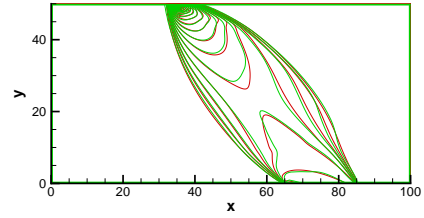
(a) $t = 60s$



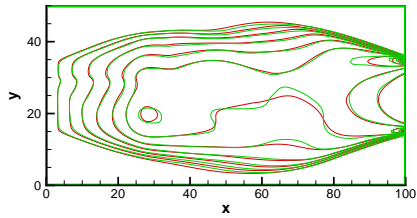
(b) $t = 60s$



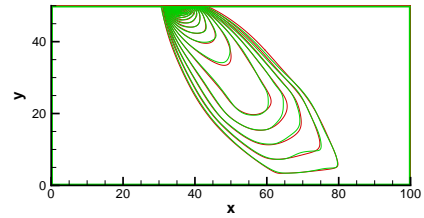
(c) $t = 90s$



(d) $t = 90s$

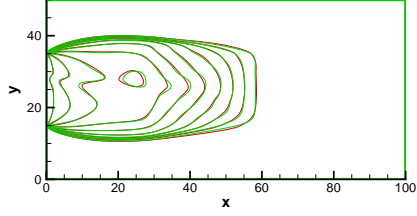


(e) $t = 120s$

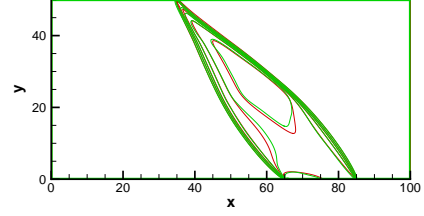


(f) $t = 120s$

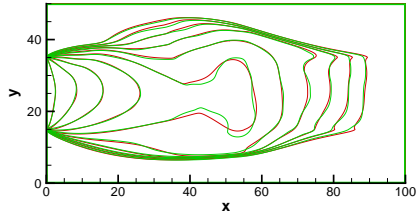
Figure 5: Contours of density ρ at $t = 60s$, $90s$ and $120s$. Left: Group a ; Right: Group b . Mesh size: 200×100 for “b33” and 800×400 for “b11”. Green: “b33”; Red: “b11”.



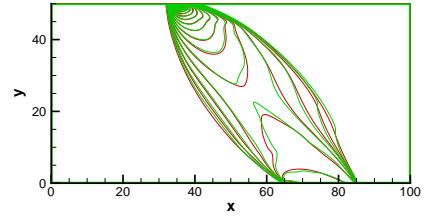
(a) $t = 60s$



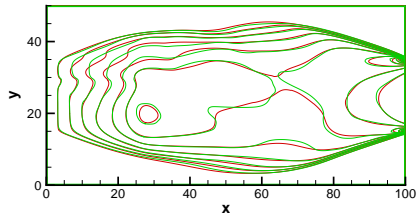
(b) $t = 60s$



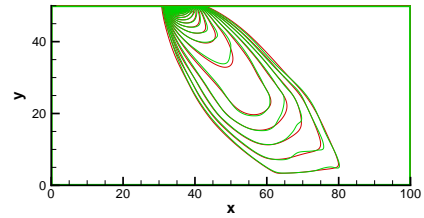
(c) $t = 90s$



(d) $t = 90s$



(e) $t = 120s$



(f) $t = 120s$

Figure 6: Contours of density ρ at $t = 60s$, $90s$ and $120s$. Left: Group a ; Right: Group b . Mesh size: 300×150 for “b33” and 1000×500 for “b11”. Green: “b33”; Red: “b11”.

with the source term given by

$$s(x, y, t) = \frac{r \cos t}{2\alpha\rho_e} + r_1 + r_2$$

where

$$\begin{aligned} r_1 &= \frac{9s_1x(1-y^2)^2s_2(1+4v_{log})}{2c_f\alpha(s_3+s_2^2)^2\rho_e} \\ r_{21} &= 3(1-y^2)^2(s_2+6x^2y) \\ r_{22} &= 4(36+9(-17+x^2)y-6(-2+3x^2)y^3+12y^4+9(-1+x^2)y^5+2y^7)v_{log} \\ r_2 &= \frac{3s_1x(r_{21}+r_{22})}{2c_f\alpha(s_3+s_2^2)^2\rho_e} \end{aligned}$$

and the initial and boundary conditions are

$$q(-2, y, t) = f_{1e}(-2, y, t), \quad \rho_0(x, y) = \rho_e(x, y, 0).$$

We solve the system with the two numerical methods of “b33” (third order) and “b11” (first order), and we take the parameters to be $r = 0.01$, $c_f = 80$, $v_f = 1.034$. For “b33”, we use high order extrapolation to obtain the values on ghost points at the numerical boundary. We set exact values on the physical boundary, such as Γ_o and Γ_d , and then use high order extrapolation to obtain the values of other ghost points. We should mention that in this and the previous examples, the physical boundary Γ_o and Γ_d are located midway between two grid points, and the given values at the boundary is used when doing the extrapolation. We compare the error and order of accuracy for “b33” and “b11” in Table 1. We can find that the expected third order and first order accuracy with the L^1 norm have been obtained, and the third order method has much smaller error than the first order method on the same mesh. A more meaningful comparison is the CPU cost used to reach the same error tolerance. We compare the computational efficiency for the third order method and the first order method in Fig. 7. Clearly, with low error tolerance, the high order method is vastly more efficient than the low order method.

Table 1: Error and order of accuracy for density ρ and potential Φ .

b33	L^1 error of ρ	order	L^1 error of Φ	order
10	4.36E-04	—	1.78E-01	—
20	8.64E-05	2.34	3.02E-02	2.56
40	1.13E-05	2.93	4.16E-03	2.86
80	1.27E-06	3.15	4.93E-04	3.08
160	1.29E-07	3.30	4.88E-05	3.34
b11	L^1 error of ρ	order	L^1 error of Φ	order
20	1.20E-03	—	7.10E-01	—
40	6.27E-04	0.94	3.76E-01	0.92
80	3.36E-04	0.90	1.94E-01	0.95
160	1.77E-04	0.92	9.93E-02	0.97

5 Conclusions

In this work, a third order WENO scheme for the conservation law coupled with a third order fast sweeping method for the Eikonal equation on rectangular meshes, similar to [10], has been designed for a reactive dynamic user equilibrium model for bi-directional pedestrian flows. This model simulates two groups of pedestrians walking in crossing directions in a continuous walking facility. Because of the complication of the model arising from the bi-directional flow, special attention must be paid to the boundary treatment and fast sweeping procedure for the high order scheme to ensure its convergence and non-oscillatory property. We have compared the third order scheme with a first order Lax-Friedrichs

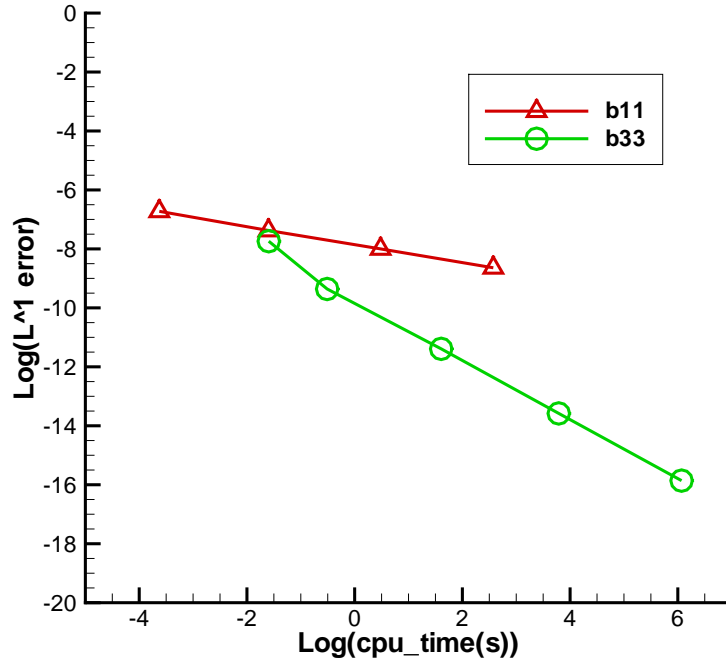


Figure 7: L^1 error versus CPU time for ρ . Each symbol corresponds to one mesh size.

difference scheme for the conservation law coupled with a fast sweeping method for the Eikonal equation on rectangular meshes, and have found that the high order scheme can obtain the same resolution (measured by cuts and contours) on much coarser meshes and with less CPU time, compared with the first order schemes. Another numerical example with known exact solutions quantitatively demonstrates that the higher order scheme can reach higher precision with much less CPU time than the lower order scheme.

6 Acknowledgments

Research of the first and second authors is partially supported by CAS grant KJCX1-YW-21. Research of the third author is partially supported by ARO grant W911NF-08-1-0520 and NSF grant DMS-0809086. Research of the fourth author is partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No.: HKU7184/10E).

References

- [1] Asano M., Sumalee A., Kuwanhara M. and Tanaka S. (2007) Dynamic cell-transmission-based pedestrian model with multidirectional flows and strategic route choices. *Transportation Research Record*, 2039, 42-49.
- [2] Crandall M. and Lions P.L. (1983) Viscosity solutions of Hamilton-Jacobi equations. *Transactions of American Mathematical Society*, 277, 1-42.
- [3] Ho H.W. and Wong S.C. (2006) Two-dimensional continuum modeling approach to transportation problems. *Journal of Transportation Systems Engineering and Information Technology*, 6 (6), 53-72.

- [4] Ho H.W. and Wong S.C. (2007) Housing allocation problem in a continuum transportation system. *Transportmetrica*, 3 (1), 21-39.
- [5] Ho H.W., Wong S.C. and Loo B.P.Y. (2006) Combined distribution and assignment model for a continuum traffic equilibrium problem with multiple user classes. *Transportation Research Part B*, 40, 633-650.
- [6] Ho H.W., Wong S.C., Yang H. and Loo B.P.Y. (2005) Cordon-based congestion pricing in a continuum traffic equilibrium system. *Transportation Research Part A*, 39, 813-834.
- [7] Hoogendoorn S.P. and Bovy P.H.L. (2004a) Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B*, 38, 169-190.
- [8] Hoogendoorn S.P. and Bovy P.H.L. (2004b) Dynamic user-optimal assignment in continuous time and space. *Transportation Research Part B*, 38, 571-592.
- [9] Hoogendoorn S.P., Bovy P.H.L. and Daamen W. (2003) Walking infrastructure design assessment by continuous space dynamic assignment modeling. *Journal of Advanced Transportation*, 38, 69-92.
- [10] Huang L., Wong S.C., Zhang M., Shu C.-W. and Lam W.H.K. (2009) Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm, *Transportation Research Part B*, 43, 127-141.
- [11] Huang L., Xia Y., Wong S.C., Shu C.-W., Zhang M. and Lam W.H.K. (2009) A dynamic continuum model for bi-directional pedestrian flows, *Proceedings of the Institution of Civil Engineers: Engineering and Computational Mechanics*, 162(2), 67-75.
- [12] Hughes R.L. (2002) A continuum theory for the flow of pedestrians. *Transportation Research Part B*, 36, 507-535.
- [13] Jiang G. and Peng D.P. (2000) Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 21, 2126-2143.
- [14] Jiang G. and Shu C.-W. (1996) Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126, 202-228.
- [15] Jiang Y.Q., Xiong T., Wong S.C., Shu C.-W., Zhang M., Zhang P. and Lam, W.H.K. (2009) A reactive dynamic continuum user equilibrium model for bi-directional pedestrian flows. *Acta Mathematica Scientia*, 29(6), 1541-1555.
- [16] Liu X.D., Osher S. and Chan T. (1994) Weighted essentially nonoscillatory schemes. *Journal of Computational Physics*, 115(1), 200-212.
- [17] Liu X.D. and Lax P.D. (2003) Positive schemes for solving multi-dimensional hyperbolic systems of conservation laws II. *Journal of Computational Physics*, 187(2), 428-440.
- [18] Shu C.-W. (1998) Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), *Lecture Notes in Mathematics*, volume 1697, Springer, Berlin, 325-432.
- [19] Shu C.-W. (2007) 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, 47-91.
- [20] Shu C.-W. and Osher S. (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77, 439-471.
- [21] Tong C.O. and Wong S.C. (2000) A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B*, 34, 625-644.

- [22] Wendroff B. (1999) A two-dimensional HLLE Riemann solver and associated Godunov-type difference scheme for gas dynamics. *Computers and Mathematics with Applications*, 38, 175-185.
- [23] Wong S.C. (1994) An alternative formulation of D'Este's trip assignment model. *Transportation Research Part B*, 28, 187-196.
- [24] Wong S.C. (1998) Multi-commodity traffic assignment by continuum approximation of network flow with variable demand. *Transportation Research Part B*, 32, 567-581.
- [25] Wong S.C., Lee C.K. and Tong C.O. (1998) Finite element solution for the continuum traffic equilibrium problems. *International Journal for Numerical Methods in Engineering*, 43 (7), 1253-1273.
- [26] Wong S.C., Leung W.L., Chan S.H., Lam W.H.K., Yung N.H.C., Liu C.Y. and Zhang P. (2010) Bi-directional pedestrian stream model with oblique intersecting angle. *ASCE Journal of Transportation Engineering*, 136 (3), 234-242.
- [27] Wong S.C. and Yang H. (1999) Determining market areas captured by competitive facilities: a continuous equilibrium modeling approach. *Journal of Regional Science*, 39, 51-72.
- [28] Wong S.C., Zhou C.W., Lo H.K. and Yang H. (2004) An improved solution algorithm for the multi-commodity continuous distribution and assignment model. *ASCE Journal of Urban Planning and Development*, 130, 14-23.
- [29] Xia Y., Wong S.C., Zhang M., Shu C.-W. and Lam W.H.K. (2008) An efficient discontinuous Galerkin method on triangular meshes for a pedestrian flow model, *International Journal for Numerical Methods in Engineering*, 76, 337-350.
- [30] Xiong T., Zhang M., Zhang Y.-T. and Shu C.-W. (2010) Fifth order fast sweeping WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment, *Journal of Scientific Computing*, published online: 16 January 2010.
- [31] Yang H and Wong S.C. (2000) A continuous equilibrium model for estimating market areas of competitive facilities with elastic demand and market externalities. *Transportation Science*, 34, 216-227.
- [32] Zhao H.K. (2005) A fast sweeping method for Eikonal equations. *Mathematics of Computation*, 74, 603-627.
- [33] Zhang Y.T., Zhao H.K. and Qian J. (2006) High order fast sweeping methods for static Hamilton-Jacobi equations. *Journal of Scientific Computing*, 29 (1), 25-56.