Article

On the Verification of the Pedestrian Evacuation Model [†]

Petr Kubera ^{1,*} and Jiří Felcman ^{2,†}

¹ Faculty of Science, Jan Evangelista Purkyně University, České Mládeže 8, 400 96 Ústí nad Labem, Czech Republic

² Faculty of Mathematics and Physics, Charles University Prague, Sokolovská 83, 186 75 Praha, Czech Republic; felcman@karlin.mff.cuni.cz

* Correspondence: petr.kubera@ujep.cz

[†] The research was supported by the grant No. UJEP-IGA-TC-2019-53-02-2 of IGA UJEP. The authors acknowledge this support.

[‡] These authors contributed equally to this work.

Abstract: In this article we deal with numerical solution of macroscopic models of pedestrian movement. From a macroscopic point of view, pedestrian movement can be described by a system of first order hyperbolic equations similar to 2D compressible inviscid flow. For the Pedestrian Flow Equations (PFEs) the density ρ and the velocity v are considered as the unknown variables. In PFEs, the social force is also taken into account, which replaces the outer volume force term used in the fluid flow formulation, e.g., the pedestrian movement is influenced by the proximity of other pedestrians. To be concrete, the desired direction μ of the pedestrian movement is density dependent and is incorporated in the source term. The system of fluid dynamics equations is thus coupled with the equation for μ . The main message of this paper is the verification of this model. Firstly, we propose two approaches for the source term discretization. Secondly, we propose two splitting schemes for the numerical solution of the coupled system. This leads us to four different numerical methods for the PFEs. The novelty of this work is the comparative study of the numerical solutions, which shows, that all proposed methods are in the good agreement.

Citation: Kubera, P.; Felcman, J. On the Verification of the Pedestrian Evacuation Model. *Mathematics* **2021**, *9*, 1525. <https://doi.org/10.3390/math9131525>

Academic Editor: Giancarlo Consolo

Received: 1 June 2021

Accepted: 25 June 2021

Published: 29 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: pedestrian flow equations; Eikonal equation; finite volume method; discontinuous Galerkin method; hyperbolic equations; pedestrian evacuation problem

1. Introduction

Modeling of the movement of pedestrians is an interesting non-linear complex problem, where physical and environmental conditions (e.g., visibility see [1]), as well as social and psychological conditions (e.g., fatigue, see [2]), manifest themselves. The complexity of the dynamics of pedestrian movement is illustrated by the emergence of collective effects and self-organisation phenomena. One of the most important practical applications of these models is the ability to simulate the movement of a crowd in panic situations, such as floods, fires, etc. In such cases, it is necessary to have evacuation plans in place and try to predict people's behavior.

In general, there are two classes of the pedestrian behaviour models, macroscopic and microscopic. We are mainly interested in the macroscopic approach where pedestrians are considered as a continuum and their behaviour is described by two-dimensional fluid dynamics equations for an inviscid adiabatic flow subject to outer volume forces. This approach was proposed in [3].

The outer volume forces represent a relaxation term towards a desired velocity. Since the model is devised for room or open space evacuation situations, the desired pedestrian velocity depends on the density distribution. The pedestrians want to avoid the regions with high densities, so the magnitude of the desired velocity is a decreasing function of the density. Such a function stems from the experimental investigations of pedestrian evacuation problems. For various definitions of density dependent magnitude of desired

velocity see [4–6]. A detailed overview can be found in [7]. Also the desired *direction* of pedestrian velocity is density dependent. The desired direction of a pedestrian motion is given as a unit tangential vector to the fastest path to the exit. This works like a car navigation system. The pedestrian knows the magnitude of the velocity in which he would like to go and seek the fastest path to exit. Instead of seeking the fastest path, the so called Eikonal equation can be used for the determination of the desired direction of pedestrian velocity. See [8] for the detailed explanation. The above described model falls into the class of *macroscopic* models, where the state of the system is described by locally averaged quantities. It was originally proposed in [9] as an extension of the vehicular traffic models described in [10,11]. More models originated from different pedestrian behaviour and strategies are proposed and analyzed in [7]. The reader will find a classification of macroscopic models, and some qualitative properties such as hyperbolicity are further studied. The model of pedestrian with distinct desired direction is studied in [12], here the two-phase pedestrian flow model is taken into account.

In the microscopic approach, the pedestrian is considered as an individual interacting with others pedestrians. Example of this approach is the model, where the pedestrians are described by the system of N ordinary differential equations (ODE), here N stands for the number of pedestrians. This model is studied in [13]. Here are stated simulations of the high-density crowd behaviour in the real world objects and situations such as sporting events and pilgrimage places. The transition from microscopic (ODE) to macroscopic model is listed in [14]. Reader finds deep compilation of the literature devoted to booth approaches in [3]. Here is stated more detailed description and differentiation of microscopic models such as a social force model, cellular automata models (CA) or gradient flow methods (macroscopic approach). The comparison of the cellular automata model with the PFEs (macroscopic) can be found in [15], where the new Cellular Automaton (CA) model is proposed for the simulation of the pedestrian flow and the comparative study is presented. The level between microscopic and macroscopic scale is represented by the mesoscopic scale. An example of this kind of models is kinetic model described in [16].

It is seen from a review of the literature that extensive research is currently underway on modelling pedestrian evacuation and related situations. In this paper, we are interested in a model that is based on fluid dynamics equations and formulated as a hyperbolic system with a source term on the right hand side of the equation. In accordance with [17], such a system is usually solved by the splitting method. Because there is a lack of experimental data from real-life situations in this case, it is difficult to validate that the numerical results correspond to real-world situations. But what is possible is to study whether the numerical solution is correct.

To this end, we have proposed two substantially different numerical methods of splitting: Splitting Scheme I and Splitting scheme II. The paper emphasizes which parts of discretization using the Finite Volume Method (FVM) and the combined Finite Volume and the Discontinuous Galerkin method (FV-DG), are original. The choice of the numerical flux in both methods is original (the original numerical flux of the Vijayasundaram type, the method of solving the Eikonal equation using Dijkstra's algorithm for the shortest path in the graph is original). The proposed numerical methods are mutually compared. A representative sample of adequate problems from pedestrian evacuation is solved. The novelty of the paper is a comparison of the results of different numerical methods for solving the given problem. The article thus deviates from the usual scheme One problem-One method-One result. We have One problem-Four methods-Four results relatively in a good agreement. The choice of method is then up to the user: simplicity versus higher order versus computational time.

The paper is organized as follows. In Section 2 the macroscopic model of pedestrian flow describing the room or open space pedestrian evacuation is formulated. The splitting technique and its numerical realization is explained in Section 3. The physically relevant computations of various evacuation scenarios are presented in Section 4. A conclusion

summarizing the results achieved and emphasizing the novelty of the paper is included in Section 5.

2. Governing Equations

Let us consider the pedestrian flow in the polygonal domain $\Omega \subset \mathbb{R}^2$ and time interval $(0, T)$ with $T > 0$. We suppose that the boundary $\partial\Omega = \bar{\Gamma}_w \cup \bar{\Gamma}_o$, where the subscript w states for wall and o for the exit (outflow). The bar in $\bar{\Gamma}$ denotes the closure of the set Γ . We further suppose that the subsets $\Gamma_w, \Gamma_o \in \partial\Omega$ are disjoint and formed by a finite number of connected sets, open with respect to the topology on $\partial\Omega$, where the one-dimensional measure $\text{meas}(\Gamma_o) > 0$, i.e., Γ_o is not a point. Such a setting is convenient for the modeling of pedestrian evacuation problems in order to predict the pedestrian behavior in panic situations. The system of the PDE's for the vector valued function w is

$$\frac{\partial w}{\partial t} + \sum_{s=1}^2 \frac{\partial f_s(w)}{\partial x_s} = s(w, \mu) \quad \text{in } Q_T = \Omega \times (0, T). \quad (1)$$

Here the unknown variable $w = w(x, t)$ forms a vector $w = (w_1, w_2, w_3)^T \equiv (\rho, \rho v)^T \equiv (\rho, \rho v_1, \rho v_2)^T$. We denote by ρ the density of pedestrians in [ped/m²] and by $v = (v_1, v_2)$ their velocity in [m/s].

The fluxes f_s are defined by

$$f_s(w) = (\rho v_s, \rho v_1 v_s + \delta_{s1} p_0 \rho^\gamma, \rho v_2 v_s + \delta_{s2} p_0 \rho^\gamma)^T, \quad p_0 > 0, \gamma > 1.$$

The power law for isentropic gases expressing the dependence of the pressure p on the density ρ is used in the definition of the fluxes f_s . In our computations the constant p_0 is set $p_0 = 1$ and the Poisson constant $\gamma = 2$. The source term is given by

$$s(w, \mu) = \left(0, \frac{1}{\tau} \rho (V\mu - v) \right)^T. \quad (2)$$

Here V represents the magnitude of the desired velocity in which the pedestrian would like to go, we suppose $V = \tilde{V}(\rho) > 0$. The notation means that $V(x, t) = \tilde{V}(\rho(x, t))$ and $(x, t) \in \Omega \times (0, T)$. The definition of $\tilde{V}(\rho)$ can be found in [3]. In our work we use the exponential dependence

$$\tilde{V}(\rho) = v_{\max} e^{-\alpha \left(\frac{\rho}{\rho_{\max}} \right)^2}, \quad (3)$$

where $\alpha = 7.5$, the maximum density $\rho_{\max}$ at which is hardly possible to move is set $\rho_{\max} = 9$ and the speed of free flow $v_{\max} \in [1, 7]$. These parameters setting is based on [4], where experimental investigation of pedestrian behavior is listed.

The unit vector μ is the preferred direction of the movement of pedestrians, $\mu = \mu(x, t)$, $(x, t) \in \Omega \times (0, T)$, and it depends on the distribution of the density ρ in the whole domain Ω , which we express by the functional dependence

$$\mu(x, t) = \Psi(\rho(\cdot, t); x, t). \quad (4)$$

The function V and the unit vector μ characterize how the speed of pedestrians changes with the density. The parameter τ represents a relaxation time describing how fast pedestrians correct their current velocity v to the desired one $V\mu$. We set this parameter $\tau = 0.61$ according to the validation of the model with experimental pedestrian movement.

2.1. Functional Dependence

To determine the direction μ (desired velocity direction) we have basically two equivalent possibilities how to define functional Ψ in (4).

2.1.1. Integral Formulation

$$\Psi_D(\rho(\cdot, t); x, t) = -\frac{\nabla\phi(x)}{|\nabla\phi(x)|}, \quad (5)$$

$$\phi(x) = \min_{\eta(\sigma; x)} \int_{\eta} \frac{dS}{V(\cdot, t)}. \quad (6)$$

The numerical approximation of ϕ in (6) is obtained via Dijkstra's algorithm, see original paper [18] or more detailed explanation in [19], and thus we denote it by the subscript D . In (6) the minimum is taken over all piecewise regular curves $\eta(\sigma, x)$ in Ω , parametrized by the parameter $\sigma \in [0, 1]$ with the property $\eta(0; x) = x, \eta(1; x) \in \Gamma_o$. Let us note that the whole distribution of the density at each time instant t has to be given to determine μ .

2.1.2. Differential Formulation

$$\Psi_{BR}(\rho(\cdot, t); x, t) = -\frac{\nabla\phi(x)}{|\nabla\phi(x)|}, \quad (7)$$

$$|\nabla\phi(x)| = \frac{1}{V(x, t)}, \quad x \in \Omega, \quad (8)$$

$$\phi(x) = 0, \quad x \in \Gamma_o. \quad (9)$$

Equation (8) is the so called Eikonal equation of pedestrian flow problem. This equation is completed by the boundary condition (9). The numerical approximation of ϕ in (8) and (9) is obtained via the Bornemann-Rasch algorithm, see [20], and thus we denote it by the subscript BR .

An alternative approach how to define the functional Ψ was introduced in [21]. The physical meaning of μ as the tangential vector to fastest path to the exit and ϕ as the shortest time to reach the exit for a given magnitude of velocity V is explained in [8,21].

Let w^0 be a vector function, the initial condition for the system (1)–(4) reads

$$w(x, 0) = w^0(x), \quad x \in \Omega. \quad (10)$$

The boundary condition is prescribed

$$B(w(x, t)) = 0 \quad \text{for } (x, t) \in \partial\Omega \times (0, T), \quad (11)$$

where B is appropriate boundary operator. For the boundary condition on the wall we use a normal component of velocity equal to zero

$$v \cdot n = 0.$$

On the exits we prescribe $w_1 = \rho \approx 0$, the space behind the exit is free and empty and pedestrians can leave this space. The more detailed specification of the boundary conditions is described in [8]. Let us note that due to the hyperbolicity of the system the Dirichlet boundary condition cannot be prescribed directly. We solve a linearized initial-boundary value problem instead.

3. Numerical Solution

The usual way how to solve the hyperbolic systems with the source term is the splitting technique.

3.1. Splitting Schemes

We consider two splitting schemes for system (1). Since there are various possibilities of the splitting, a question arises, if the equations are solved “right”. This is related to the notion of “verification of the model”. We answer this question from the numerical point of view. In what follows, we discretize parts of the continuum model by substantially different numerical techniques. Finally we show that the numerical results obtained by the different numerical methods are in a good agreement.

3.1.1. Splitting Scheme I

The first splitting scheme for system (1) reads:

$$\frac{\partial w}{\partial t} + \sum_{s=1}^2 \frac{\partial f_s(w)}{\partial x_s} = 0, \quad (12)$$

$$\mu(x, t) = \Psi(w_1(\cdot, t); x, t), \quad (13)$$

$$\frac{\partial w}{\partial t} = s(w, \mu). \quad (14)$$

3.1.2. Splitting Scheme II

The second splitting scheme for system (1) reads:

$$\mu(x, t) = \Psi(w_1(\cdot, t); x, t), \quad (15)$$

$$\frac{\partial w}{\partial t} + \sum_{s=1}^2 \frac{\partial f_s(w)}{\partial x_s} = s(w, \mu). \quad (16)$$

Remember that the first component of w has the physical meaning of density, $w_1 = \rho$.

Above stated splitting Schemes (12)–(14) (Splitting Scheme I) and (15) and (16) (Splitting Scheme II) are employed to numerically solve system (1). Booth schemes are applied in an iterative manner. We employ Finite Volume Method (FV) for the discretization (12)–(14), see Section 3.2, while Finite Volume Method in space and the Discontinuous Galerkin Method in time are used for (15) and (16) (FV-DG), see Section 3.5.

In booth schemes we use triangular mesh $\mathcal{T}_h = \{K_i\}_{i \in I}$, where $I \subset \mathbb{Z}^+ = \{0, 1, \dots\}$ is an index set and $h > 0$. Closed triangles $K_i, i \in I$ have usual properties from the finite element method. By the discretization a partition $0 = t_0 < t_1 < \dots < t_r = T$ of the time interval $[0, T]$ is meant. We denote by $\tau_m = t_m - t_{m-1}$ the size of time step between t_{m-1} and t_m and $I_m = (t_{m-1}, t_m)$, $m = 1, \dots, r$. We denote the set of all vertices of the triangulation $\mathcal{T}_h$ by $\mathcal{V}_h$ and the set of all edges of $\mathcal{T}_h$ by $\mathcal{F}_h$.

3.2. FV Discretization of the Splitting Scheme I

Since we employ the Finite Volume Method in the Splitting scheme I, the approximate solution is considered as piecewise constant in Ω at the time instant t_m

$$w(\cdot, t_m)|_{K_i} \approx w_i^m. \quad (17)$$

Further we denote

$$\mathbf{w}^m = \{w_i^m\}_{i \in I}$$

and derive the FV scheme for $\mathbf{w}^m$.

3.2.1. FV Discretization of the Hyperbolic System (12)

Let us assume that w is the exact solution of (12). We integrate (12) over K_i , exchange the integration and partial differentiation in the first term and use the Green’s theorem with respect to the space coordinates. Then

$$\frac{d}{dt} \int_{K_i} w(x, t) dx + \int_{\partial K_i} \sum_{s=1}^2 n_s f_s(w(x, t)) dS = 0. \quad (18)$$

Writing $\partial K_i = \bigcup_{j \in S(i)} \Gamma_{ij}$, $\Gamma_{ij} = \partial K_i \cap \partial K_j$, where $S(i)$ is an index set of neighboring triangles K_j to K_i (special treatment has to be done if K_i has a side on $\partial\Omega$) and approximating

$$w(x, t)|_{K_i} \approx w_i(t), \quad (19)$$

$$\int_{K_i} w(x, t) dx \approx |K_i| w_i(t),$$

$$\sum_{s=1}^2 n_s f_s(w(\cdot, t)) \Big|_{\Gamma_{ij}} \approx H(w_i(t), w_j(t), n_{ij}), \quad n_{ij} = n|_{\Gamma_{ij}}, \quad (20)$$

where $|K_i|$ is the area of K_i and H is the so called *numerical flux*, we derive from (18) the semi-discrete approximation of the hyperbolic system (12)

$$\frac{d}{dt} w_i(t) = -\frac{1}{|K_i|} \sum_{j \in S(i)} |\Gamma_{ij}| H(w_i(t), w_j(t), n_{ij}). \quad (21)$$

Here $|\Gamma_{ij}|$ is the length of the side Γ_{ij} . We use the original numerical flux H of Vijayasundaram type proposed in [22]. Its brief description can be found in [8].

For K_i having a side on the boundary $\partial\Omega$, the boundary conditions have to be taken into account. On the walls we use inviscid mirror boundary conditions from ([23], p. 415). Boundary conditions on exit are obtained from solution of a linearized 1D Riemann problem. The formulas for boundary conditions can be found in [8].

If we take Equation (21) for all $i \in I$, we get the vector form of the semi-discrete approximation of (12)

$$\frac{d}{dt} \mathbf{w}(t) = \mathbf{H}(\mathbf{w}(t)), \quad (22)$$

where

$$\mathbf{w}(t) = \{w_i(t)\}_{i \in I}, \quad (23)$$

$$\mathbf{H}(\mathbf{w}(t)) = \left\{ -\frac{1}{|K_i|} \sum_{j \in S(i)} |\Gamma_{ij}| H(w_i(t), w_j(t), n_{ij}) \right\}_{i \in I}. \quad (24)$$

Equation (22) is equipped with the initial condition

$$\mathbf{w}(0) = \{w_i(0)\}_{i \in I}, \quad w_i(0) = \frac{1}{|K_i|} \int_{K_i} w^0(x) dx, \quad (25)$$

where w^0 is defined in (10). We denote the approximate solution of (22) at time instant t_m by $\mathbf{w}^m$, i.e.,

$$\mathbf{w}(t_m) \approx \mathbf{w}^m = \{w_i^m\}_{i \in I}. \quad (26)$$

The following second order Runge–Kutta method can be used for the discretization of Equation (22)

$$\mathbf{w}^m = \frac{1}{2} \left(\mathbf{w}^{m-1} + \tilde{\mathbf{w}} + \tau_m \mathbf{H}(\tilde{\mathbf{w}}) \right), \quad (27)$$

where

$$\tilde{\mathbf{w}} = \mathbf{w}^{m-1} + \tau_m \mathbf{H}(\mathbf{w}^{m-1}). \quad (28)$$

Another possibility is to use the Euler method for the discretization of Equation (22)

$$\mathbf{w}^m = \mathbf{w}^{m-1} + \tau_m \mathbf{H}(\mathbf{w}^{m-1}). \quad (29)$$

The initial condition $\mathbf{w}^0$ is set according to (25)

$$\mathbf{w}^0 = \mathbf{w}(0) = \left\{ w_i^0 \right\}_{i \in I}, \quad w_i^0 = \frac{1}{|K_i|} \int_{K_i} w^0(x) \, dx. \quad (30)$$

The time step τ_m is given by the CFL stability condition stemming from the finite volume approach. For the Runge–Kutta method, the time step is halved. Relations (27)–(29) represent the discrete analog to the hyperbolic system (12), respectively.

3.2.2. FV Approximation of μ in (13)

The numerical approximation of μ will be sought in the space of piecewise constant functions on K_i at t_m , i.e.,

$$\mu(\cdot, t_m)|_{K_i} \approx \mu_i^m. \quad (31)$$

Two formulations of the functional Ψ are considered in (4), namely Ψ_D and Ψ_{BR} defined in (5)–(9). The subscripts mean that the Dijkstra or Bornemann-Rasch algorithm are used for the computation of Ψ .

3.2.3. FV Approximation of Ψ_D

To compute integral in (6) we approximate $\phi(P)$ at vertex $P \in \mathcal{V}_h$ by ϕ_P . The approximation ϕ_P is obtained by the minimization process over edges of the triangulation $\mathcal{T}_h$ only. To this end we approximate the quantity $V(\cdot, t_m)$ on each edge $E \in \mathcal{F}_h$ by a arithmetic mean of $V_{P_s}^m, V_{P_e}^m$ at vertices P_s, P_e of the edge E , see (32). The approximations V_P^m of point values of $V(P, t_m)$ are obtained from the piecewise constant approximations w_i^m of $w(\cdot, t_m)|_{K_i}$ on triangles K_i as weighted averages of values on the set of triangles $\mathcal{K}_P$ having the point P in common. The definition of V and relation (3) is taking into account. Let us note that the first component of w_i^m is the approximation of the density. Assuming above mentioned leads the integral in (6) can be determined exactly. The meaning of this integral is the time it takes to go from the vertex P to the exit Γ_o . The pedestrian path is formed by edges E of the graph given by the triangulation $\mathcal{T}_h$ and we get

$$\phi(P) \approx \phi_P = \sum_{E \in \mathcal{E}} \frac{|E|}{\frac{1}{2}(V_{P_s}^m + V_{P_e}^m)}. \quad (32)$$

Here $|E|$ represents the euclidean norm (length) of the edge E , $\mathcal{E}$ is the shortest path in the graph starting from the vertex P and ending in some vertex of Γ_o . Since we can define norm $\|\cdot\|$ of an edge E

$$\|E\| = \frac{|E|}{\frac{1}{2}(V_{P_s}^m + V_{P_e}^m)}, \quad (33)$$

the term *shortest* path means with respect to this norm. According to the physical meaning of quantities in the definition (33) the shortest means the fastest.

The problem of finding shortest path is solved by modified Dijkstra's algorithm where we use norm (33) instead of the length of the edge E . We utilize the fact the graph is not oriented and choose the starting vertex as the vertex of the boundary Γ_o and set the ending vertex $P \in \mathcal{V}_h$. In [24] the multiple starting point Dijkstra's algorithm has been proposed. This algorithm computes the approximations ϕ_P for all vertices $P \in \mathcal{V}_h$ in one run taking an arbitrary vertex lying on Γ_o as the starting point. The more detailed description can be found in [8].

From point values $\phi_P, P \in \mathcal{V}_h$, the piecewise linear approximation ϕ_h of ϕ in the space

$$X_h = \{\phi_h \in C(\overline{\Omega}); \phi_h|_{K_i} \text{ are piecewise linear on the triangulation } \mathcal{T}_h\}$$

is constructed. The approximation $\Psi_{D,i}(\mathbf{w}^m)$ of Ψ_D in (5) on K_i at the time t_m is defined

$$\Psi_{D,i}(\mathbf{w}^m) = -\frac{\nabla(\phi_h|_{K_i})}{|\nabla(\phi_h|_{K_i})|}$$

and used for the approximation of μ in (31)

$$\mu_i^m = \Psi_{D,i}(\mathbf{w}^m). \quad (34)$$

For the simplicity of notation the argument in $\Psi_{D,i}$ is written as $\mathbf{w}^m$ although μ_i^m depends only on the piecewise constant approximation of the density. Relation (34) expresses the fact that μ_i^m depends on the approximation of the density on all $K_j, j \in I$.

3.2.4. FV Approximation of Ψ_{BR}

We implemented the Bornemann and Rasch algorithm [20] for the solution of the Eikonal Equations (8) and (9). The function ϕ (called potential) is approximated by the piecewise linear function $\phi_h \in X_h$. The key point of this algorithm is the use of the Hopf-Lax formula in order to obtain the piecewise linear finite element solution of the certain local Dirichlet problem. To this end we construct the approximations V_P^m of point values of $V(P, t_m)$ from the piecewise constant approximations w_i^m of $w(\cdot, t_m)|_{K_i}$ on triangles K_i as weighted averages of values on the set of triangles $\mathcal{K}_P$ having the point P in common.

The sought potential ϕ_h is the fix point of the operator Λ_h , where $\Lambda_h : X_h \rightarrow X_h$ is the so called *Hopf-Lax update* function:

$$(\Lambda_h \phi_h)(P) = \begin{cases} \min_{x \in \partial \mathcal{K}_P} (\phi_h(x) + |P - x| \frac{1}{V_P^m}), & \text{where } P \in \Omega, \\ \phi_h(P) & P \in \partial \Omega. \end{cases} \quad (35)$$

Here $\partial \mathcal{K}_P$ is the boundary of the set of volumes having the vertex P in common. The details about the minimization process in Hopf-Lax update Formula (35) and the convergence of the fix point iterations

$$\phi_h^{n+1} = \Lambda_h \phi_h^n, \quad n = 0, 1, 2, \dots, \text{ with } \phi_h^0(P) = \begin{cases} \infty & P \in \Omega, \\ 0 & P \in \Gamma_0, \end{cases} \quad (36)$$

can be found in [20]. Having computed the potential $\phi_h \in X_h$ at each vertex of the mesh, we assess the gradient $\nabla(\phi_h|_{K_i})$ at the centre of gravity of the triangle K_i and the quantity

$$\Psi_{BR,i}(\mathbf{w}^m) = -\frac{\nabla(\phi_h|_{K_i})}{|\nabla(\phi_h|_{K_i})|}$$

is used for the approximation of μ in (31)

$$\mu_i^m = \Psi_{BR,i}(\mathbf{w}^m). \quad (37)$$

Relations (34) and (37) represent the discrete analog of the determination of the desired direction μ in (13).

3.3. Discretization of System (14)

Integrating (14) over K_i , we proceed analogously as in the case of the discretization of the hyperbolic system (12). We define the semi-discrete approximation of (14) as

$$\frac{d}{dt} \mathbf{w}(t) = \mathbf{S}(\mathbf{w}(t)). \quad (38)$$

The notation (19), (23) and (26) and the analogous notation for μ and the source term s

$$\mu(x, t)|_{K_i} \approx \mu_i(t), \quad (39)$$

$$\mu(t) = \{\mu_i(t)\}_{i \in I}, \quad (40)$$

$$\mathbf{S}(\mathbf{w}(t)) = \{s(w_i(t), \mu_i(t))\}_{i \in I} \quad (41)$$

$$\mu(t_m) \approx \mu^m = \{\mu_i^m\}_{i \in I} \quad (42)$$

$$\mathbf{S}(\mathbf{w}^m) = \{s(w_i^m, \mu_i^m)\}_{i \in I} \quad (43)$$

is used. The quantity μ_i^m is obtained from $\mathbf{w}^m$ using the relations (34) or (37). Equation (38) can be solved numerically by the same Runge–Kutta method as Equation (22).

$$\mathbf{w}^m = \frac{1}{2} \left(\mathbf{w}^{m-1} + \tilde{\mathbf{w}} + \tau_m \mathbf{S}(\tilde{\mathbf{w}}) \right), \quad (44)$$

where

$$\tilde{\mathbf{w}} = \mathbf{w}^{m-1} + \tau_m \mathbf{S}(\mathbf{w}^{m-1}). \quad (45)$$

Another possibility is to use the Euler method for the discretization of Equation (38)

$$\mathbf{w}^m = \mathbf{w}^{m-1} + \tau_m \mathbf{S}(\mathbf{w}^{m-1}). \quad (46)$$

Relations (44)–(46) represent the discrete analog of system (14), respectively.

The finite volume discretization of the Splitting Scheme I (12)–(14) consists of the passage through discretizations (27) and (28) or (29), (34) or (37) and (44) and (45) or (46) in an iterative manner.

3.4. Splitting Algorithm I

The numerical solution of the Splitting Scheme I is numerically performed by the following Algorithms 1 and 2:

Algorithm 1 Runge-Kutta splitting

Set the initial condition $\mathbf{w}^0$ and the method M for the determination of direction μ ,
 $M \in \{D, BR\}$,
for $m = 1, 2, \dots$ **do**
 (1) $\tilde{\mathbf{w}} := \mathbf{w}^{m-1} + \tau_m \mathbf{H}(\mathbf{w}^{m-1})$;
 $\mathbf{w}^{hyp} := \frac{1}{2} (\mathbf{w}^{m-1} + \tilde{\mathbf{w}} + \tau_m \mathbf{H}(\tilde{\mathbf{w}}))$;
 (2) $\mu_i^{hyp} := \Psi_{M,i}(\mathbf{w}^{hyp}), i \in I$;
 (3) $\tilde{\mathbf{w}} := \mathbf{w}^{hyp} + \tau_m \mathbf{S}(\mathbf{w}^{hyp})$;
 $\tilde{\mu}_i := \Psi_{M,i}(\tilde{\mathbf{w}}), i \in I$;
 $\mathbf{w}^m := \frac{1}{2} (\mathbf{w}^{hyp} + \tilde{\mathbf{w}} + \tau_m \mathbf{S}(\tilde{\mathbf{w}}))$;
end for

In (1) the following notation is used

$$\tilde{\mathbf{w}} = \{\tilde{w}_i\}_i \in I, \quad \mathbf{w}^{hyp} = \{w_i^{hyp}\}_i \in I.$$

In (3) the following notation is used

$$\mathbf{S}(\mathbf{w}^{hyp}) = \{s(w_i^{hyp}, \mu_i^{hyp})\}_{i \in I}, \quad \mathbf{S}(\tilde{\mathbf{w}}) = \{s(\tilde{w}_i, \tilde{\mu}_i)\}_{i \in I}.$$

The quantity μ_i^{hyp} is obtained from $\mathbf{w}^{hyp}$ using relations (34) or (37). Analogously $\tilde{\mu}_i$ is obtained from $\tilde{\mathbf{w}}$.

If the Euler method (29) and (46) is used for the numerical solution of (22) and (38), respectively, then steps (1) to (3) are replaced by

Algorithm 2 Euler splitting

```

for  $m = 1, 2, \dots$  do
  (1)  $\mathbf{w}^{hyp} := \mathbf{w}^{m-1} + \tau_m \mathbf{H}(\mathbf{w}^{m-1});$ 
  (2)  $\mu_i^{hyp} := \Psi_{M,i}(\mathbf{w}^{hyp}), i \in I;$ 
  (3)  $\mathbf{w}^m := \mathbf{w}^{hyp} + \tau_m \mathbf{S}(\mathbf{w}^{hyp});$ 
end for
  
```

Definition 1. We say that the piecewise constant vector valued function $\mathbf{w}_h^m, m = 1, 2, \dots$, defined almost everywhere in the computational domain Ω , is the finite volume solution of (12)–(14), if $\mathbf{w}_h^m|_{\overset{\circ}{K}_i} = \mathbf{w}_i^m$ for all $i \in I$, where $\overset{\circ}{K}_i$ is the interior of K_i , i.e., $\overset{\circ}{K}_i = K_i \setminus \partial K_i$, and $\mathbf{w}_i^m$ are obtained from the Splitting Algorithm I as components of $\mathbf{w}^m$. The function $\mathbf{w}_h^m$ is the approximate solution at time $t = t_m$ on the triangulation $\mathcal{T}_h$. The vector $\mathbf{w}_i^m$ is the value of the approximate solution on the finite volume $K_i \in \mathcal{T}_h$ at time t_m .

3.5. FV-DG Discretization of the Splitting Scheme II

The FV-DG space-time discontinuous Galerkin method was derived in [8]. In the following paragraphs we briefly summarize it. Let q be an integer and $P_l(0, 1)$ the space of polynomials of degree l on $(0, 1)$ for $l = 0, \dots, q$. The approximate solution $\mathbf{w}_h$ is sought in the finite element space

$$\mathbf{S}_{h,\tau}^q = \left\{ \mathbf{w} \in L^2(Q_T)^3; \mathbf{w}|_{K_i \times I_m} = \sum_{l=0}^q \mathbf{w}_{i,m}^l v_l\left(\frac{t - t_{m-1}}{\tau_m}\right), \right. \\ \left. \mathbf{w}_{i,m}^l \in \mathbb{R}^3, K_i \in \mathcal{T}_h, l = 0, \dots, q, i \in I, v_l(t) \in P_l(0, 1) \right\}. \quad (47)$$

The approximate solution $\mathbf{w}_h \in \mathbf{S}_{h,\tau}^q$ can be written in the form

$$\mathbf{w}_h(x, t) = \sum_{m=1}^r \mathbf{w}_h^m(x, t), \text{ where} \\ \mathbf{w}_h^m \in \mathbf{S}_{h,\tau}^q, \mathbf{w}_h^m(x, t) = \begin{cases} \mathbf{w}_h(x, t) & \text{for } t \in I_m, x \in \Omega, \\ 0 & \text{for } t \notin I_m, x \in \Omega, \end{cases}, \quad m = 1, \dots, r. \quad (48)$$

Provided ξ is a function defined in $\bigcup_{m=1}^r I_m$, we use the notation

$$\xi|_m^\pm = \xi(t_m \pm) = \lim_{t \rightarrow t_m \pm} \xi(t), \quad \{\xi\}_m = \xi|_m^+ - \xi|_m^-, \quad (49)$$

if the one-sided limits $\lim_{t \rightarrow t_m \pm} \xi(t)$ exist. We put $\mathbf{w}_h|_0^- = \Pi \mathbf{w}^0$, where $\Pi \mathbf{w}^0$ is the piecewise constant projection of the initial condition $\mathbf{w}^0$ on the mesh $\mathcal{T}_h$.

3.6. FV-DG Approximation of μ in (15)

The desired direction μ in (15) depends on the distribution of $\rho(\cdot, t)$. Two formulations of the functional Ψ are considered in (15), namely Ψ_D and Ψ_{BR} defined in (5)–(9). The meaning of subscripts is the same as above.

We denote the approximation $\mu|_{\Omega \times I_m}$ as μ_h^{m-1} . This approximation is computed from values $\mathbf{w}_h^{m-1}|_{m-1}^-$, precisely from the density component $\rho_h^{m-1}(\cdot, t_{m-1}-)$ of value $\mathbf{w}_h^{m-1}|_{m-1}^-$.

3.6.1. FV-DG Approximation of Ψ_D

Assume in the given time instant t_{m-1} we have the piecewise constant density $\rho_h^{m-1}(\cdot, t_{m-1}-)$, for $m=1$ we have initial condition. We construct a continuous piecewise linear recovery $\tilde{\rho}_h(\cdot, t_{m-1}-)$ from $\rho_h(\cdot, t_{m-1}-)$ and determine the approximations V_P^{m-1} of velocity V given by (3) for all vertices $P \in \mathcal{V}_h$

$$V_P^{m-1} = \tilde{V}(\tilde{\rho}_h(P, t_{m-1}-)). \quad (50)$$

Having point values V_P^{m-1} the situation is similar to the Section 3.2.3 and with the same notation the approximation ϕ_P of $\phi(P)$ is defined

$$\phi(P) \approx \phi_P = \sum_{E \in \mathcal{E}} \frac{|E|}{2(V_{P_s}^{m-1} + V_{P_e}^{m-1})}. \quad (51)$$

To find the shortest path in the graph given by the triangulation $\mathcal{T}_h$ the modified Dijkstra's algorithm is employed. From the values $\phi_P, P \in \mathcal{V}_h$, the piecewise linear function ϕ_h is constructed. This recovery is used to compute gradients of ϕ_h on triangles $K \in \mathcal{T}_h$. The point approximations of Ψ by weighted average of gradients of ϕ_h is determined

$$\mu_P^{m-1} = -\frac{\tilde{\nabla} \phi_h(P)}{|\tilde{\nabla} \phi_h(P)|}, \quad \tilde{\nabla} \phi_h(P) = \frac{\sum_{K \in \mathcal{K}_P} |K| \nabla \phi_h|_K}{\sum_{K \in \mathcal{K}_P} |K|}, \quad (52)$$

$$\Psi(\rho(P, t_m); P, t_{m-1}-) \approx \mu_P^{m-1}, \quad (53)$$

where $\mathcal{K}_P$ is the set of triangles from $\mathcal{T}_h$ having the vertex P in common and weights in (52) are areas $|K|$ of each triangle $K \in \mathcal{T}_h$.

3.6.2. FV-DG Approximation of Ψ_{BR}

The point values V_P^{m-1} are computed from relation (50) and ϕ_h is expressed from (35) and (36) (Bornemann-Rasch algorithm). Relations (52) and (53) are used to approximate point values of μ .

3.6.3. FV-DG Approximation of the Direction μ

Having approximate point values of μ , for each of the above mentioned approaches, the approximation μ_h^{m-1} on $K_i \times I_m$ is constructed. Firstly we define the piecewise linear function $\tilde{\mu}_h^{m-1}$ on $\mathcal{T}_h$. Secondly this function is extended on $K_i \times I_m$ to be constant and the approximation μ_h^{m-1} is set

$$\mu_h^{m-1}(\cdot, t) = \tilde{\mu}_h^{m-1}(\cdot), \quad t \in I_m. \quad (54)$$

The passage from $w_h^{m-1}|_{m-1}^-$ to the definition (54) can be written as

$$\mu_h^{m-1} = \Psi_{D,h}(w_h^{m-1}|_{m-1}^-) \quad \text{or} \quad \mu_h^{m-1} = \Psi_{BR,h}(w_h^{m-1}|_{m-1}^-), \quad (55)$$

where the functionals $\Psi_{D,h}$ and $\Psi_{BR,h}$ represent the process described above, see Sections 3.6.1 and 3.6.3. Here w_h^{m-1} denotes the restriction $w_h|_{\Omega \times I_{m-1}}$. The upper index of μ_h^{m-1} indicates that μ is computed from values w_h^{m-1} , although μ_h^{m-1} is defined on $\Omega \times I_m$.

3.7. FV-DG Discretization of (16)

The discretization of (16) is described in detail in [8]. Here we mention the main ideas. Let w be the exact solution of (1). Multiplying (1) by $\xi \in S_{h,\tau}^q$ and integrating over $K_i \times I_m$, using the Green's theorem with respect to the space coordinates and taking sum over all

elements $K_i \in \mathcal{T}_h$ we obtain (some manipulation needed and using the fact ξ is constant with respect to the space)

$$\int_{I_m} (w', \xi) dt - \int_{I_m} \left(\sum_{K_i \in \mathcal{T}_h} \int_{\partial K_i} \sum_{s=1}^2 n_s f_s(w) \cdot \xi dS \right) dt = \int_{I_m} (s(w, \mu), \xi) dt, \quad (56)$$

where by w' we denoted $\partial w / \partial t$.

Firstly we discretize the time derivative. Detailed procedure for the treatment of the time derivative can be found in [8]. This procedure is based on the integration by parts. The result of this is the following identity

$$\begin{aligned} \int_{I_m} (w', \xi) dt &= - \int_{I_m} (w, \xi') dt + (w_m^-, \xi|_m^-) - (w_{m-1}^-, \xi|_{m-1}^+) \\ &= \dots \int_{I_m} (w', \xi) dt + (\{w\}_{m-1}, \xi|_{m-1}^+), \end{aligned} \quad (57)$$

where we use the notation introduced in (49) and $\xi' = \partial \xi / \partial t$.

Secondly, the integral over ∂K_i in (56) is approximated. We use the numerical flux

$$\sum_{s=1}^2 n_s f_s(w) \Big|_{\partial K_i} \approx H(w|_{K_i}^{\text{int}}, w|_{K_i}^{\text{ext}}, n_{\partial K_i}), \quad (58)$$

where $w|_{K_i}^{\text{int}}$ and $w|_{K_i}^{\text{ext}}$ denote the values of w on ∂K_i considered from the interior and exterior of K_i , respectively.

The same numerical flux as in (12) is used including the boundary conditions specification.

3.8. Resulting FV-DG Scheme

Using both key ingredients, time derivative and numerical flux H together with treatment of the boundary condition for each time interval I_m , $m = 1, \dots, r$, we define the following form

$$\begin{aligned} A_h^m(\mu; w, \xi) &= \int_{I_m} (w', \xi) dt + (\{w\}_{m-1}, \xi|_{m-1}^+) - \int_{I_m} (s(w, \mu), \xi) dt \\ &\quad - \int_{I_m} \left(\sum_{K_i \in \mathcal{T}_h} \int_{\partial K_i \setminus \partial \Omega} H(w|_{K_i}^{\text{int}}, w|_{K_i}^{\text{ext}}, n_{\partial K_i}) \cdot \xi dS \right) dt \\ &\quad - \int_{I_m} \left(\sum_{K_i \in \mathcal{T}_h} \int_{\partial K_i \cap \partial \Omega} H(w|_{K_i}^{\text{int}}, w_{BC}, n_{\partial K_i}) \cdot \xi dS \right) dt, \quad w, \xi \in S_{h,\tau}^q, \end{aligned} \quad (59)$$

where $\partial K_i \setminus \partial \Omega$ denotes the part of the boundary of $K_i \in \mathcal{T}_h$ lying in the interior of Ω , $\partial K_i \cap \partial \Omega$ denotes the part of the boundary of $K_i \in \mathcal{T}_h$ lying on $\partial \Omega$ and w_{BC} is given by the boundary conditions.

Definition 2. We say that $w_h \in S_{h,\tau}^q$ is the approximate solution of (1) if

$$A_h^m(\mu_h; w_h, \xi) = 0 \quad \forall \xi \in S_{h,\tau}^q, \quad m = 1, \dots, r, \quad (60)$$

where $\mu_h \in S_{h,\tau}^q$, $\mu_h|_{\Omega \times I_m} = \mu_h^{m-1}$ and μ_h^{m-1} is defined in (55). We put $w_h|_0^- = \Pi w^0$, where Πw^0 is the piecewise constant projection of the initial condition w^0 on the mesh $\mathcal{T}_h$.

If we consider ξ are the basis functions of $S_{h,\tau}^q$, the relation (60) for each $m = 1, \dots, r$ is a system of nonlinear algebraic equations. The number of equations is equal $N_h^q := (q+1)\#\mathcal{T}_h$, here $\#\mathcal{T}_h$ is the number of triangles. The system (60) is solved for

each time level t_m . For this task we use a Newton-like method for the numerical solution of the compressible Euler equations, cf. ([23] Chapter 8).

Solution of FV-DG Scheme

As the result of the used numerical flux of Vijayasundaram type, we can linearize the form A_h^m given by (59) as

$$A_h^m(\mu; w, \xi) = A_{L,h}^m(\mu; w, w, \xi) - a_h^m(w, \xi) \quad \forall w, \xi \in S_{h,\tau}^q, \quad (61)$$

where the form $A_{L,h}^m$ is linear with respect to its third argument. The formula for $A_{L,h}^m$ contains the linearized numerical flux H and together with $a_h^m(w, \xi)$ can be found in [8].

Taking into account (48), we write system (60) as

$$A_h^m(\mu_h^{m-1}; w_h^m, \xi) = 0 \quad \forall \xi \in S_{h,\tau}^q, \quad m = 1, \dots, r, \quad (62)$$

which is system of algebraic equations (for each $m = 1, \dots, r$). Using linearization (61) we approximate w_h^m by the following sequence of approximations $w_h^{m,l}$, $l = 0, 1, 2, \dots$

$$w_h^{m,l+1} = w_h^{m,l} + d_h^{m,l}, \quad (63)$$

where $d_h^{m,l}$ solves

$$A_{L,h}^m(\mu; w_h^{m,l}, d_h^{m,l}, \xi) = -A_h^m(\mu_h^{m-1}; w_h^{m,l}, \xi) \quad \forall \xi \in S_{h,\tau}^q, \quad l = 0, 1, \dots \quad (64)$$

This scheme represents a *Newton-like* method. Here the matrix given by the linearized form $A_{L,h}^m$ in (64) is used instead of the Jacobi matrix in the original Newton. For the solution of the linear system (64) the GMRES method with block ILU(0) preconditioner is used.

To improve the convergence the adaptive *damping factor* $\lambda \in (0, 1]$ in (63) is used $w_h^{m,l+1} = w_h^{m,l} + \lambda d_h^{m,l}$. More details about adaptive dumping factor can be found in [25,26]. In [8] we discuss stopping criteria for the used iterative solver.

4. Results

We present the comparative study of the proposed Splitting schemes I and II from the numerical point of view. The described methods are compared on domains with different geometries. Several meshes grading from coarse to fine are used. Let us stress that we have tested two different numerical methods on the representative sample of computational domains. Moreover, the software for the Finite Volume Method was written by P. Kubera (the first author), while the software for the Discontinuous Galerkin Method was written by V. Dolejší and it was used as the black box with his permission. The detailed description of Dolejší's method can be found in [8] and generally in [23]. In each splitting scheme I and II the two different approaches for the determination of μ , namely Bornemann-Rasch and Dijkstra's algorithms, were used. So we gained a more general insight on the behavior of different methods for pedestrian movement. See also the further comparison with the new Cellular Automaton model proposed recently by the authors in [15].

Quantities we are interested in are the distribution of the density ρ (the dimension being pedestrian per square meter in the distinction of the fluid dynamics density in kg/m^3) over the domain and the evacuation time in seconds T_{evac} , i.e., the time when the domain is considered to be empty of any people. See the first paragraph in Section 3 for the details how the results have been computed. For each of test cases we use two methods for the determination of μ in (4). For the numerical solution of ϕ in the integral Formulation (6) the abbreviation DA is used and it suggests the Dijkstra's algorithm. For the numerical solution of ϕ in the differential Formulation (8) the abbreviation BR indicates the Bornemann-Rasch algorithm.

Let us define the quantity $M(t_m)$

$$M(t_m) := \int_{\Omega} \rho(x, t_m) \, dx, \quad (65)$$

giving the number of pedestrians in the domain Ω at the time instant t_m . The quantity $M(t_m)$ is approximated numerically using the computed piecewise constant approximation of the density at time t_m . We consider the domain Ω to be empty if the quantity $M(t_m) \leq \#PEDS$, where $\#PEDS$ is a prescribed number. We set $\#PEDS = 2$ for all methods.

The parameters of the model are listed below. In the power law for isentropic gases $p = p_0(\rho/\rho_0)^\gamma$ we set $p_0 = \rho_0 = 1$, $\gamma = 2$ (See the definition of fluxes f_s in (1)). Further we set $v_{max} = 2$, $\alpha = 7.5$, $\rho_{max} = 9$ in (3) and $\tau = 0.61$ in the source term s in (1). We use our original numerical flux H of Vijayasundaram type in (24) and (59), for its definition see [8,22].

4.1. Domain with One Obstacle

The first example is the evacuation of the rectangular domain 40×10 m with one circular obstacle, see Figure 1 (top). The outflow Γ_o is assumed on the entire vertical right hand side. The wall boundary Γ_w is assumed elsewhere. This can simulate the open space amphitheatre with a stage in the case when the people are running out of the amphitheatre through a broken wall denoted as the outflow Γ_o . Initial condition was set

$$\rho_0 = \begin{cases} 2 & x \in [0, 20] \times [0, 10], \\ 0 & \text{elsewhere,} \end{cases} \quad \text{and } v^0 = 0. \quad (66)$$

Figure 1. Examples of finite volume meshes with one circular obstacle at point $[32, 5]$ with radius $R = 2$ m (top) and with three circular obstacles at points $[32, 5]$, $[37, 2.5]$ and $[37, 7.5]$ with radius $R = 1$ m (bottom).

The results concerning the evacuation time T_{evac} are summarized in Table 1. We see that the difference in T_{evac} is less than two seconds.

Table 1. Domain with one obstacle, FV (Splitting Scheme I-left) and FV-DG (Splitting Scheme II-right)-Comparison of computations on series of meshes, $NELEM = \#\mathcal{T}_h$, T_{evac}^{BR} = evacuation time with Bornemann-Rasch algorithm for the determination of ϕ in (8), T_{evac}^{DA} = evacuation time with Dijkstra's algorithm for the determination of ϕ in (6).

$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]	$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]
1114	35.0	35.1	1126	35.2	36.2
2148	34.7	34.7	2132	35.2	36.1
4610	34.9	34.7	4627	35.0	35.5
9056	34.8	34.7	9082	34.8	35.4

In Figure 2, the evolution in time $t = 10, 15, 20$ s of the pedestrian density is drawn. The intuitive movement of pedestrians towards the right hand side can be observed. In each row of figures, the FV and the FV-DG methods are compared. We see all methods are in a good agreement. In Figure 3 the time evolution of the total number of pedestrians $M(t_m)$ is shown.

Figure 2. Domain with one obstacle, FV(BR), FV(DA) (top) and FV-DG(BR), FV-DG(DA) (bottom). Density distribution at time $t = 10, 15, 20$ s obtained on the finest meshes.

Figure 3. Domain with one circular obstacle, the time evolution of the total number of pedestrians $M(t_m)$ for the finest grid $\#T_h = 9056$ (FVM) and $\#T_h = 9082$ (FV-DG), total view (left) and the detailed view close to the end of the evacuation (right).

4.2. Domain with Three Obstacles

Here we present results for the simulation of the evacuation of the rectangular domain 40×10 m with three circular obstacles, see Figure 1 (bottom). We suppose boundary and initial conditions and also parameters of the model to be the same as for one circular obstacle above.

Computational results are shown in Table 2. We can see the difference in T_{evac} being smaller than one second for all methods on all tested meshes. The evolution of the pedestrian density in time is depicted in Figure 4. The intuitive behaviour with the high-density

domain in front of the obstacles can be seen for all methods. The dependence of the total number of pedestrians $M(t_m)$ for all tested methods is shown in Figure 5. The curves almost coincide (left). The small difference can be seen on the close view (right). We see that all methods are in a good agreement.

Table 2. Domain with three obstacles, FV (left) and FV-DG (right)-Comparison of computations on the series of meshes, $NELEM = \#\mathcal{T}_h$, T_{evac}^{BR} = evacuation time with Bornemann-Rasch algorithm, T_{evac}^{DA} = evacuation time with Dijkstra's algorithm.

$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]	$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]
1622	34.5	34.9	1653	35.0	35.6
2653	34.6	34.5	2616	34.9	35.5
7115	34.7	34.5	7084	34.9	35.1
11,660	35.0	34.6	11,636	34.7	35.0

Figure 4. Domain with three obstacles, FV(BR), FV(DA) (top) and FV-DG(BR), FV-DG(DA) (bottom). Density distribution at time $t = 10, 15, 20$ s obtained on the finest meshes.

Figure 5. Domain with three circular obstacles, the time evolution of the total number of pedestrians $M(t_m)$ for the finest grid $\#\mathcal{T}_h = 11,660$ (FV) and $\#\mathcal{T}_h = 11,636$ (FV-DG), total view (left) and the detailed view close to the end of the evacuation (right).

4.3. H-Shape Domain Evacuation

Further we present an evacuation through a narrowing followed by an empty space with the non-symmetric exit, see Figure 6. The so-called H-shape domain represents an example of the non-convex domain. The convexity of the domain does not play any role in the design of the proposed numerical methods. The geometry is defined

$$\Omega = \{(0, 60) \times (0, 25)\} \setminus \{ \{ [40, 45] \times [0, 7] \} \cup [40, 45] \times [18, 25] \} \}.$$

The outflow boundary condition is prescribed on

$$\Gamma_o = \{x; x_1 = 60, x_2 \in (0, 5)\}$$

and the wall boundary condition on $\Gamma_w = \partial\Omega \setminus \bar{\Gamma}_o$. We prescribe the initial condition

$$\rho^0 = \begin{cases} 2 & x \in [0, 20] \times [0, 25], \\ 0 & \text{elsewhere,} \end{cases} \quad \text{and } v^0 = \mathbf{0}. \quad (67)$$

Figure 6. H-shape domain, examples of the computational grids (FV-DG) with $\#\mathcal{T}_h = 2941$ (left) and $\#\mathcal{T}_h = 12,742$ (right) elements, the exits at the right-hand side bottom corner are marked by the blue color.

Table 3 shows the computational results obtained by the FV and by the FV-DG methods. See the description of table, the use of the DA or BR algorithm in the pedestrian flow model (both FV and FV-DG) causes more significant differences in the evacuation time T_{evac} . On the other hand the FV and FV-DG model with the same type of the computation of the desired direction of motion (i.e., DA or BR) produce the T_{evac}^{BR} and T_{evac}^{DA} being in a good agreement. The dependence of the pedestrian evacuation model on the use of the DA or BR algorithms has to be further studied.

Table 3. H-shape domain, FV (left) and FV-DG (right), $NELEM = \#\mathcal{T}_h$, T_{evac}^{BR} = evacuation time with Bornemann-Rasch algorithm, T_{evac}^{DA} = evacuation time with Dijkstra's algorithm.

$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]	$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]
2994	90.8	81.0	2941	87.5	79.1
12,835	90.5	81.0	12,742	90.5	80.0

In Figure 7 the density evolution for selected time instants is depicted. Here the local increase of the pedestrian density in the left part of the narrowing and also close to the exit of the domain is noticeable. In Figure 8 the evolution of number of pedestrian in the domain is shown. The main differences are due to the use of the DA or BR algorithms.

Figure 7. H-shape domain, FV(BR), FV(DA) (top) and FV-DG(BR), FV-DG(DA) (bottom). Density distribution obtained on the finest meshes at time $t = 5, 15, 30, 45$ s.

Figure 8. H-shape domain, the time evolution of the total number of pedestrians $M(t_m)$ for the finest grid (left) and for the coarsened grid (right).

4.4. T-Shape Domain Evacuation

Here we present numerical example for the multiple exits non-convex domain, the so-called T-shape domain, see Figure 9 for details. The x_1 -coordinates and the x_2 -coordinates of the corners are $\{-14, -10, -4, 4, 10, 14\}$ and $\{-16, 0, 2, 4, 6\}$, respectively.

The outflow Γ_o is situated on the vertical right- and left-hand sides of $\partial\Omega$, i.e.,

$$\Gamma_o = \{x = (x_1, x_2); x_1 = -14, x_2 \in (0, 2) \cup (4, 6)\} \cup \{x = (x_1, x_2); x_1 = 14, x_2 \in (0, 2) \cup (4, 6)\}.$$

Moreover, the wall boundary is obviously $\Gamma_w = \partial\Omega \setminus \bar{\Gamma}_o$. The initial condition is set

$$\rho^0 = \begin{cases} 2 & x \in [-4, 4] \times [-16, -6], \\ 0 & \text{elsewhere,} \end{cases} \quad \text{and } v^0 = \mathbf{0}. \quad (68)$$

Figure 9. T-shape domain, computational grids (FV-DG) with $\#\mathcal{T}_h = 2578$ (left) and $\#\mathcal{T}_h = 10,848$ (right) elements, the exits are marked by the blue color.

Table 4 shows the computational results for the T-shape domain. There is a small difference in T_{evac} between FV and FV-DG but the results are still in a good agreement from the practical point of view. The evolution of number of pedestrians in T-shape domain is shown in Figure 10. The pedestrian density distribution over the domain for different time instants is depicted in Figure 11.

Table 4. T-shape domain, FV (left) and FV-DG (right), $NELEM = \#\mathcal{T}_h$, T_{evac}^{BR} = evacuation time with Bornemann-Rasch algorithm, T_{evac}^{DA} = evacuation time with Dijkstra's algorithm.

$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]	$NELEM$	T_{evac}^{BR} [s]	T_{evac}^{DA} [s]
2678	26.7	25.4	2578	28.1	27.9
10,310	26.7	25.2	10,848	28.4	26.6

Figure 10. T-shape domain, the time evolution of the total number of pedestrians $M(t_m)$ for the finest grid $\#\mathcal{T}_h = 10,310$ (FVM) and $\#\mathcal{T}_h = 10,848$ (FV-DG), total view (left) and the detailed view close to the end of the evacuation (right).

Figure 11. T-shape domain, FV(BR), FV(DA) (**top**) and FV-DG(BR), FV-DG(DA) (**bottom**). Density distribution obtained on the finest meshes at time $t = 9, 15$ s.

The Figure 12 shows the trajectories of “selected” pedestrians. Trajectories of the pedestrians are given by the solution of the differential equation

$$\frac{dx(t)}{dt} = v(x, t), \quad (69)$$

with initial condition $x(0) = x^0$. The initial condition arranges the pedestrian selection. The Equation (69) is solved by the forward Euler method. Here we use the piecewise constant approximation of $v(x, t)$ obtained from the numerical solution $\mathbf{w}_h^m$ and w_h (see Definitions 1 and 2). The trajectories in Figure 12 show us the influence of the density distribution on the movement of pedestrians and their trajectory, since μ depends on the density.

Figure 12 (blue trajectories) shows the situation, where more distant outflow is selected, because of the pedestrian density distribution. The initial positions is set $x^0 = [\pm 0.4, -12]$. Pedestrians from this position cross the line $[\cdot, 0]$ at time $t = 9$ s and reach the more distant exit at time $t = 18$ s. The reason is the density in the corners in time $t = 9$ s is quite high and pedestrians prefer (dependence of μ on the density) to get round domains with high density. Whereas pedestrians with initial position $x^0 = [\pm 1, -16]$ cross the line $[\cdot, 0]$ later at time $t = 15$ s and reach the nearer exit at time $t = 23$ s. The domains with high density at time $t = 15$ s are smaller and thus nearer exits are preferred. This corresponds to intuitive imagination of the pedestrian behaviour in real situations.

Figure 12. T-shape domain pedestrian trajectories. First row FV in order Bornemann-Rasch algorithm and Dijkstra's algorithm, second row FV-DG method in the same order. Pedestrian trajectories for starting position $x^0 = [\pm 0.4, -12]$ (blue color) and $x^0 = [\pm 1, -16]$, respectively (green color). The pedestrians starting at $x^0 = [\pm 0.4, -12]$ cross the line $[\cdot, 0]$ at time $t = 9$ s and reach the more distant exit at time $t = 18$ s. The pedestrians starting at $x^0 = [\pm 1, -16]$ cross the line $[\cdot, 0]$ at time $t = 15$ s and reach the nearer exit at time $t = 23$ s.

5. Conclusions

The paper deals with the comparative study of two splitting methods-Splitting Method I and Splitting Method II. They were described and numerically tested for solving several pedestrian flow problems. The FV approach was compared with FV-DG approach. The former use the piece-wise constant approximation together with the explicit time discretization and is advantageous from the implementation point of view. The latter is based on the discontinuous piece-wise polynomial approximation and uses the implicit discretization by space time discontinuous Galerkin method. The FV-DG approach requires the solution of the system of nonlinear algebraic equations, but on the other hand the implicit time discretization together with the high polynomial degree enable us to use the longer time steps. Further the FV-DG approach allow us to use different space grids on different time levels which is useful for adaptive methods.

The alternative method to that proposed by Bornemann and Rasch for the computation of the desired direction of move μ in (4) based on the Dijkstra's algorithm was proposed and tested. This method can be considered as a competitive method to the Bornemann-Rasch algorithm for the solution of the Eikonal equation. The key difference is in leaving out the differential formulation given by the Eikonal equation (8) and replacing its iterative solution by the modification of the Dijkstra's algorithm for the shortest path in a graph. The curvilinear integral in (6) is minimized in the class of piece-wise linear curves given by edges of the triangulation. Compared with the Bornemann-Rasch algorithm (BR) the proposed method has several different properties. Firstly, the method is tolerance free in the distinction from the iterative BR algorithm, where the stopping criterion must be prescribed. Secondly, in the proposed method the computation of the gradient of ϕ in vertex P can be avoided by using the direction of the edge which lies on the shortest path from P to the outflow.

The special advantage of presented simulations is that the system of partial differential equations is solved by two different software systems with the results being in the good agreement. This is related to the key word "verification" of the model, i.e., to the question,

if the equations are solved “right”. Another step is the question, if we solve the “right” equations, which is related to the notion “validation” of the model. This suppose the comparison with the experimental data. Two ingredients of the proposed methods are original. We use the numerical flux which is the original modification of the Vijayasundaram numerical flux for the Euler equations. Despite the fact that in the distinction from the Euler fluxes the fluxes for the pedestrian flow equations are not homogeneous, we were able to propose the numerical flux of the Vijayasundaram type which is easy to be coded. For details see [8,22]. Further we are able to avoid the numerical solution of the Eikonal equation. This is related to the notion of the fastest path in a graph. The details can be found in [8,27]. We use the triangular finite volumes and the numerical flux mentioned above. In [3] the dual finite volumes are used and the HLL numerical is flux applied. The results can be mutually compared.

Author Contributions: Conceptualization, P.K. and J.F.; Formal analysis, J.F.; Software, P.K.; Visualization, P.K.; Writing—original draft, J.F.; Writing—review and editing, P.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by IGA UJEP, grant No. UJEP-IGA-TC-2019-53-02-2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The FV-DG solution of the problem was computed by the software written by Vít Dolejší (see [23]). The authors acknowledge his help in the adaptation of the software for the pedestrian flow problem.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, R.; Fu, Z.; Schadschneider, A.; Wen, Q.; Chen, J.; Liu, S. Modeling the effect of visibility on upstairs crowd evacuation by a stochastic FFCA model with finer discretization. *Phys. A Stat. Mech. Appl.* **2019**, *531*, 121723. [\[CrossRef\]](#)
2. Fu, Z.; Zhan, X.; Luo, L.; Schadschneider, A.; Chen, J. Modeling fatigue of ascending stair evacuation with modified fine discrete floor field cellular automata. *Phys. Lett. A* **2019**, *383*, 1897–1906. [\[CrossRef\]](#)
3. Twarogowska, M.; Goatin, P.; Duvigneau, R. Macroscopic modeling and simulation of room evacuation. *Appl. Math. Model.* **2014**, *38*, 5781–5795. [\[CrossRef\]](#)
4. Buchmueller, S.; Weidmann, U. Parameters of pedestrians, Pedestrian Traffic and Walking Facilities. *IVT Schriftenreihe* **2007**, *132*. [\[CrossRef\]](#)
5. Bellomo, N.; Marasco, A.; Romano, A. From the modelling of driver’s behavior to hydrodynamics models and problems of traffic flow. *Nonlinear Anal. RWA* **2002**, *3*, 339–363. [\[CrossRef\]](#)
6. Venuti, F.; Bruno, L.; Bellomo, N. Crowd dynamics on a moving platform: Mathematical modelling and application to lively footbridges. *Math. Comp. Model.* **2007**, *45*, 252–269. [\[CrossRef\]](#)
7. Bellomo, N.; Dogbé, C. On the modelling crowd dynamics from scaling to hyperbolic macroscopic models. *Math. Model. Methods Appl. Sci.* **2008**, *18*, 1317–1345. [\[CrossRef\]](#)
8. Dolejší, V.; Felcman, J.; Kubera, P. FV-DG Method for the Pedestrian Flow Problem. *Comput. Fluids* **2019**, *183*, 1–15. [\[CrossRef\]](#)
9. Jiang, Y.; Zhang, P.; Wong, S.; Liu, R. A higher-order macroscopic model for pedestrian flows. *Phys. A Stat. Mech. Appl.* **2010**, *389*, 4623–4635. [\[CrossRef\]](#)
10. Payne, H. *Models of Freeway Traffic and Control*; Simulation Councils, Incorporated: La Jolla, CA, USA, 1971.
11. Whitham, G. *Linear and Nonlinear Waves*; Pure and Applied Mathematics; Wiley: Hoboken, NJ, USA, 1974.
12. Berres, S.; Ruiz-Baier, R.; Schwandt, H.; Tory, E. An adaptive finite-volume method for a model of two-phase pedestrian flow. *Netw. Heterog. Media* **2011**, *6*, 401–423. [\[CrossRef\]](#)
13. Dridi, M.H. Simulation of high density pedestrian flow: Microscopic model. *Open J. Model. Simul.* **2015**, *3*, 81–95. [\[CrossRef\]](#)
14. Marno, P. Crowded-Macroscopic and Microscopic Models for Pedestrian Dynamics. Ph.D. Thesis, University of Reading, Reading, UK, 2002.
15. Felcman, J.; Kubera, P. A cellular automaton model for a pedestrian flow problem. *Math. Model. Nat. Phenom.* **2021**, *16*, 11. [\[CrossRef\]](#)
16. Dogbe, C. On the modelling of crowd dynamics by generalized kinetic models. *J. Math. Anal. Appl.* **2012**, *387*, 512–532. [\[CrossRef\]](#)
17. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*; Springer: Berlin, Germany, 1997.
18. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [\[CrossRef\]](#)

19. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2001.
20. Bornemann, F.; Rasch, C. Finite-element Discretization of Static Hamilton-Jacobi Equations based on a Local Variational Principle. *Comput. Vis. Sci.* **2006**, *9*, 57–69. [[CrossRef](#)]
21. Felcman, J.; Dolejší, V.; Kubera, P. Discontinuous Galerkin Method for the Pedestrian Flow Problem. In *ICNAAM 2017 AIP Conference Proceedings 1978:1*; Simos, T.E., Tsitouras, C., Eds.; American Institute of Physics : Melville, NY, USA, 2018; pp. 0300191–0300194. [[CrossRef](#)]
22. Kubera, P.; Felcman, J. On a numerical flux for the pedestrian flow equations. *J. Appl. Math. Stat. Inform.* **2015**, *11*, 79–96. [[CrossRef](#)]
23. Dolejší, V.; Feistauer, M. *Discontinuous Galerkin Method*; Springer: New York, NY, USA, 2015.
24. Petrášová, T. Application of the Dijkstra's Algorithm in the Pedestrian Flow Problem. Bachelor's Thesis, Charles University in Prague, Prague, Czech Republic, 2016.
25. Deuflhard, P. *Newton Methods for Nonlinear Problems*; Springer Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2004; Volume 35.
26. Dolejší, V.; Roskovec, F.; Vlasák, M. Residual based error estimates for the space-time discontinuous Galerkin method applied to the compressible flows. *Comput. Fluids* **2015**, *117*, 304–324. [[CrossRef](#)]
27. Felcman, J.; Kubera, P. On the Eikonal Equation in the Pedestrian Flow Problem. In *ICNAAM 2016 AIP Conference Proceedings 1863:1*; Simos, T.E., Tsitouras, C., Eds.; American Institute of Physics : Melville, NY, USA, 2017; pp. 5600241–5600244. doi:10.1063/1.4992707.