

# Conservative Semi-Lagrangian Advection on Adaptive Unstructured Meshes

ARMIN ISKE and MARTIN KÄSER

**Abstract.** A conservative semi-Lagrangian method is designed in order to solve linear advection equations in two space variables. The advection scheme works with finite volumes on an unstructured mesh, which is given by a Voronoi diagram. Moreover, the mesh is subject to adaptive modifications during the simulation, which serves to effectively combine good approximation quality with small computational costs. The required adaption rules for the refinement and the coarsening of the mesh rely on a customized error indicator. The implementation of boundary conditions is addressed. Numerical results finally confirm the good performance of the proposed conservative and adaptive advection scheme.

## 1 Introduction

Many physical phenomena in transport processes are modelled by time-dependent hyperbolic conservation laws. Finite volume schemes provide well-established conservative methods for solving the governing advection equations. Moreover, high-order finite volume schemes were developed in order to treat high gradients and discontinuities of the solution.

The analysis of available standard methods for hyperbolic equations is typically dominated by time step restrictions, mainly for the sake of their stability. In contrast to Eulerian schemes, Lagrangian particle methods require often less restrictive conditions on the time step. On the other hand, however, Lagrangian methods are usually not conservative. A more comprehensive discussion on these and related aspects concerning Eulerian versus Lagrangian schemes for hyperbolic conservation laws is offered in the textbooks [6, 10, 11].

Considerable effort has been made recently in order to construct *conservative* semi-Lagrangian methods [9, 13, 15, 17]. But the resulting advection schemes are not adaptive. In fact, these papers' methods work with a rectangular grid, respectively, which is fixed throughout the simulation. However, in order to balance the method's approximation quality and its computational costs effectively, *adaptivity* is an essential requirement, especially when modelling multiscale phenomena.

In this paper, we propose a second order semi-Lagrangian finite volume method for passive advection, which is both conservative and adaptive. To this end, our method works with control volumes of an unstructured adaptive mesh, which is given by the Voronoi diagram of current nodes. The node set is corresponding to a set of moving particles. Moreover, the node set is subject to adaptive modifications during the simulation, and so is the Voronoi diagram to be updated accordingly. This requires customized adaption rules for the dynamic refinement and coarsening of the node set. The adaption rules rely on available local error estimates at the current nodes. Details on this are deferred to Section 5.

We remark that the proposed method can be viewed as an extension of the previous conservative but non-adaptive advection schemes by Scroggs & Semazzi [17] and by Phillips & Williams [13]. But adaptivity is one crucial feature of our method. Indeed, the effective node adaption serves to enhance our method's accuracy, and it allows us to cope with complicated geometries and discontinuities of the solution. This is confirmed by the numerical results in the test case scenarios of Section 6, where the good performance of the proposed advection scheme is illustrated.

The outline of this paper is as follows. A precise formulation of the mathematical problem and a preliminary discussion on the method's basic features are provided in the following Section 2. Relevant ingredients of the semi-Lagrangian finite volume method are then discussed in Section 3, where in particular a brief recap of Voronoi diagrams is provided. The construction of the method's mass conservation is explained in Section 4, where also the implementation of boundary conditions is discussed. Numerical results are provided in Section 6, and a final conclusion is drawn in Section 7.

## 2 Preliminary Discussion

In this paper, we regard *passive advection* equations, given by a scalar time-dependent *hyperbolic conservation law* of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot (au) = 0, \quad (1)$$

where for a compact time interval  $I = [0, T] \subset \mathbb{R}$ ,  $T > 0$ , and the computational domain  $\Omega = \mathbb{R}^2$ , the *velocity field*

$$a = a(t, x), \quad t \in I, x \in \Omega,$$

is assumed to be given. In this case, the scalar solution  $u : I \times \Omega \rightarrow \mathbb{R}$  of (1) usually corresponds to a physical quantity, such as density or concen-

tration. Especially when modelling physical phenomena in relevant applications, such as chemical tracer transportation, *mass conservation*, i.e.,

$$\frac{d}{dt} \int_{\mathbb{R}^2} u(t, x) dx = 0,$$

is always an important requirement.

In this paper, we consider solving (1) numerically, on the given initial condition

$$u(0, x) = u_0(x), \quad \text{for } x \in \Omega. \quad (2)$$

To this end, we work with a *conservative* semi-Lagrangian adaptive advection scheme, whose construction is a subject of the discussion in this paper. For the moment of the following discussion, we wish to avoid boundary conditions. To this end, we assume that the computational domain is the whole plane, i.e.,  $\Omega = \mathbb{R}^2$ . But later in Subsection 4.3, where the implementation of boundary conditions is explained, we drop this assumption.

The following construction of our method combines the particle-based semi-Lagrangian approach of the previous papers [1, 2, 3] with a finite volume scheme on adaptive unstructured meshes. We remark that the particle-based scheme of [1] is applied in [2] on a real-world test case scenario concerning chemical tracer transportation over the Arctic stratosphere. An extension of the scheme in [1] to *nonlinear* transport equations has recently been suggested in [3]. However, none of the *meshfree* particle schemes in [1, 2, 3] is conservative, which is a severe drawback in several relevant applications, such as the above mentioned tracer advection simulation.

Before we expand all relevant ingredients of our method in detail, let us briefly explain the basic ideas for the construction of the *conservative* scheme of this paper. Similar to [1, 2, 3], the discretization works with a finite set  $\Xi \subset \Omega$  of nodes, each of which corresponds to one flow particle at a time  $t \in I$ . Any *current* node set  $\Xi$  defines a unique Voronoi diagram  $\mathcal{V}_\Xi$ , which yields a partitioning of the computational domain  $\Omega = \mathbb{R}^2$  into  $\#\Xi$  finite control volumes, *Voronoi cells*, so that each cell in  $\mathcal{V}_\Xi$  contains exactly one node from  $\Xi$ .

Given the *cell average values* over the current Voronoi cells at time  $t$ , initially given by using a suitable node set  $\Xi$  along with the initial condition (2), the advection step  $t \rightarrow t + \tau$  in our scheme is accomplished as follows. For each current Voronoi cell  $V \in \mathcal{V}_\Xi$ , a corresponding *upstream cell*  $U \subset \Omega$  is constructed. The upstream cell  $U$  contains at time  $t$  the proportion of mass which is advected into the cell  $V$  during the time step  $t \rightarrow t + \tau$ . This duality relation between  $U$  and  $V$  is used in order to establish (local) mass

conservation by

$$\int_U u_h(t, x) dx = \int_V u_h(t + \tau, x) dx, \quad (3)$$

where  $u_h \approx u$  is an approximation to the solution  $u$  of the Cauchy problem (1), (2). Details on the construction of the upstream cells are discussed in Section 3.

Moreover, be it sufficient for the moment to remark that, in our second order advection scheme,  $u_h$  in (3) is a piecewise linear function over the current Voronoi diagram  $\mathcal{V}_\Xi$ . Details on the construction of  $u_h$  are explained in Subsections 4.1 and 4.2, whereas the implementation of boundary conditions is explained in Subsection 4.3. Finally, the adaption rules for the nodes in  $\Xi$  are discussed in Section 5.

### 3 Semi-Lagrangian Advection on Voronoi Cells

In this section, the construction of the above mentioned upstream cells is discussed. To this end, let us first recall some relevant ingredients from computational geometry, in particular Voronoi diagrams (see the textbook [14] for more details on Voronoi diagrams). For a fixed finite node set  $\Xi \subset \mathbb{R}^2$ , the *Voronoi diagram*  $\mathcal{V}_\Xi = \{V_\xi\}_{\xi \in \Xi}$  of  $\Xi$  is a planar graph, which yields a partitioning

$$\mathbb{R}^2 = \bigcup_{\xi \in \Xi} V_\xi$$

of the plane into *Voronoi cells* of the form

$$V_\xi = \left\{ x \in \mathbb{R}^2 : \min_{\zeta \in \Xi} \|x - \zeta\| = \|x - \xi\| \right\} \subset \mathbb{R}^2.$$

Figure 1 shows the Voronoi diagram  $\mathcal{V}_\Xi$  of a planar node set  $\Xi$ . For any  $\xi \in \Xi$ , its corresponding Voronoi cell  $V_\xi$  is a convex polytope containing all points in the plane whose nearest point in  $\Xi$  is  $\xi$ . The vertices of  $V_\xi$  are said to be the *Voronoi vertices* of  $V_\xi$ . The nodes in  $\Xi$  whose Voronoi cells are adjacent to the Voronoi cell  $V_\xi$  are said to be the *Voronoi neighbours* of  $\xi$ .

#### 3.1 Computation of Upstream Vertices

The construction of any upstream cell  $U$  relies on the construction of its *upstream vertices*. In order to explain the construction of the upstream vertices, let  $V \equiv V_\xi$  be the Voronoi cell of any node  $\xi \in \Xi$ . Moreover, let

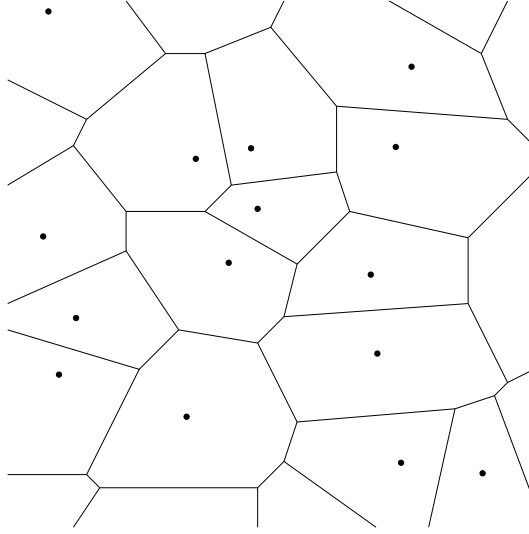


Figure 1: Voronoi diagram  $\mathcal{V}_{\Xi}$  of a node set  $\Xi$ .

$\mathbf{v}_1, \dots, \mathbf{v}_n$  denote the  $n$  Voronoi vertices of  $V$ , labelled in a counter-clockwise ordering. For the purpose of constructing the upstream cell  $U$  corresponding to  $V$ , we first approximate the upstream positions of the Voronoi vertices  $\mathbf{v}_1, \dots, \mathbf{v}_n$  of  $V$ . This is accomplished as follows.

Let  $\mathbf{u}^-$  be the exact *upstream position* of any vertex  $\mathbf{v}$  of  $V$  satisfying  $u(t, \mathbf{u}^-) = u(t + \tau, \mathbf{v})$ . The upstream position  $\mathbf{u}^-$  of  $\mathbf{v}$  can be viewed as the location of a flow particle at time  $t$ , which by traversing along its trajectory arrives at the vertex  $\mathbf{v}$  at time  $t + \tau$ , see Figure 2. In case of passive advection, the shape of the particles' flow trajectories are entirely and uniquely determined by the given velocity field  $a = a(t, x)$ .

Note that the exact upstream position  $\mathbf{u}^-$  of  $\mathbf{v}$  is usually unknown. In order to compute an approximation  $\mathbf{u}$  to the upstream point  $\mathbf{u}^-$ , this amounts to numerically solving the *ordinary differential equation* (ODE)

$$\dot{x} = \frac{dx}{dt} = a(t, x), \quad (4)$$

with initial condition  $x(t + \tau) = \mathbf{v}$ , so that the solution  $x$  of the initial value problem satisfies  $x(t) = \mathbf{u}^-$ .

Adopting some standard notation from dynamical systems [4], we express the upstream position  $\mathbf{u}^-$  of the vertex  $\mathbf{v}$  as

$$\mathbf{u}^- = \Phi^{t, t+\tau} \mathbf{v}, \quad (5)$$

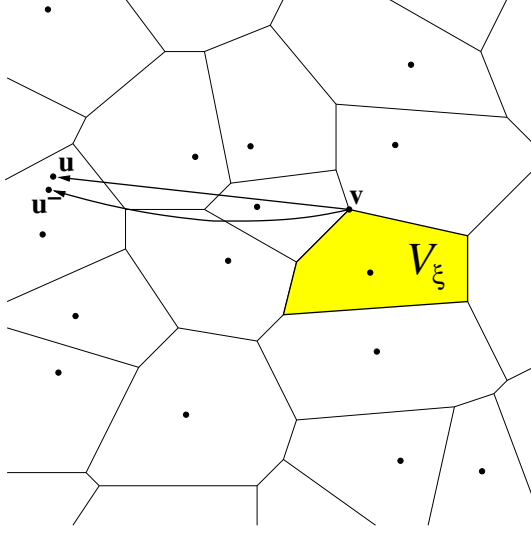


Figure 2: The backward trajectory from the Voronoi vertex  $\mathbf{v} \in V_\xi$  to its upstream point  $\mathbf{u}^-$ , and its linear approximation, leading to  $\mathbf{u}$ .

where  $\Phi^{t,t+\tau} : \Omega \rightarrow \Omega$  denotes the *continuous evolution* of the (backward) flow of (4). An equivalent formulation for (5) is given by  $\mathbf{v} = \Phi^{t+\tau,t} \mathbf{u}^-$ , since  $\Phi^{t+\tau,t}$  is the inverse of  $\Phi^{t,t+\tau}$ .

Likewise, for the sake of notational simplicity in the following of this text, it is convenient to express the approximation  $\mathbf{u}$  of  $\mathbf{u}^-$  as

$$\mathbf{u} = \Psi^{t,t+\tau} \mathbf{v}, \quad (6)$$

where  $\Psi^{t,t+\tau} : \Omega \rightarrow \Omega$  is the *discrete evolution* of the flow. Note that the operator  $\Psi^{t,t+\tau}$  is usually given by any suitable numerical method for solving the above ODE (4). This, however, is only a generic definition for  $\Psi^{t,t+\tau}$ .

In order to be more concrete on the upstream point approximation, we remark that our implementation works with the fixpoint iteration

$$\beta^{(k+1)} = \tau \cdot a \left( t + \frac{\tau}{2}, \mathbf{v} - \frac{\beta^{(k)}}{2} \right), \quad k = 0, 1, 2, \dots, \quad (7)$$

where we let  $\beta^{(0)} = 0$ . This yields after merely a few iterations a sufficiently accurate linear approximation  $\beta$  of the backward trajectory at  $\mathbf{v}$ . In this case, the desired approximation to the upstream point  $\mathbf{u}^-$  is given by

$$\mathbf{u} = \mathbf{v} - \beta,$$

see Figure 2. We remark that the iteration (7) has already been recommended in the seminal paper on semi-Lagrangian methods by Robert [16], see also [11, equation (7.66a)].

### 3.2 Construction of Upstream Cells

Recall the generic notation of the discrete evolution  $\Psi^{t,t+\tau}$  in (6). Note that for any Voronoi vertex  $\mathbf{v}_j$ ,  $1 \leq j \leq n$ , of a cell  $V \in \mathcal{V}_\Xi$  we obtain an approximation to its corresponding upstream point by

$$\mathbf{u}_j = \Psi^{t,t+\tau} \mathbf{v}_j, \quad 1 \leq j \leq n.$$

By connecting the sequence  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n, \mathbf{u}_1$  of consecutive upstream point approximations we obtain a closed polygon, denoted by  $U$ . This defines the *upstream cell* of the (bounded) Voronoi cell  $V$ . Again, for the sake of notational simplicity, it is convenient to use the notation

$$U = \Psi^{t,t+\tau} V, \quad \text{for } V \in \mathcal{V}_\Xi. \quad (8)$$

Figure 3 shows an example for an upstream cell  $U \equiv U_\xi$ , corresponding to a Voronoi cell  $V \equiv V_\xi$ ,  $\xi \in \Xi$ .

In order to be able to facilitate the following computations, we require that every upstream cell  $U$  is *convex* and *nondegenerate* (details on this are explained in Subsection 4.2). We achieve this by introducing a time step restriction on  $\tau$ , similar to the one suggested in [13, 17]. To be more precise, we first select one initial value  $\tau_0 > 0$ , which is gradually reduced by letting  $\tau_{k+1} = \tau_k/2$ ,  $k \geq 0$ , whenever at least one non-convex or degenerate upstream cell occurs. In our numerical experiments, where we let  $\tau_0 = 0.1$ , this iteration requires only at most four steps. Note that by using this time step restriction, the duality relation (8) is well-defined.

We remark that our condition on the time step  $\tau$  is, when compared with Eulerian schemes, much less restrictive. Indeed, the stability of explicit Eulerian methods is typically dominated by rather restrictive CFL conditions, which in turn leads to very small time steps, especially when working with adaptive meshes containing very small finest control volumes [10].

Now according to the construction of the upstream cells, and in view of mass conservation, any upstream cell  $U$  is supposed to contain the proportion of mass, which is to be advected into its corresponding Voronoi cell  $V = \Psi^{t+\tau,t} U$  at the (current) time step  $t \rightarrow t + \tau$ . This requirement is accommodated by (3).

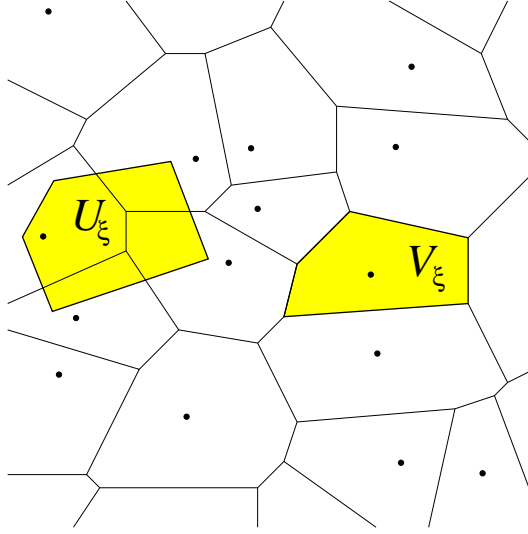


Figure 3: Voronoi cell  $V_\xi$  of the node  $\xi \in \Xi$ , and its upstream cell  $U_\xi$ .

Yet it remains to determine the *total mass*

$$m_U \equiv m_U(t) = \int_U u_h(t, x) dx$$

contained in any single upstream cell  $U$ , at time  $t$ , from the given mass distribution in the current Voronoi diagram  $\mathcal{V}_\Xi$ . To this end, we compute the intersections of  $U$  with its overlapping Voronoi cells in  $\mathcal{V}_\Xi$ . Further details on this are discussed in the following section.

## 4 Mass Conservation by Construction

Given the current Voronoi diagram  $\mathcal{V}_\Xi$ , at time  $t$ , let  $\mathcal{U}_\Xi = \{U_\xi\}_{\xi \in \Xi}$  denote the corresponding set of current upstream cells, each of which is given by the duality relation (8). Note that the collection  $\mathcal{U}_\Xi$  of upstream cells yields (besides the Voronoi diagram  $\mathcal{V}_\Xi$ ) yet another partitioning of the plane, so that we have

$$\mathbb{R}^2 = \bigcup_{\xi \in \Xi} U_\xi.$$

We make use of the duality (8) between the Voronoi cells and the upstream cells in order to design an advection scheme which is *mass conservative by construction*.



## 4.1 Reconstruction from Cell Average Values

The modelling taken in this approach works with piecewise linear functions over the Voronoi diagram  $\mathcal{V}_\Xi$ , so that we obtain a *second order* finite volume scheme. In order to be more precise on this, the current approximation  $u_h$  to the solution  $u$  of the Cauchy problem (1), (2) is, at any time  $t$ , an element of the linear function space

$$\mathcal{S}_1(\mathcal{V}_\Xi) = \{u_h : \Omega \rightarrow \mathbb{R} : u_h|_{V \cap \Omega} \text{ linear; for all } V \in \mathcal{V}_\Xi\}$$

containing all piecewise linear functions on  $\mathcal{V}_\Xi \cap \Omega$ .

Starting point for computing the numerical solution  $u_h(t, \cdot) \in \mathcal{S}_1(\mathcal{V}_\Xi)$ , at a time  $t$ , are known *cell average values*

$$\bar{u}_V(t) \approx \frac{1}{|V|} \int_V u(t, x) dx, \quad \text{for all } V \in \mathcal{V}_\Xi,$$

over the current Voronoi diagram  $\mathcal{V}_\Xi$ , where  $|V|$  denotes the volume of  $V$  in  $\mathbb{R}^2$ . Initially, for a *suitable* set  $\Xi$  of nodes, we let  $\bar{u}_{V_\xi}(0) = u_0(\xi)$  for all  $\xi \in \Xi$ , by using the initial condition (2).

Given  $\{\bar{u}_V(t)\}_{V \in \mathcal{V}_\Xi}$ , we wish to determine  $u_h(t, \cdot) \in \mathcal{S}_1(\mathcal{V}_\Xi)$  from the cell averages, such that

$$\bar{u}_V(t) = \frac{1}{|V|} \int_V u_h(t, x) dx, \quad \text{for all } V \in \mathcal{V}_\Xi, \quad (9)$$

holds. This is accomplished as follows.

For any Voronoi cell  $V_\xi \in \mathcal{V}_\Xi$ , we first determine the best approximation  $u^* \in \mathcal{P}_1$  satisfying

$$\min_{u \in \mathcal{P}_1} \sum_{\nu \in \mathcal{N}} |\bar{u}_\nu - u(\nu)|^2 = \sum_{\nu \in \mathcal{N}} |\bar{u}_\nu - u^*(\nu)|^2.$$

Here,  $\mathcal{P}_1$  is the space of all linear bivariate polynomials, and  $\mathcal{N}$  denotes the set of Voronoi neighbours of  $\xi$ . Hence, the function  $u^*$  is the *least squares fit* of the cell average values  $\bar{u}_\nu$ ,  $\nu \in \mathcal{N}$ , in the neighbourhood of  $V_\xi$ .

Next, we determine a constant  $c$  such that the function  $u_h = u^* + c \in \mathcal{P}_1$  satisfies (9). This is achieved by letting

$$c = \frac{1}{|V|} \cdot \left( m_V - \int_V u^* dx \right), \quad (10)$$

where  $m_V = |V| \cdot \bar{u}_V(t)$  is the total mass in the cell  $V$ . In other words, the constant  $c$  in (10) adjusts the linear least square fit  $u^*$  in order to guarantee mass conservation (9) locally.

Additionally, we constrain the slopes of the linear approximation  $u_h$  by requiring the two conditions

$$\begin{aligned} \min_{x \in V} u_h(x) &\geq \min_{\nu \in \mathcal{N}} \bar{u}_\nu \\ \max_{x \in V} u_h(x) &\leq \max_{\nu \in \mathcal{N}} \bar{u}_\nu, \end{aligned} \tag{11}$$

which can be viewed as a *slope limiter*. Slope limiters are typically employed in TVD (Total Variation Diminishing) schemes in order to avoid spurious oscillations of the solution. In order to match the restrictions (11), this needs merely a local correction of  $u_h$ , so that the resulting modification of  $u_h$  continues to satisfy (9). To this end, we follow along the lines of the construction in [19].

## 4.2 Conservative Advection of Cell Average Values

Having computed  $u_h(t, \cdot)$  from the current cell averages  $\{\bar{u}_V(t)\}_{V \in \mathcal{V}_\Xi}$ , we are in a position to compute, for any upstream cell  $U$ , its total mass

$$m_U = \int_U u_h(t, x) dx$$

which is advected into the corresponding Voronoi cell  $V = \Psi^{t+\tau, t}U$ . To this end, we first decompose  $U$  into smaller tiles by computing the intersections between  $U$  and its overlapping Voronoi cells in  $\mathcal{V}_\Xi$  (see Figure 4).

In order to be able to facilitate the computation of these intersections, the upstream cell  $U$  is supposed to be *convex*, which explains the time step restriction suggested in the previous Section 3.2. With assuming convexity for  $U$ , this namely allows us to use the efficient intersection algorithm proposed in [12]. The algorithm in [12] requires merely  $\mathcal{O}(p + q)$  operations for the intersection of two *convex* polygons with  $p$  and  $q$  vertices, respectively. In contrast, computing the intersection of two *non-convex* polygons would cost  $\mathcal{O}(p \cdot q)$  operations. For further details on this, see [12].

Now the intersections between  $U$  and the cells in the Voronoi diagram  $\mathcal{V}_\Xi$  yield by

$$U = \bigcup_{V \in \mathcal{V}_\Xi} (U \cap V)$$

a partitioning of  $U$  into merely a *small* number of tiles, which requires only local computations. Indeed, this tiling for  $U$  is given by all *non-empty* intersections between  $U$  and Voronoi cells in  $\mathcal{V}_\Xi$ . Figure 4 shows one example, where one upstream cell  $U$  is decomposed into four tiles,  $U_1, U_2, U_3, U_4$ , so that  $U = U_1 \cup U_2 \cup U_3 \cup U_4$ .

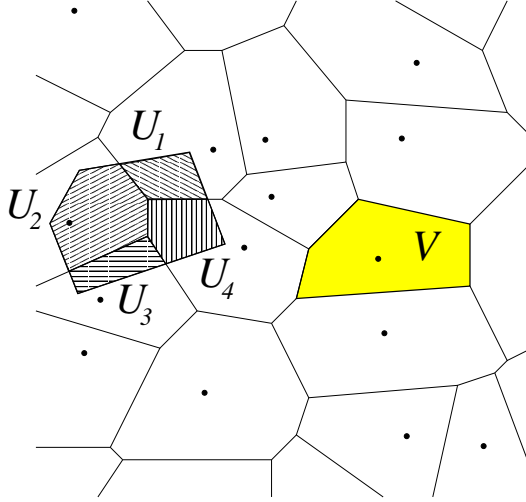


Figure 4: The decomposition of an upstream cell  $U$  into four tiles,  $U_1, U_2, U_3$  and  $U_4$ .

The current approximation  $u_h(t, \cdot) \in \mathcal{S}_1(\mathcal{V}_\Xi)$  to the solution  $u$  in (1) is now used in order to determine the total mass  $m_U$  in  $U$ . This is done as follows. Note that the restriction of  $u_h(t, \cdot)$  to any Voronoi cell  $V \in \mathcal{V}_\Xi$  is a linear function by definition. So, in particular the restriction of  $u_h(t, \cdot)$  to any non-empty tile  $U \cap V$  is linear. This allows us to compute, for any  $V \in \mathcal{V}_\Xi$ , at time  $t$ , the current mass

$$m_{U \cap V} = \int_{U \cap V} u_h(t, x) dx, \quad \text{for } V \in \mathcal{V}_\Xi, \quad (12)$$

of the linear function  $u_h$  over the tile  $U \cap V$  exactly. If  $U \cap V$  is empty, then we have  $m_{U \cap V} = 0$ . This in turn yields the total mass

$$m_U = \sum_{V \in \mathcal{V}_\Xi} m_{U \cap V} \quad (13)$$

in the upstream cell  $U$  at time  $t$ . In the advection step  $t \rightarrow t + \tau$ , the total mass  $m_U$  is advected from the upstream cell  $U$  into the Voronoi cell  $V$ . We establish this *local mass conservation* by replacing the current cell average values  $\bar{u}_V(t)$  in (9) by the updates

$$\bar{u}_V(t + \tau) = \frac{1}{|V|} m_U, \quad \text{for all } V \in \mathcal{V}_\Xi. \quad (14)$$

Altogether, we obtain a *conservative* semi-Lagrangian advection scheme, each of whose time steps  $t \rightarrow t + \tau$  is given by the following algorithm.

**Algorithm 1 (Conservative Semi-Lagrangian Advection).**

**INPUT:** Time step size  $\tau > 0$ , node set  $\Xi \subset \Omega$ , Voronoi diagram  $\mathcal{V}_\Xi$ , and cell average values  $\bar{u}_V(t)$  for all  $V \in \mathcal{V}_\Xi$ .

- Compute piecewise linear  $u_h(t, \cdot) \in \mathcal{S}_1(\mathcal{V}_\Xi)$  satisfying (9) and (11).
- **FOR** each  $V \in \mathcal{V}_\Xi$  **DO**
  - Compute upstream cell  $U = \Psi^{t, t+\tau}V$ ;
  - Compute the total mass  $m_U$  over the upstream cell via (12), (13);
  - Update cell average values  $\bar{u}_V(t + \tau)$  using (14);

**OUTPUT:** Updated cell average values  $\bar{u}_V(t + \tau)$  for all  $V \in \mathcal{V}_\Xi$ .

### 4.3 Implementation of Boundary Conditions

Now let us finally turn to the implementation of boundary conditions. In the above discussion until now, we have considered the special case where the computational domain  $\Omega$  is the whole plane, i.e.,  $\Omega = \mathbb{R}^2$ . However, in specific applications of interest,  $\Omega$  is bounded, and, moreover, boundary conditions are of relevance. Therefore, suppose from now that  $\Omega$  is bounded.

Recall that our proposed scheme works with a partitioning of the plane by using Voronoi cells, given by the Voronoi diagram  $\mathcal{V}_\Xi$  of the current node set  $\Xi$ . In the situation of a bounded domain  $\Omega \subset \mathbb{R}^2$ , we work with restricted Voronoi cells of the form  $\tilde{V} = V \cap \Omega$ , which yields a decomposition of  $\Omega$  by

$$\Omega = \bigcup_{V \in \mathcal{V}_\Xi} \tilde{V}.$$

Note that  $\tilde{V} = V$  for  $V \subset \Omega$ , so that this restriction is only relevant for Voronoi cells which intersect the boundary  $\partial\Omega$  of the domain  $\Omega$ , in which case  $\tilde{V} \neq V$  (see Figure 5 for illustration).

Now the collection  $\{\tilde{U}_\xi\}_{\xi \in \Xi}$  of all upstream cells,  $\tilde{U}_\xi = \Psi^{t, t+\tau}\tilde{V}_\xi$ , yields by their union

$$\Omega^- = \bigcup_{\xi \in \Xi} \tilde{U}_\xi$$

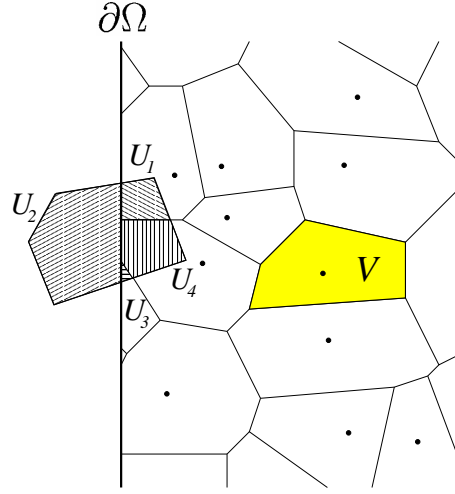


Figure 5: The upstream cell  $U$  of the Voronoi cell  $V$  intersects the boundary  $\partial\Omega$  of the domain  $\Omega$ . A boundary value  $m_{U_2}$  is assigned to the tile  $U_2$ .

an *upstream domain*  $\Omega^- \subset \mathbb{R}^2$ . This gives rise to define the two sets

$$\Omega_{\text{in}} = \Omega^- \setminus \Omega \quad \text{and} \quad \Omega_{\text{out}} = \Omega \setminus \Omega^-.$$

The set  $\Omega_{\text{in}}$  corresponds to a region outside of  $\Omega$  through which mass is advected into  $\Omega$  across its boundary  $\partial\Omega$ , whereas  $\Omega_{\text{out}}$  contains the mass which is advected from  $\Omega$  to the exterior of  $\Omega$  across  $\partial\Omega$ .

More precisely, we can explain this as follows. Note that an upstream cell  $U$  may partly or entirely lie outside the domain  $\Omega$ . Figure 5 shows an upstream cell  $U$ , which intersects the boundary  $\partial\Omega$ . This leads to a tiling of  $U$  as before, but where at least one tile lies entirely outside of the domain  $\Omega$ . In the situation of Figure 5, this is the tile  $U_2$ . Now in order to implement boundary conditions concerning the *incoming flow*, we assign a boundary value to each such tile of  $\Omega_{\text{in}}$ , such as  $U_2 \in \Omega_{\text{in}}$ . This boundary value, say  $m_{U_2}$ , determines the mass which is advected through the tile  $U_2$  into the domain  $\Omega$ .

As regards *outgoing flow*, we remark that the upstream domain  $\Omega^-$  may not cover the entire domain  $\Omega$ , in which case the set  $\Omega_{\text{out}}$  is non-empty. The mass contained in  $\Omega_{\text{out}}$  is advected into regions outside of the domain  $\Omega$ .

## 5 Adaption Rules

One important feature of our advection schemes is its adaptivity. Adaptivity requires the modification of the node set  $\Xi$  after each time step  $t \rightarrow t + \tau$  of the above Algorithm 1. This is in order to be able to balance the two conflicting requirements of good approximation quality and small computational costs. In fact, for the sake of reducing the computational complexity we wish to reduce the size of the node set  $\Xi$ , whereas for the sake of good approximation quality we prefer to increase the density (and thus the size) of the node set  $\Xi$  in  $\Omega$ .

We have combined the conservative advection scheme of this paper with the ideas of the adaption strategy in [1, 2, 3], in order to obtain an *adaptive* and *conservative* semi-Lagrangian advection method. In order to keep this paper widely self-contained, we wish to discuss the basic ideas of the node adaption strategy of the previous work [2] in this section.

### 5.1 Error Indication

An effective strategy for the adaptive modification of the node set  $\Xi$  requires well-motivated refinement and coarsening rules as well as a customized error indicator. We understand the error indicator  $\eta : \Xi \rightarrow [0, \infty)$  as a function of the current node set  $\Xi$  (at time  $t$ ) which serves to assign a *significance* value  $\eta(\xi)$  to each  $\xi \in \Xi$ . The value  $\eta(\xi)$  is required to reflect the local approximation quality of  $\bar{u}_{V_\xi}(t)$  around  $\xi \in \Xi$ . The significances  $\eta(\xi)$ ,  $\xi \in \Xi$ , are then used in order to flag single nodes  $\xi \in \Xi$  as “to be refined” or “to be coarsened” according to the following criteria.

**Definition 1** *Let  $\eta^* = \max_{\xi \in \Xi} \eta(\xi)$ , and let  $\theta_{\text{crs}}, \theta_{\text{ref}}$  be two tolerance values satisfying  $0 < \theta_{\text{crs}} < \theta_{\text{ref}} < 1$ . We say that a node  $\xi \in \Xi$  is **to be refined**, iff  $\eta(\xi) > \theta_{\text{ref}} \cdot \eta^*$ . We say that  $\xi \in \Xi$  is **to be coarsened**, iff  $\eta(\xi) < \theta_{\text{crs}} \cdot \eta^*$ .*

In our numerical experiment, we let  $\theta_{\text{crs}} = 0.05$  and  $\theta_{\text{ref}} = 0.2$  for the relative tolerances. Note that a node  $\xi$  cannot be refined and be coarsened at the same time; in fact, it may neither be refined nor be coarsened.

Now let us turn to the definition of the error indicator  $\eta$ . We follow along the lines of [7], where a local scheme for the detection of discontinuities of a surface from scattered data was developed. Therefore, we let

$$\eta(\xi) = |\bar{u}_{V_\xi}(t) - s(\xi)|, \quad (15)$$

where for a set  $\mathcal{N} \subset \Xi \setminus \{\xi\}$  of neighbouring nodes of  $\xi$ , the *thin plate spline* interpolant  $s \equiv s_{\mathcal{N}}$  in (15), satisfying the interpolation conditions

$s(\nu) = \bar{u}_{V_\nu}(t)$  for all  $\nu \in \mathcal{N}$ , is of the form

$$s = \sum_{\nu \in \mathcal{N}} c_\nu \|\cdot - \nu\|^2 \log(\|\cdot - \nu\|) + p.$$

Here,  $p$  is a linear polynomial in two variables and  $\|\cdot\|$  denotes the Euclidean norm. For more details concerning thin plate spline interpolation, due to Duchon [5], and related interpolation methods, we refer to the recent tutorial [8].

Hence, the thin plate spline interpolant  $s$  in (15) matches current cell average values of  $\bar{u}_{V_\nu}(t)$  in the neighbourhood of the Voronoi cell  $V_\xi$ , but not at  $V_\xi$  itself, i.e., we have  $\bar{u}_{V_\xi}(t) \neq s(\xi)$  in general. Now the error indication  $\eta(\xi)$  for the node  $\xi$  is small whenever the reproduction quality of  $\bar{u}$  by  $s$  around  $\xi$  is good. In contrast, a high value of  $\eta(\xi)$  typically indicates that  $\bar{u}$  is subject to strong variation locally around  $\xi$ . Indeed, this observation relies on available local error estimates for thin plate spline interpolation (see the corresponding discussion on this in [1, 2, 3]). We remark that the error indicator allows us to locate discontinuities of the solution  $u$  quite effectively. This is supported by the numerical results in the following Section 6.

## 5.2 Coarsening and Refinement

In order to balance the approximation quality of the model against the required computational complexity we insert new nodes into regions where the value of  $\eta$  is high (refinement), whereas we remove nodes from  $\Xi$  in regions where the value of  $\eta$  is small (coarsening).

To avoid additional computational overhead and complicated data structures, effective adaption rules are required to be as simple as possible. In particular, these rules ought to be given by *local* operations on the current node set  $\Xi$  and thus on its Voronoi diagram  $\mathcal{V}_\Xi$ . The following coarsening rule is in fact very simple and, in combination with the refinement, it turned out to be very effective as well.

**Coarsening.** A node  $\xi \in \Xi$  is *coarsened* by its removal from the current node set  $\Xi$ . I.e., in this case we let  $\Xi = \Xi \setminus \xi$ , and the Voronoi diagram  $\mathcal{V}_\Xi$  is updated accordingly in order to obtain the modified Voronoi diagram  $\mathcal{V}_{\Xi \setminus \xi}$ .

**Refinement.** A node  $\xi \in \Xi$  is *refined* by the insertion of the  $n$  Voronoi vertices  $\mathbf{v}_1, \dots, \mathbf{v}_n$  of its corresponding Voronoi cell  $V_\xi$ . I.e., in this case we let  $\Xi = \Xi \cup \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ , and the current Voronoi diagram  $\mathcal{V}_\Xi$  is updated accordingly.

## 6 Numerical Results

In this section, the performance of our advection scheme is evaluated by using two numerical experiments. In the first experiment, the accuracy and convergence is analyzed. This is done by considering the test case suggested by Phillips & Williams in [13], which allows us to compare our numerical results with those in [13] directly. In the second experiment, we apply our advection scheme to the *slotted cylinder*, a well-known test case suggested by Zalesak [21]. This illustrates the efficacy of the chosen adaption strategy, and, moreover, it confirms that our method is conservative. The numerical experiments were prepared on a personal computer, model IBM 236623G (Genuintel Pentium(R) 4 1600MHz processor, 256MB physical memory). The algorithms were implemented by using MATLAB, Version 6.5, Release 13.

### 6.1 Experiment 1

According to the numerical experiment suggested in [13], we consider solving the hyperbolic equation (1) on the computational domain  $\Omega = [1, 2]^2 \subset \mathbb{R}^2$ . We let  $a(x) = (x_1, -x_2)$ ,  $x = (x_1, x_2)$ , for the velocity field, as displayed in Figure 6 (a).

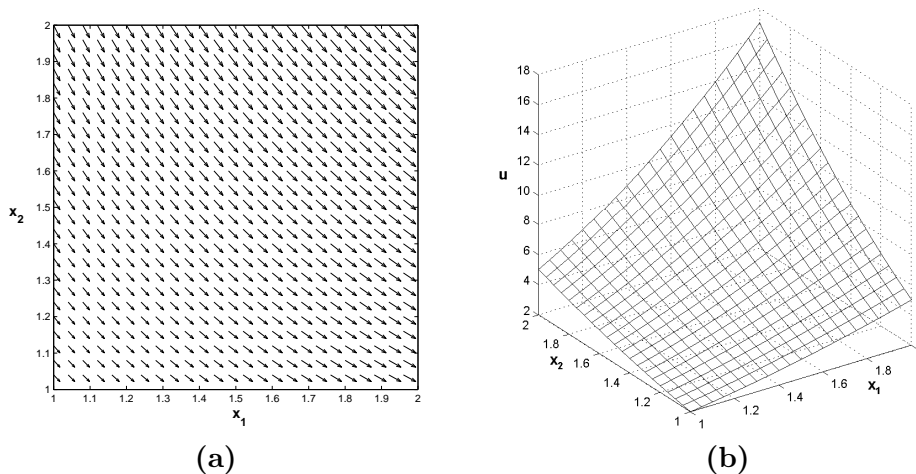


Figure 6: (a) Velocity field and (b) steady state solution  $u$  on  $\Omega$ .

The initial condition suggested in [13] is given by

$$u(0, x) = 0, \quad \text{for } x \in (1, 2] \times [1, 2).$$



For the boundary conditions at the two inflow boundaries, we let

$$\begin{aligned} u(t, x) &= 1 + x_2^2, & \text{for } x_1 = 1, x_2 \in [1, 2], t \geq 0, \\ u(t, x) &= 1 + 4x_1^2, & \text{for } x_2 = 2, x_1 \in [1, 2], t \geq 0. \end{aligned} \quad (16)$$

This condition (16) determines the boundary values of the incoming flow.

We remark that in this model problem, the solution of (1) converges to the *steady state* solution

$$u(x) = 1 + (x_1 x_2)^2, \quad x \in \Omega, \quad (17)$$

as displayed in Figure 6 **(b)**. In order to compare our results with those of Phillips & Williams [13], we proceed as follows.

We first create a nested sequence of four different node sets  $\Xi_h \subset \Omega$ , such that each corresponding Voronoi diagram  $\mathcal{V}_{\Xi_h}$  yields a regular mesh with mesh size  $h$ . The sequence of these four meshes is displayed in Figure 7. In order to make a fair comparison with the results in [13], we keep the node set  $\Xi_h$  fixed throughout each simulation. Moreover, we also let  $\tau = 0.01$  fixed, which leads to convex and nondegenerate upstream cells during the simulation.

According to [13], the simulation terminates, as soon as the stopping criterion

$$\frac{\|\bar{u}_V(t + \tau) - \bar{u}_V(t)\|_\infty}{\tau} \leq 10^{-5}, \quad \text{for all } V \in \mathcal{V}_{\Xi_h}, \quad (18)$$

is satisfied. In this case, the numerical solution has reached the *steady state* approximatively.

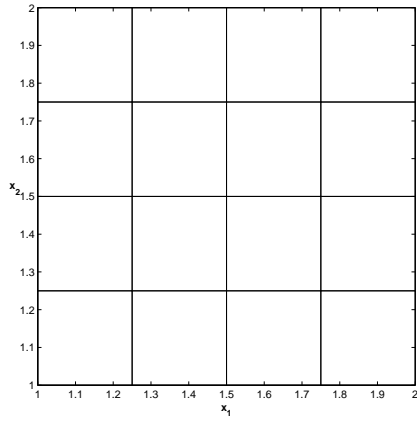
In order to measure accuracy and convergence order, let  $\tilde{u}_h \approx u$  denote this final approximation to the *steady state*  $u$  in (17), on a mesh of width  $h$ . The accuracy of  $\tilde{u}_h$  is measured by using the relative error

$$E_p(h) = \frac{\|u - \tilde{u}_h\|_p}{\|\tilde{u}_h\|_p},$$

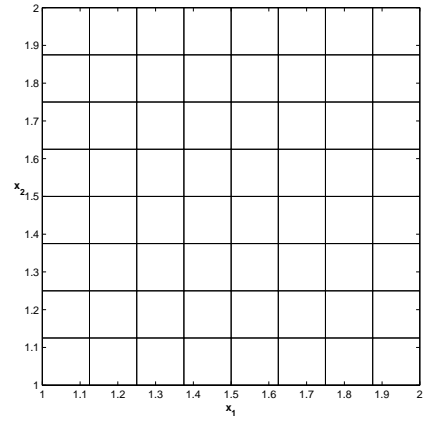
where we let  $p = 1, 2$  or  $p = \infty$  for the corresponding norm. Furthermore, the expression

$$k_p = \frac{\log\left(E_p(h) / E_p(h/2)\right)}{\log(2)},$$

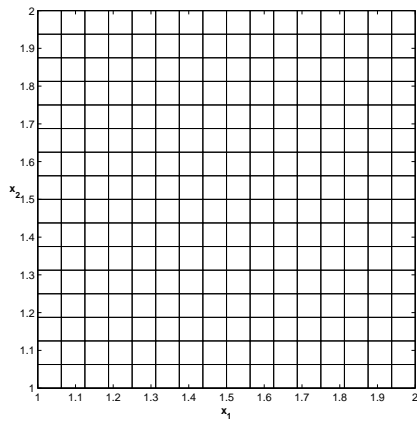
yields an estimate for the convergence order, where  $E_p(h)$  and  $E_p(h/2)$  are the errors (w.r.t. the selected norm  $p = 1, 2, \infty$ ) observed on two subsequent meshes (see Figure 7).



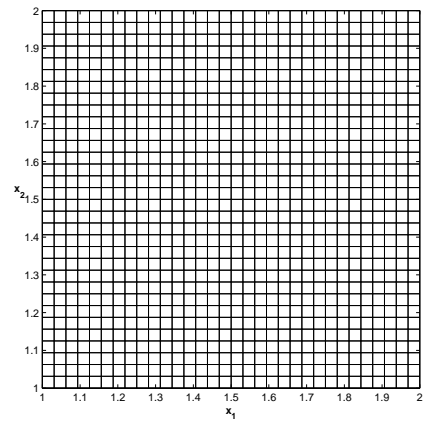
(a)



(b)



(c)



(d)

Figure 7: A sequence of four regular Voronoi diagrams  $\mathcal{V}_{\Xi_h}$  with mesh widths (a)  $h = 0.25$ , (b)  $h = 0.125$ , (c)  $h = 0.0625$ , (d)  $h = 0.03125$ .

h	$E_1(h)$	$k_1$	$E_2(h)$	$k_2$	$E_\infty(h)$	$k_\infty$
0.25	$3.962 \cdot 10^{-2}$	-	$4.653 \cdot 10^{-2}$	-	$7.230 \cdot 10^{-2}$	-
0.125	$9.598 \cdot 10^{-3}$	2.05	$1.178 \cdot 10^{-2}$	1.98	$2.441 \cdot 10^{-2}$	1.57
0.0625	$1.872 \cdot 10^{-3}$	2.36	$2.228 \cdot 10^{-3}$	2.37	$6.471 \cdot 10^{-3}$	1.91
0.03125	$2.789 \cdot 10^{-4}$	2.74	$3.391 \cdot 10^{-4}$	2.75	$1.452 \cdot 10^{-3}$	2.16

Table 1: Convergence results for *regular* Voronoi diagrams  $\mathcal{V}_{\Xi_h}$  (Figure 7).

Table 1 shows the dependence of the error  $E_p(h)$  on the mesh width  $h$  for the norms  $\|\cdot\|_1$ ,  $\|\cdot\|_2$ , and  $\|\cdot\|_\infty$ , together with the corresponding convergence orders  $k_p$ . The results agree very well with those of Phillips & Williams in [13]. In fact, our scheme reaches the expected order of 2 in all three norms.

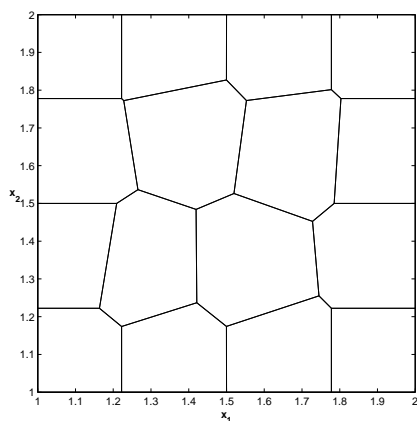
In [13] four different schemes, denoted as **A**, **B**, **C**, and **D**, are introduced. In comparison with the schemes **A**, **C**, and **D**, our method is more accurate. We can explain this as follows. Firstly, in contrast to the schemes **A** and **C** in [13], our scheme is based on a *centered* reconstruction stencil, i.e., on each cell  $V_\xi$  we use all *Voronoi neighbours* of the node  $\xi$  in order to reconstruct the piecewise linear  $u_h$ . Secondly, our slope limiter in (11) is less restrictive than the one used in scheme **D** of [13].

But the scheme **B** in [13] yields more accurate results in terms of approximation errors and convergence orders. This is because centered differences without any slope limiter are used in the scheme **B** of [13]. The absence of the slope limiter, however, often leads to an oscillatory solution  $u_h$ , which is considered to be a severe drawback in many relevant applications.

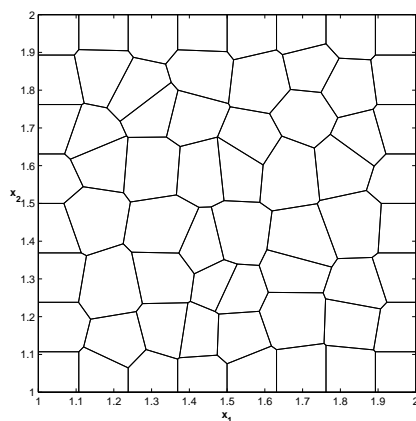
h	$E_1(h)$	$k_1$	$E_2(h)$	$k_2$	$E_\infty(h)$	$k_\infty$
0.25	$1.999 \cdot 10^{-2}$	-	$2.749 \cdot 10^{-2}$	-	$4.734 \cdot 10^{-2}$	-
0.125	$7.308 \cdot 10^{-3}$	1.45	$9.569 \cdot 10^{-3}$	1.52	$1.706 \cdot 10^{-2}$	1.47
0.0625	$2.065 \cdot 10^{-3}$	1.82	$2.483 \cdot 10^{-3}$	1.95	$5.362 \cdot 10^{-3}$	1.67
0.03125	$5.491 \cdot 10^{-4}$	1.91	$6.432 \cdot 10^{-4}$	1.95	$1.469 \cdot 10^{-3}$	1.87

Table 2: Convergence results for *irregular* Voronoi diagrams  $\mathcal{V}_{\Xi_h}$  (Figure 8).

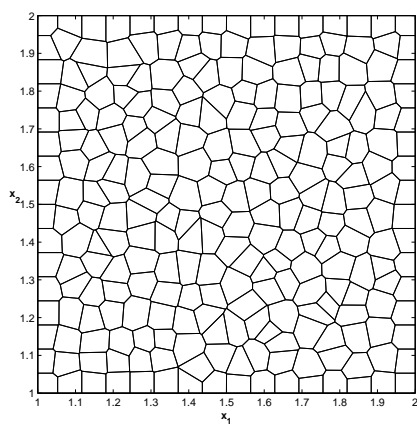
In the following of this experiment, we now investigate the influence of mesh irregularity on the accuracy of our scheme. To this end, we work with a sequence of four irregular meshes  $\mathcal{V}_{\Xi_h}$  (displayed in Figure 8) comprising the same number  $\#\Xi_h$  of cells as in the regular case (see Table 3). In this



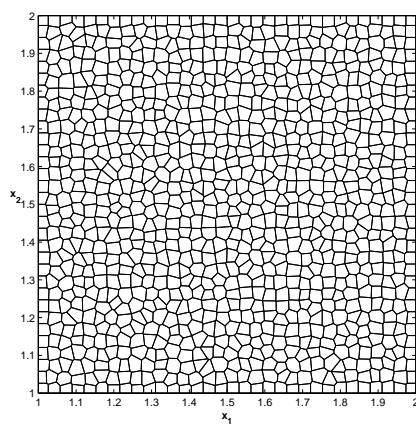
(a)



(b)



(c)



(d)

Figure 8: A sequence of four irregular Voronoi diagrams  $\mathcal{V}_{\Xi_h}$  with mesh widths (a)  $h = 0.25$ , (b)  $h = 0.125$ , (c)  $h = 0.0625$ , (d)  $h = 0.03125$ .

irregular case,  $h$  is considered as a measure of an average mesh width.

Table 2 shows the approximation errors and convergence orders which were obtained by using the same model problem as discussed above, but with using the mesh sequence in Figure 8 instead of the one in Figure 7. Observe that the errors  $E_p(h)$  are of the same magnitude as in the regular case, but the convergence orders  $k_p$  are slightly lower.

We remark that a higher number of time steps, and thus more CPU seconds, are necessary for the irregular meshes in order to reach the steady state  $\tilde{u}_h$  satisfying the stopping criterion (18). This is shown in Table 3. Note also from Table 3 that the advection scheme converges with a fewer number of time steps (for both cases, regular and irregular), as the mesh width  $h$  is reduced. This confirms the utility of semi-Lagrangian schemes. In contrast, Eulerian schemes typically require smaller time steps when the mesh is refined, which in turn leads to an increasing number of necessary time steps.

$h$	$\#\Xi_h$	regular mesh		irregular mesh	
		time steps	CPU seconds	time steps	CPU seconds
0.25	16	189	1.442	202	1.926
0.125	64	129	4.356	157	6.520
0.0625	256	108	12.016	125	17.245
0.03125	1024	95	30.424	99	39.314

Table 3: Time steps and CPU seconds required to reach the *steady state*  $\tilde{u}_h$ .

In conclusion, this numerical experiment shows that our semi-Lagrangian scheme reaches second order accuracy on both structured and unstructured Voronoi diagrams. This provides more flexibility, especially in view of more complicated geometries. In the following experiment, we investigate the conservation properties of our advection scheme. Moreover, we combine the enhanced flexibility of unstructured meshes with local mesh adaption.

## 6.2 Experiment 2

In this experiment we consider the rotating *slotted cylinder*, a popular test case suggested by Zalesak [21]. Here,  $\Omega = [-0.5, 0.5]^2 \subset \mathbb{R}^2$  and the initial condition is given by

$$u(0, x) = \begin{cases} 1 & \text{for } x \in D, \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

where  $D \subset \Omega$  is the slotted disc of radius  $r = 0.15$ , centered at  $(0, 0.25)$  with slot width 0.06 and length 0.22, see Figure 9 (a).

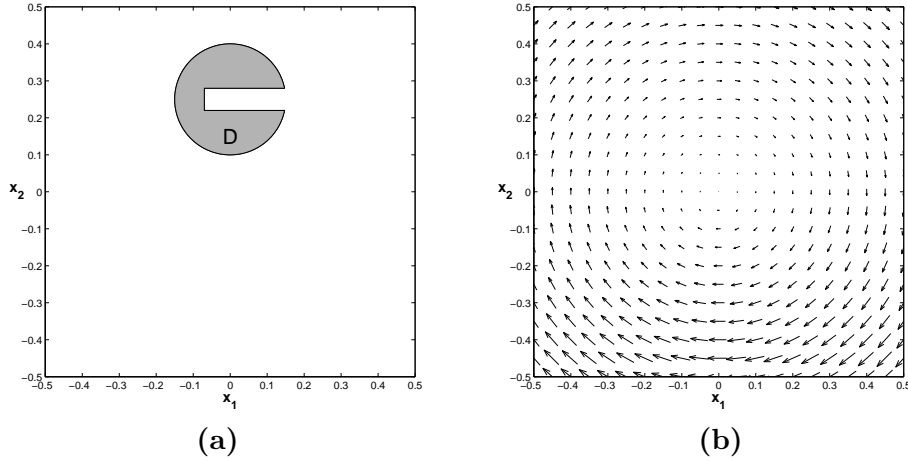


Figure 9: The slotted cylinder. (a) Initial condition and (b) velocity field.

In the original test case of Zalesak, the slotted cylinder is rotated by a steady flow field  $a(x) \sim (x_2, -x_1)$ , where  $x = (x_1, x_2)$ . We decided to replace this velocity field by

$$a(x) = (x_2, -x_1) \begin{cases} \frac{1}{2} \sin(2\varphi(x) - \frac{\pi}{2}) + \frac{3}{2} & \text{for } x_2 < 0, \\ 1 & \text{for } x_2 \geq 0, \end{cases} \quad (20)$$

whose azimuth angle is given by

$$\varphi(x) = \begin{cases} \arctan(-x_2/x_1) & \text{for } x_1 > 0, \\ \arctan(x_1/x_2) + \frac{\pi}{2} & \text{for } x_1 \leq 0. \end{cases}$$

This velocity field rotates the slotted cylinder clockwise with constant angular velocity in the first and second quadrant, whereas the cylinder is accelerated in the fourth quadrant, and decelerated in the third quadrant, see Figure 9 (b). The maximum angular velocity  $\omega = 2$  is attained in the lower half of the coordinate system, namely at the points on the vertical line

$$\{x = (x_1, x_2) : x_1 = 0, x_2 < 0\}.$$

The slotted cylinder is *stretched* when passing through the *acceleration part* of the velocity field in the fourth quadrant, whereas it is *squashed* in the

*deceleration part* of the third quadrant in order to recover its original shape of the initial condition at each full revolution.

Initially, a set  $\Xi \subset \Omega$  of 1500 randomly distributed nodes is chosen. The initial condition (19) is used in order to assign a cell average value  $\bar{u}_V(0)$  in (9) to each Voronoi cell  $V \in \mathcal{V}_\Xi$  of the initial Voronoi diagram  $\mathcal{V}_\Xi$ . The initial nodes are automatically adapted to the discontinuities of the initial condition  $u_0$ , by using the adaption strategy discussed in the previous Section 5, see Figures 10 **(b)**,**(c)**.

At each revolution of the slotted cylinder, the cell average values  $\bar{u}_V$  are decreasing, as soon as the cylinder enters the acceleration part of the velocity field, see Figure 11. This behaviour is due to the mass conservation of the scheme. In contrast to this, in the deceleration part, the cell average values  $\bar{u}_V$  are increasing. Moreover, in this region, the initial shape of slotted cylinder is gradually recovered, see the Figures 11, 12, and 13.

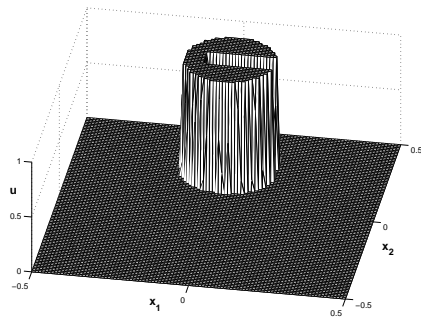
Our simulation of this model problem comprises six full revolutions of the slotted cylinder. During the simulation, we have recorded the number of current nodes, the variation of the time step size  $\tau$ , and the ratio of the first mass moment

$$\text{RFM}(t) = \frac{\sum_{V \in \mathcal{V}_\Xi} \bar{u}_V(t) - m_{\text{in}}(t) + m_{\text{out}}(t)}{\sum_{V \in \mathcal{V}_\Xi} \bar{u}_V(0)}, \quad (21)$$

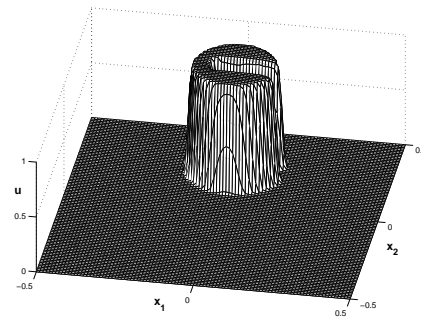
where  $m_{\text{in}}(t)$  is the total mass of the incoming flow and  $m_{\text{out}}(t)$  is the total mass of the outgoing flow, during time  $[0, t]$ , respectively. Our numerical results are reflected by Figure 14. Let us provide a few comments on the three graphs of this figure.

The number of nodes is increasing, whenever the slotted cylinder passes through the accelerating part of the velocity field (20). In this case, the cylinder is stretched, and so more nodes are needed in order to adaptively resolve the elongated edges of the cylinder. We remark that the moderate increase in numbers of nodes at the beginning of the simulation is due to numerical diffusion. In contrast to this, the number of nodes is decreasing, whenever the cylinder enters the decelerating part of the velocity field. In this case, the cylinder is gradually squashed back to its original shape, and so fewer nodes are needed in order to adaptively resolve the cylinder's edges. Altogether, this explains the periodic behaviour of the graph concerning the number of nodes in Figure 14, first row.

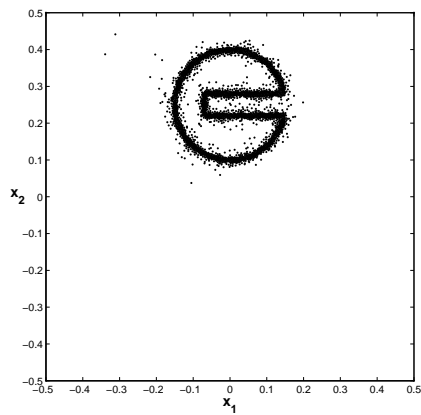
Figure 14 shows also the variation of the time step  $\tau$  in its second row. As mentioned in Subsection 3.2, the time step size is determined, such that



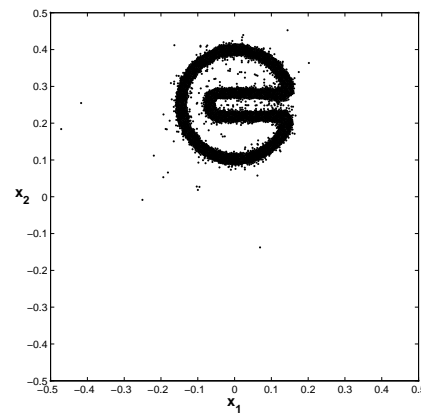
(a)



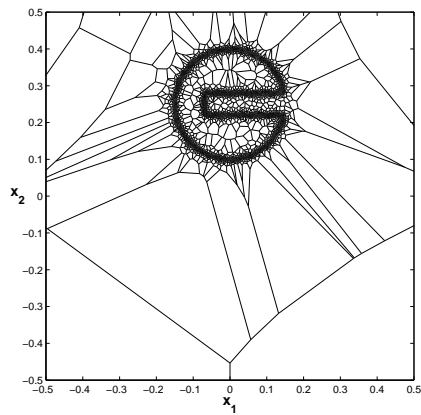
(d)



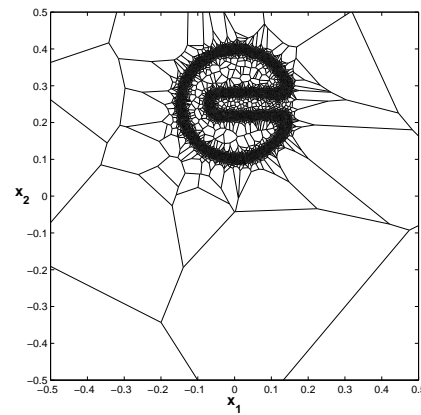
(b)



(e)



(c)



(f)

Figure 10: The slotted cylinder. (a) 3D view, (b) node distribution, and (c) Voronoi diagram of the initial condition (left column), and after six revolutions (right column), (d),(e),(f).



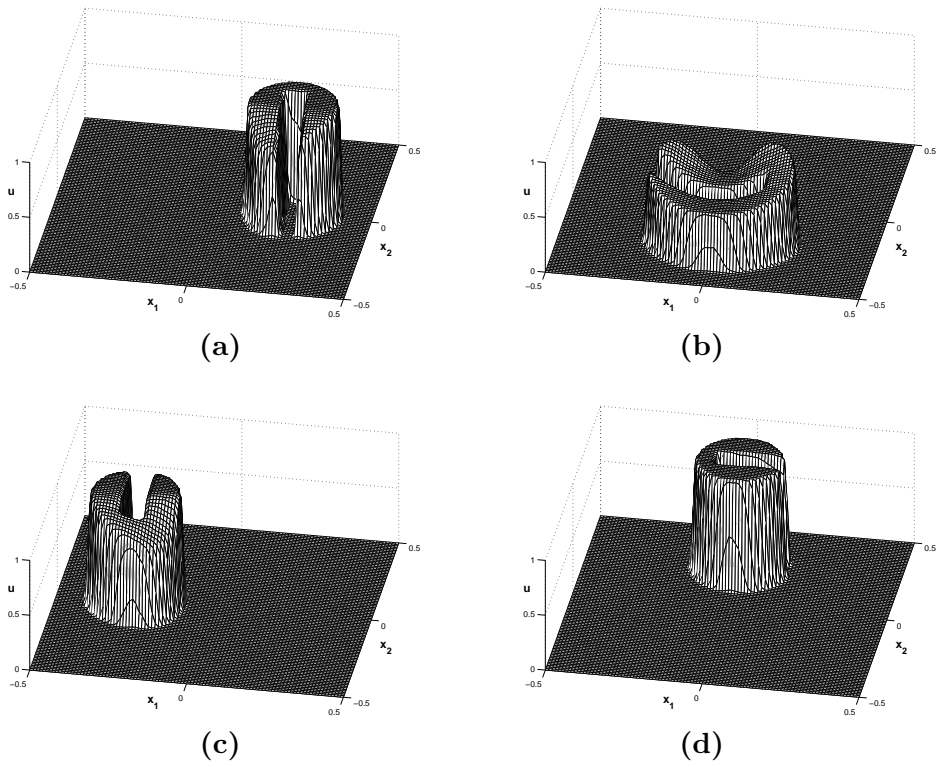


Figure 11: The slotted cylinder. 3D view on the evolution of  $\bar{u}_V(t)$  at four different times, (a)  $t = t_{53}$ ; (b)  $t = t_{70}$ ; (c)  $t = t_{102}$ ; (d)  $t = t_{180}$ , during the first revolution.

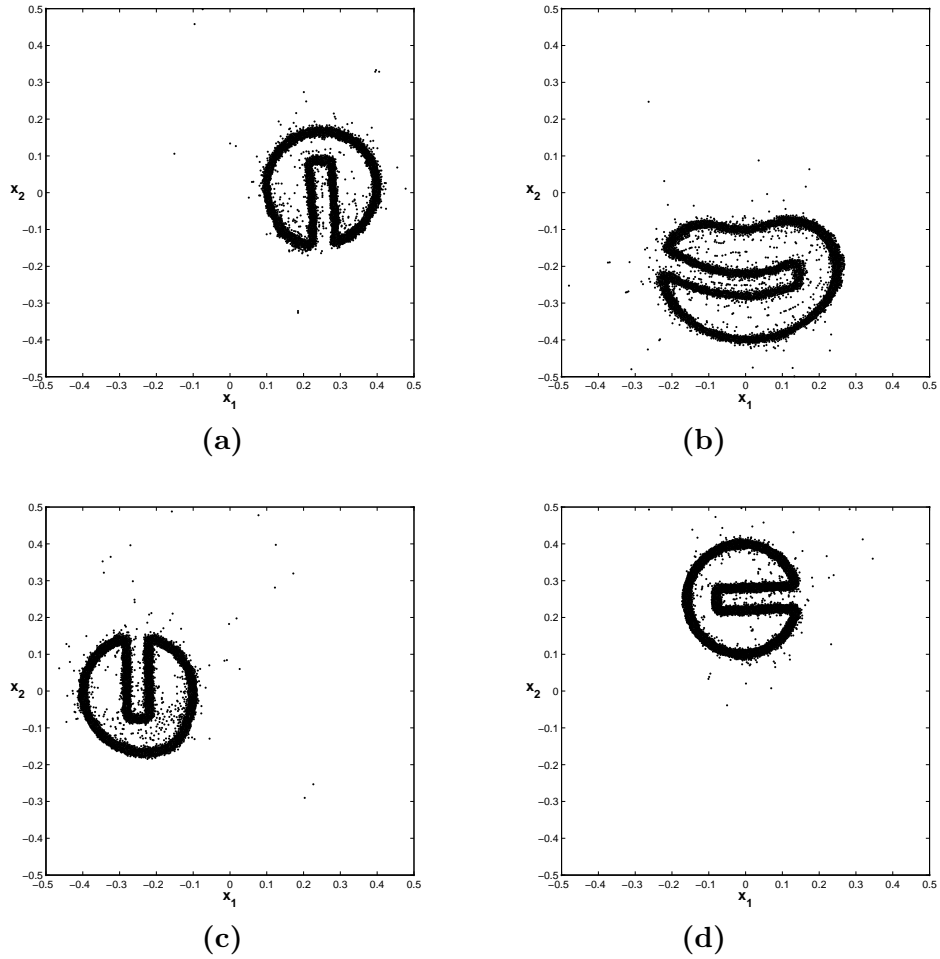
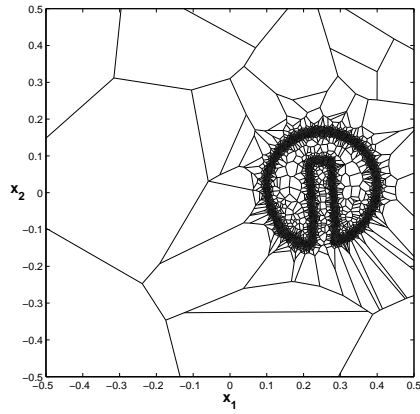
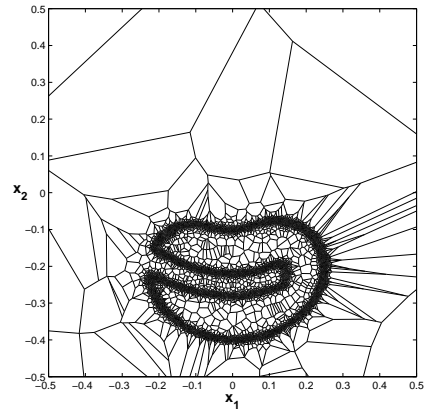


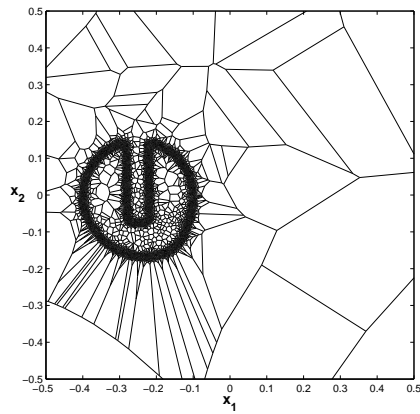
Figure 12: The slotted cylinder. Node distribution during the simulation at four different times, (a)  $t = t_{53}$ ; (b)  $t = t_{70}$ ; (c)  $t = t_{102}$ ; (d)  $t = t_{180}$ , during the first revolution.



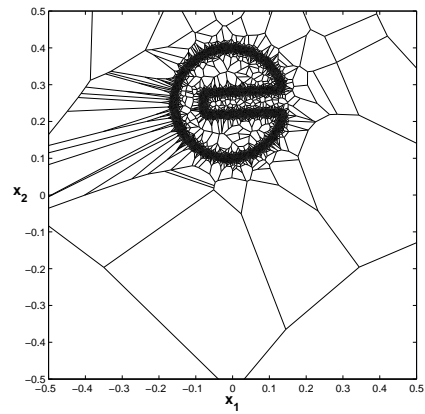
(a)



(b)



(c)



(d)

Figure 13: The slotted cylinder. Voronoi diagram  $\mathcal{V}_{\Xi}$  during the simulation at four different times, (a)  $t = t_{53}$ ; (b)  $t = t_{70}$ ; (c)  $t = t_{102}$ ; (d)  $t = t_{180}$ , during the first revolution.

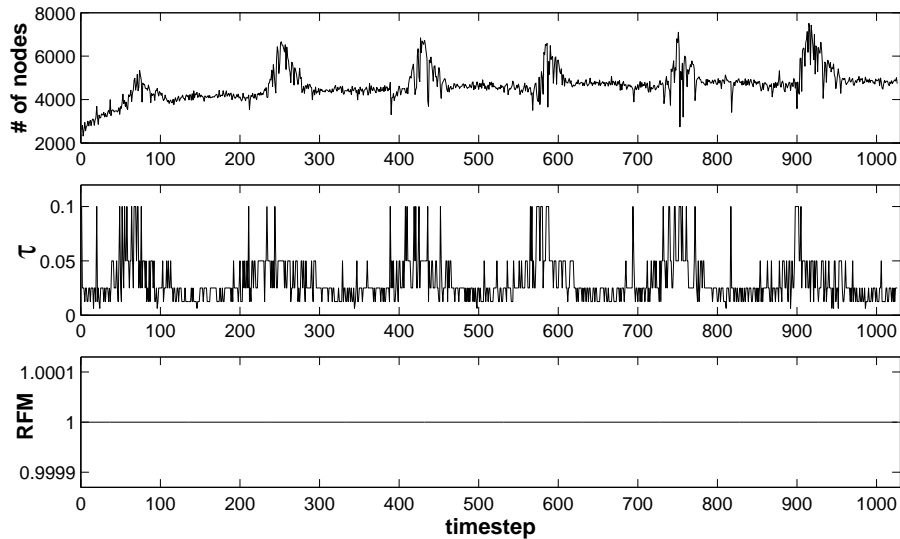


Figure 14: The slotted cylinder. Number of nodes, time step size  $\tau$ , and ratio of the first mass moment (RFM).

all upstream cells are convex. Not surprisingly, this leads to an acceleration (long time steps), whenever the cylinder passes through the accelerating part of the velocity field, whereas a slow down of the cylinder (short time steps) is observed in its decelerating part. Altogether, this explains the correlation between the time step size  $\tau$  and the number of nodes, as shown in Figure 14.

As to mass conservation, we have implemented boundary conditions, as explained in Subsection 4.3. We let  $m_U = 0$  for all tiles  $U \in \Omega_{\text{in}}$ , and so we have  $m_{\text{in}}(t) = 0$ , for all  $t \in [\tau, T]$ , for the total mass of the incoming flow. Figure 14, third row, shows the ratio of the first mass moment,  $\text{RFM}(t)$  in (21). Note that

$$\text{RFM}(t) \equiv 1, \quad \text{for all } t \in [0, T],$$

and so this confirms that our proposed advection scheme is *conservative*.

Figure 10 shows the 3D view of the cell averages  $\bar{u}_V$ , the node distribution, and the Voronoi diagram of the initial condition (19) in the left column, in comparison to the corresponding numerical result after six full revolutions (right column). Observe that the shape of the cylinder is accurately maintained during the simulation, and numerical diffusion is widely suppressed. Finally, the employed TVD slope limiter helps to avoid spurious oscillations of the numerical solution  $u_h$ . In fact, the slope limiter serves to guarantee

the non-negativity of the cell averages throughout the entire simulation, i.e.,  $\bar{u}_V(t) \geq 0$  for all  $t \in I, V \in \mathcal{V}_\Xi$ .

## 7 Conclusion and Future Research

We have proposed a conservative and adaptive advection scheme for linear hyperbolic conservation laws. This semi-Lagrangian method works with finite volumes of an unstructured mesh, given by the Voronoi diagram of a current node set. The nodes, and so the Voronoi diagram, is subject to adaptive modifications during the simulation. These modifications are done according to customized adaption rules, which rely on available error estimates. This adaptive approach helps to reduce the required computational costs while maintaining the accuracy, due to higher resolution around discontinuities in the solution. The implementation of boundary conditions is considered in order to control mass flow into or out of the computational domain. As confirmed in two different numerical experiments, the proposed advection scheme is of second order. The scheme avoids spurious oscillations of the solution by using a TVD slope limiter. Altogether, the numerical results confirm the good performance of the proposed conservative advection scheme.

Finally, we remark that the proposed conservative semi-Lagrangian approach cannot be extended to nonlinear advection equations in a straight forward manner. In fact, the simulation of nonlinear transport processes requires additional sophisticated techniques, especially for modelling shock front propagation. The construction of conservative methods for nonlinear transport equations is currently an active research area. Promising approaches include high order accurate **ADER** schemes (**A**rbitrary high order schemes using high order **DER**ivatives), which were recently introduced in [18, 20]. In view of future research, we intend to combine conservative ADER schemes on unstructured meshes with our powerful and robust node adaption strategy of Section 5.

## Acknowledgment

The authors were partly supported by the European Union within the project NetAGES (Network for Automated Geometry Extraction from Seismic), contract no. IST-1999-29034.

## References

- [1] J. Behrens and A. Iske (2002) Grid-free adaptive semi-Lagrangian advection using radial basis functions. *Comput. Math. Appl.* **43**, 319–327.
- [2] J. Behrens, A. Iske, and S. Pöhn (2001) Effective node adaption for grid-free semi-Lagrangian advection. *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, T. Sonar and I. Thomas (eds.), Logos Verlag, Berlin, 110–119.
- [3] J. Behrens, A. Iske, and M. Käser (2002) Adaptive meshfree method of backward characteristics for nonlinear transport equations. *Mesh-free Methods for Partial Differential Equations*, M. Griebel and M.A. Schweitzer (eds.), Springer-Verlag, Heidelberg, 21–36.
- [4] P. Deuffhard and F. Bornemann (2002) *Scientific Computing with Ordinary Differential Equations*. Springer, New York.
- [5] J. Duchon (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller (eds.), Springer, Berlin, 85–100.
- [6] D.R. Durran (1999) *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer, New York.
- [7] T. Gutzmer and A. Iske (1997) Detection of discontinuities in scattered data approximation. *Numerical Algorithms* **16:2**, 155–170.
- [8] A. Iske (2002) Scattered data modelling using radial basis functions. *Tutorials on Multiresolution in Geometric Modelling*, A. Iske, E. Quak, and M.S. Floater (eds.), Springer-Verlag, Heidelberg, 205–242.
- [9] J.P.R. Laprise and A. Plante (1995) A class of semi-Lagrangian integrated-mass (SLIM) numerical transport algorithms. *Monthly Weather Review* **123**, 553–565.
- [10] R.L. LeVeque (2002) *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, UK.
- [11] K.W. Morton (1996) *Numerical Solution of Convection-Diffusion Problems*. Chapman & Hall, London.
- [12] J. O’Rourke, C.-B. Chien, T. Olson, and D. Naddor (1982) A new linear algorithm for intersecting convex polygons. *Comput. Graph. Image Process.* **19**, 384–391.

- [13] T.N. Phillips and A.J. Williams (2001) Conservative semi-Lagrangian finite volume schemes. *Numer. Meth. Part. Diff. Eq.* **17**, 403–425.
- [14] F.P. Preparata and M.I. Shamos (1985) *Computational Geometry*. Springer, New York.
- [15] A. Priestley (1993) A quasi-conservative version of the semi-Lagrangian advection scheme. *Monthly Weather Review* **121**, 621–629.
- [16] A. Robert (1981) A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean* **19**, 35–46.
- [17] J.S. Scroggs and F.H.M. Semazzi (1995) A conservative semi-Lagrangian method for multidimensional fluid dynamics applications. *Numer. Meth. Part. Diff. Eq.* **11**, 445–452.
- [18] V.A. Titarev and E.F. Toro (2002) ADER: Arbitrary high order Godunov approach. *J. Sci. Comput.* **17**, No.1-4, 609–618.
- [19] E.F. Toro (1999) *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag, Berlin.
- [20] E.F. Toro, R.C. Millington, and L.A.M. Nejad (2001) Towards very high order Godunov schemes. *Godunov methods, Theory and Applications*, E.F. Toro (ed.), Kluwer, New York, 907–940.
- [21] S.T. Zalesak (1979) Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31**, 335–362.

**Authors' addresses:**

Armin Iske (corresponding author)  
 Zentrum Mathematik  
 Technische Universität München  
 D-85747 Garching, GERMANY  
 iske@ma.tum.de  
 Phone: ++49-89-289 17956  
 Fax: ++49-89-289 17985

Martin Käser  
 Zentrum Mathematik  
 Technische Universität München  
 D-85747 Garching, GERMANY  
 kaeser@ma.tum.de