

---

# On the Construction of Mass Conservative and Meshless Adaptive Particle Advection Methods

Armin Iske

Department of Mathematics, University of Hamburg, D-20146 Hamburg, Germany,  
[iske@math.uni-hamburg.de](mailto:iske@math.uni-hamburg.de)

**Summary.** This contribution concerns the construction of meshless Lagrangian particle methods for the numerical simulation of multiscale phenomena in linear transport problems, where mass conservative discretization methods are essentially required. The proposed discretization scheme works with a finite set of unstructured nodes, each corresponding to one flow particle. In this method, corresponding particle average values are maintained during the simulation. The discrete nodes are subject to adaptive modifications, leading to semi-Lagrangian particle simulations, whose adaption rules rely on the insertion (refinement) and removal (coarsening) of the nodes at each time step. The resulting meshless particle method is mass conservative by construction. The required algebraic rules for the downstream particle advection and the local redistribution of the particle masses are developed. Moreover, the implementation of boundary conditions is addressed. The efficacy of the proposed conservative and meshless adaptive particle method is finally shown by using one numerical simulation concerning the slotted cylinder, a popular standard test case for passive advection.

## 1 Introduction

Many physical phenomena in time-dependent evolution processes are modelled by hyperbolic conservation laws, where relevant applications, e.g. from fluid flow simulation, essentially require *conservative* discretization schemes. The *finite volume method* (FVM) is a standard conservative method to construct numerical approximations for solutions of hyperbolic conservation problems. The FVM relies on a suitable partitioning of the computational domain into small cells, the control volumes, where each control volume of the partitioning bears a cell average value. To enforce mass conservation, fluxes are described at the cell interfaces, resulting in a explicit one-step Eulerian discretization scheme. For a comprehensive treatment of finite volume methods, see [10].

Therefore, the classical FVM is *mesh-based* by the domain partitioning, although its construction relies merely on (mesh-independent) geometrical

coefficients: cell volumes, cell surfaces, and cell normal vectors [8]. This observation has recently motivated the construction of a meshless finite volume particle method (FVPM) [6].

Another approach for solving linear hyperbolic equations is given by the *method of characteristics*. This Lagrangian method relies on downstream advection of particles along their streamlines, see [4, 11]. In contrast to Eulerian schemes, Lagrangian particle methods are better suited to construct *meshless* methods, leading to highly flexible discretizations, which are particularly useful for problems with complicated domain geometries, moving boundaries, and large-scale deformations of the solution.

On the down side, Lagrangian methods are usually not conservative, although considerable effort has been made recently in order to construct conservative Lagrangian advection methods, see [9, 12, 13, 14]. In our previous paper [7], a conservative and adaptive semi-Lagrangian method for passive advection has finally been proposed. Yet it seems to be difficult to combine the basic concepts of conservative schemes with those of meshless methods.

This contribution combines three desirable requirements for the construction of effective Lagrangian particle methods in linear hyperbolic conservation problems:

- mass conservation;
- adaptivity;
- mesh-independence.

To this end, algebraic rules for mass conservation are developed. The algebraic rules, given by local mass balance equations, are concerning the mass transfer for the advection step, the particle adaption (i.e., adaptive insertion and removal of particles), and the implementation of boundary conditions. The proposed method can be viewed as a generalization of our previous paper [7], which is conservative and adaptive but not meshless.

The outline of this work is as follows. In the following Section 2, the governing equations for passive advection are briefly reviewed, before some basic ingredients of finite volume particle methods are introduced in Section 3. Then, in Section 4, the mass conservative and meshless advection of *particle averages* is discussed. Section 5 is concerning the adaption of the particles. The implementation of boundary conditions is then explained in Section 6. The efficacy of the proposed conservative and meshless adaptive particle advection is finally illustrated in Section 7 by one numerical simulation concerning the slotted cylinder, a popular standard test case for passive advection.

## 2 Passive Advection and Mass Conservation

The numerical simulation of linear transport processes, *passive advection*, is governed by scalar time-dependent *hyperbolic conservation laws* of the form

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

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*

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

is assumed to be given. The scalar solution  $u : I \times \Omega \rightarrow \mathbb{R}$  of (1) is the density (or concentration) of a physical quantity, which is subject to a conservation law. We consider solving (1) numerically under given initial condition

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

Later in Section 6, boundary conditions are added to the Cauchy problem (1), (2), where the computational domain  $\Omega \subset \mathbb{R}^2$  is assumed to be bounded.

In many relevant applications, *mass conservation*,

$$\frac{d}{dt} \int_{\Omega} u(t, \mathbf{x}) d\mathbf{x} = 0, \quad (3)$$

is an important requirement. The aim of this work is to construct mass conservative particle advection methods satisfying (3).

### 3 Finite Volume Particle Methods

This section discusses selected features of finite volume particle methods, where we combine basic ingredients from the classical Eulerian FVM with recent concepts from particle methods. To this end, let us first briefly recall some basic details of the (mesh-based) FVM, before we turn to a mesh-independent (re)formulation to design meshless particle advection schemes.

#### 3.1 Mesh-Based Formulation

According to the classical FVM discretization, the computational domain  $\Omega$  is first decomposed into a partition  $\mathcal{V} = \{V\}_{V \in \mathcal{V}} \subset \Omega$  of finitely many pairwise disjoint cells  $V$ , *control volumes*, so that

$$\Omega = \bigcup_{V \in \mathcal{V}} V.$$

In order to establish mass conservation, corresponding *cell average values*

$$\bar{u}_V(t) = \frac{1}{|V|} \int_V u(t, \mathbf{x}) d\mathbf{x}, \quad \text{for } V \in \mathcal{V},$$

of the numerical solution  $u \equiv u(t, \mathbf{x})$  are maintained during the simulation. Here, for any  $V \in \mathcal{V}$ ,  $|V|$  denotes the volume of the cell  $V \subset \Omega$ , and so

$$m_V(t) = |V| \cdot \bar{u}_V(t), \quad \text{for } V \in \mathcal{V},$$

is the *total mass* contained in the cell  $V$  at time  $t$ . Therefore,

$$M(t) = \int_{\Omega} u(t, \mathbf{x}) \, d\mathbf{x} = \sum_{V \in \mathcal{V}} \int_V u(t, \mathbf{x}) \, d\mathbf{x} = \sum_{V \in \mathcal{V}} m_V(t)$$

yields the total mass over the domain  $\Omega$  at time  $t$ .

Note that a finite volume discretization of the above form cannot be meshless. Indeed, as soon as the computational domain  $\Omega$  is partitioned into finite control volumes, this essentially requires a mesh.

### 3.2 Meshless Formulation

Now let us work with a more general (generic) formulation of the FVM to obtain a particle advection method, which does not necessarily rely on a mesh, but which is mass conservative by construction. This needs a few conceptual preparations and some reformulations of the FVM, to be explained in the remainder of this subsection.

Meshless finite volume particle methods work with a finite set  $\mathbf{V} = \{\mathbf{v}\}_{\mathbf{v} \in \mathbf{V}} \subset \Omega$  of nodes, each of which corresponds at a time  $t$  to one flow particle. The basic idea of the meshless method, proposed in this work, is to assign, to any node  $\mathbf{v} \in \mathbf{V}$ , a *volume*  $|\mathbf{v}|$  and a *particle average value*

$$\bar{u}_{\mathbf{v}}(t) \approx u(t, \mathbf{v}), \quad \text{for } \mathbf{v} \in \mathbf{V},$$

so that

$$m_{\mathbf{v}}(t) = |\mathbf{v}| \cdot \bar{u}_{\mathbf{v}}(t), \quad \text{for } \mathbf{v} \in \mathbf{V},$$

is the total mass which is, at time  $t$ , attached to the node  $\mathbf{v}$ .

The choice of this particular construction is motivated by the mean value theorem, which states that for an “influence area”  $V \in \Omega$  surrounding node  $\mathbf{v} \in \mathbf{V}$  we have the identity

$$\bar{u}_{\mathbf{v}}(t) \approx \bar{u}_V(t) = u(t, \xi), \quad \text{for some } \xi \in V,$$

which establishes a one-to-one correspondence between the control volumes  $\mathcal{V} = \{V\}_{V \in \mathcal{V}}$  of the previous subsection and the node set  $\mathbf{V} = \{\mathbf{v}\}_{\mathbf{v} \in \mathbf{V}}$  of this subsection.

This way, the meshless concept replaces the *cell average values*  $\bar{u}_V(t)$  and the *cell volumes*  $|V|$  of the mesh-based FVM by *particle average values*  $\bar{u}_{\mathbf{v}}(t)$  and *particle volumes*  $|\mathbf{v}|$ . In this modified concept, the volume  $|\mathbf{v}|$  is for any  $\mathbf{v} \in \mathbf{V}$  regarded as a geometric quantity that should depend on the local geometry of the nodes  $\mathbf{V}$  around  $\mathbf{v}$ , rather than on any connectivities between the nodes. Moreover,  $|\mathbf{v}|$  should reflect the density of the nodes  $\mathbf{V}$  in a local neighbourhood of  $\mathbf{v}$ , and the sum of the node volumes is required to make up the total volume  $|\Omega|$  of the domain  $\Omega$ , i.e.,

$$|\Omega| = \sum_{\mathbf{v} \in \mathbf{V}} |\mathbf{v}|.$$

Therefore, for any node  $\mathbf{v} \in \mathbf{V}$ , its corresponding volume  $|\mathbf{v}|$  can be viewed as the *weight* of  $\mathbf{v}$  in  $\mathbf{V}$ .

Initially, for a suitable node set  $\mathbf{V} = \{\mathbf{v}\}_{\mathbf{v} \in \mathbf{V}} \subset \Omega$ , the particle average values are given by the initial condition (2), so that we let

$$\bar{u}_{\mathbf{v}}(0) = u_0(\mathbf{v}), \quad \text{for all } \mathbf{v} \in \mathbf{V}. \quad (4)$$

This in turn yields the total mass

$$M \equiv M(0) = \sum_{\mathbf{v} \in \mathbf{V}} |\mathbf{v}| \cdot \bar{u}_{\mathbf{v}}(0) = \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(0)$$

at initial time  $t = 0$ .

The aim of the subsequent construction is to solve the Cauchy problem (1),(2) numerically, such that the total mass  $M \equiv M(0)$  is constant during the simulation. To be more precise, at any time step  $t \rightarrow t + \tau$ , with given time step size  $\tau > 0$ , we wish to establish algebraic rules for the required identity  $M(t) = M(t + \tau)$ , i.e.,

$$M(t) = \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(t) = \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(t + \tau) = M(t + \tau), \quad \text{for } t, t + \tau \in I, \quad (5)$$

so that  $M \equiv M(t)$  holds for all  $t \in I$  and all  $\mathbf{V} \equiv \mathbf{V}(t)$ . Later in Section 6, where the implementation of boundary conditions is addressed, we replace the algebraic conditions accordingly, by considering incoming and outgoing flow across the boundary  $\partial\Omega$  of a bounded domain  $\Omega \subset \mathbb{R}^2$ .

## 4 Mass Conservation by Construction

This section explains the key ingredients for mass conservation in meshless particle advection methods. To this end, we wish to compute, at any time step  $t \rightarrow t + \tau$  and for a current node set  $\mathbf{V} \equiv \mathbf{V}(t)$ , updated mass values  $\{m_{\mathbf{v}}(t + \tau) : \mathbf{v} \in \mathbf{V}\}$  from the current masses  $\{m_{\mathbf{v}}(t) : \mathbf{v} \in \mathbf{V}\}$ , which are assumed to be known, where the initial masses  $\{m_{\mathbf{v}}(0) : \mathbf{v} \in \mathbf{V}\}$  are given by (4). According to our mass conservative construction (to be explained below), the updates on the masses are done, such that the identity (5) holds.

To this end, we follow our previous paper [7], where a mesh-based conservative advection method is proposed. The Lagrangian approach in [7] works with downstream mass advection from a finite set  $\mathcal{U}$  of *upstream cells* onto a current set  $\mathcal{V}$  of (downstream) control cells. To be more precise, any *upstream cell*  $U \in \mathcal{U}$  contains, at time  $t$ , those particles, which by traversing along their *streamlines* arrive at time  $t + \tau$  in the corresponding control cell  $V \in \mathcal{V}$ . By this duality relation, there is a one-to-one correspondence between the upstream cells and the control cells.

#### 4.1 Upstream Nodes

In what follows, we wish to establish a similar duality relation between current nodes in  $\mathbf{V}$  and *upstream nodes*, to be collected in a point set  $\mathbf{U}$ . But this requires a few comments concerning the *streamlines* of the particle flow in the context of Lagrangian advection schemes. For any node  $\mathbf{v} \in \mathbf{V}$ , its corresponding *upstream node* is given by  $\mathbf{u} = \mathbf{x}(t)$ , where  $\mathbf{x}$  denotes the unique solution of the *ordinary differential equation* (ODE)

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \mathbf{a}(t, \mathbf{x}), \quad (6)$$

with initial condition  $\mathbf{x}(t + \tau) = \mathbf{v}$ .

Therefore, the upstream node  $\mathbf{u} \equiv \mathbf{u}(\mathbf{v})$  of  $\mathbf{v}$  can be viewed as the unique location of a flow particle at time  $t$ , which by traversing along its trajectory arrives at node  $\mathbf{v}$  at time  $t + \tau$ , see Figure 1. In case of passive advection, governed by the linear transport equation (1), the shape of the particles' flow trajectories, termed *streamlines*, are entirely and uniquely determined by the given velocity field  $\mathbf{a} = \mathbf{a}(t, \mathbf{x})$ . Moreover, the solution  $u$  of (1) is constant along these streamlines, and so the streamlines are the characteristic curves of the hyperbolic equation (1).

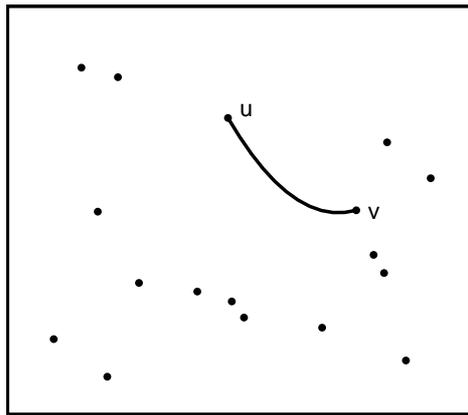


Fig. 1. Upstream node  $\mathbf{u}$  of  $\mathbf{v} \in \mathbf{V}$ .

Adopting some standard notation from dynamical systems [3], we express the upstream node  $\mathbf{u}$  of  $\mathbf{v}$  as

$$\mathbf{u} = \Phi^{t, t+\tau} \mathbf{v}, \quad (7)$$

where  $\Phi^{t, t+\tau} : \Omega \rightarrow \Omega$  denotes the *continuous evolution* of the (backward) flow of (6). An equivalent formulation for (7) 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, it is convenient to express any approximation to upstream node  $\mathbf{u}$  as

$$\Psi^{t,t+\tau} \mathbf{v} \approx \mathbf{u}, \quad \text{for } \mathbf{v} \in \mathbf{V},$$

where  $\Psi^{t,t+\tau} : \Omega \rightarrow \Omega$  is the *discrete evolution* of the (backward) flow, and where the operator  $\Psi^{t,t+\tau}$  is given by any suitable numerical method for solving the above ODE (6). Note that the implementation of  $\Psi^{t,t+\tau}$ , and thus the numerical approximation to  $\mathbf{u}$ , does *not* need a mesh.

For details concerning the implementation of  $\Psi^{t,t+\tau}$  we refer to our previous paper [7], where the duality relation for the pair  $(\mathbf{U}, \mathbf{V})$  is expressed as

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

Moreover, in the mesh-based setting of [7], global mass conservation is obtained by establishing *local mass conservation* through the balance equation

$$\int_{\mathbf{U}} u_h(t, \mathbf{x}) dx = \int_{\mathbf{V}} u_h(t + \tau, \mathbf{x}) dx, \quad \text{for every } \mathbf{V} \in \mathcal{V},$$

where  $u_h$  denotes a piecewise linear approximation to the solution  $u$  of (1),(2).

In the meshless particle advection scheme proposed in this work, a corresponding identity for local mass conservation is accommodated by the mass balance equation

$$m_{\mathbf{u}}(t) = m_{\mathbf{v}}(t + \tau), \quad \text{for every } \mathbf{v} \in \mathbf{V}, \quad (9)$$

where  $m_{\mathbf{u}}(t)$  denotes the portion of mass which is, at time step  $t \rightarrow t + \tau$ , advected from  $\mathbf{u}$  onto  $\mathbf{v}$ . Note that (9) is equivalent to

$$\bar{u}(t + \tau, \mathbf{v}) = \frac{|\mathbf{u}|}{|\mathbf{v}|} \bar{u}(t, \mathbf{u}), \quad \text{for every } \mathbf{v} \in \mathbf{V},$$

where  $|\mathbf{u}|$  is the volume of  $\mathbf{u}$  w.r.t.  $\mathbf{U}$ . But neither the mass  $m_{\mathbf{u}}(t)$  nor the particle average value  $\bar{u}(t, \mathbf{u})$  is known at time  $t$ , unless  $\mathbf{u}$  is a current node in  $\mathbf{V}$ .

## 4.2 Conservative Mass Transfer

For the sake of mass conservation, we determine the masses  $\{m_{\mathbf{u}}(t) : \mathbf{u} \in \mathbf{U}\}$  from the current masses  $\{m_{\mathbf{v}}(t) : \mathbf{v} \in \mathbf{V}\}$ , such that the *local* mass balance equations (9) are satisfied. This then yields the desired *global* mass conservation (5) by construction.

To establish suitable algebraic conditions for local mass conservation (9), we transfer for any node  $\mathbf{v} \in \mathbf{V}$  its entire mass  $m_{\mathbf{v}}(t)$  to upstream nodes  $\{\mathbf{u}\}_{\mathbf{u} \in \mathbf{U}}$  lying in the local neighbourhood of  $\mathbf{v}$ . This is accomplished as follows. Let  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}} \in [0, 1]$  denote the fraction of mass which is being transferred from  $\mathbf{v} \in \mathbf{V}$  to any  $\mathbf{u} \in \mathbf{U}$ , so that

$$m_{\mathbf{v} \rightarrow \mathbf{u}}(t) = \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \cdot m_{\mathbf{v}}(t), \quad \text{for } \mathbf{v} \in \mathbf{V}, \mathbf{u} \in \mathbf{U},$$

is the portion of mass being transferred from  $\mathbf{v}$  to  $\mathbf{u}$ . Hence, if we let  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}} = 0$ , then no mass is transferred from  $\mathbf{v}$  to  $\mathbf{u}$ . This is the case for upstream nodes  $\mathbf{u} \in \mathbf{U}$  which are *not* lying in the local neighbourhood of  $\mathbf{v}$ . In contrast, we select a positive value  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}} > 0$ , whenever  $\mathbf{u} \in \mathbf{U}$  lies in the local neighbourhood of  $\mathbf{v}$ .

Therefore, for any  $\mathbf{u} \in \mathbf{U}$ , the total mass  $m_{\mathbf{u}}$ , which is transferred from the nodes in  $\mathbf{V}$  to  $\mathbf{u}$ , is given by the weighted sum

$$m_{\mathbf{u}}(t) = \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v} \rightarrow \mathbf{u}}(t) = \sum_{\mathbf{v} \in \mathbf{V}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \cdot m_{\mathbf{v}}(t). \quad (10)$$

Now for the sake of mass conservation, we essentially require the property

$$\sum_{\mathbf{u} \in \mathbf{U}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} = 1, \quad \text{for all } \mathbf{v} \in \mathbf{V}, \quad (11)$$

i.e., the non-negative multipliers  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  are required to form a *partition of unity*.

Note that the condition in (11) states that the total mass  $m_{\mathbf{v}}$  of any  $\mathbf{v} \in \mathbf{V}$  is completely transferred to the nodes in  $\mathbf{U}$ . But this immediately establishes the desired identity (5) by the following simple calculations, where we use (9) (10), and (11).

$$\begin{aligned} M(t + \tau) &= \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(t + \tau) \\ &= \sum_{\mathbf{u} \in \mathbf{U}} m_{\mathbf{u}}(t) \\ &= \sum_{\mathbf{u} \in \mathbf{U}} \sum_{\mathbf{v} \in \mathbf{V}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \cdot m_{\mathbf{v}}(t) \\ &= \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(t) \sum_{\mathbf{u} \in \mathbf{U}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \\ &= \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}(t) \\ &= M(t). \end{aligned}$$

### 4.3 Barycentric Coordinates and Voronoi Coefficients

Yet it remains to determine the coefficients  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  satisfying (11), whose construction should depend on the node sets  $\mathbf{V}$  and  $\mathbf{U}$ , and so by the duality (8) also on the solution  $u$  of (1),(2).

One possible option for the construction of the coefficients  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  is to work with *generalized barycentric coordinates* [5]. In this case, non-negative coordinates  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}} \in [0, 1]$  are computed subject to conditions

$$\mathbf{v} = \sum_{\mathbf{u} \in \mathbf{U}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \cdot \mathbf{u} \quad \text{with} \quad \sum_{\mathbf{u} \in \mathbf{U}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} = 1, \quad \text{for } \mathbf{v} \in \mathbf{V}. \quad (12)$$

Note that the linear system for the coordinates  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  in (12) is overdetermined, whenever  $\mathbf{U}$  contains more than three points. In this case, the solution of (12) is not unique. For different options and characterizations of generalized barycentric coordinates, we refer to [5].

In the proposed particle advection method, however, we prefer to work with *Voronoi coefficients*, yielding another possibility for computing the non-negative multipliers  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  subject to constraints (11). As supported by our numerical experiments, this leads to a very flexible mass conservative discretization with superior accuracy, which moreover works particularly well in combination with particle adaption and the implementation of boundary conditions.

In this method, we regard for any node  $\mathbf{v} \in \mathbf{V}$ , the Voronoi tile

$$\text{Vor}_{\mathbf{V}}(\mathbf{v}) = \{\mathbf{x} \in \Omega : \|\mathbf{x} - \mathbf{v}\| \leq \|\mathbf{x} - \mathbf{w}\| \text{ for all } \mathbf{w} \in \mathbf{V}\},$$

of  $\mathbf{v}$ , and we let  $|\mathbf{v}| = |\text{Vor}_{\mathbf{V}}(\mathbf{v})|$  for the volume of  $\mathbf{v}$ . Accordingly, for any  $\mathbf{u} \in \mathbf{U}$ , we let  $|\mathbf{u}| = |\text{Vor}_{\mathbf{U}}(\mathbf{u})|$  for the particle volume of  $\mathbf{u}$ , where

$$\text{Vor}_{\mathbf{U}}(\mathbf{u}) = \{\mathbf{x} \in \Omega : \|\mathbf{x} - \mathbf{u}\| \leq \|\mathbf{x} - \mathbf{w}\| \text{ for all } \mathbf{w} \in \mathbf{U}\}.$$

Then we let

$$\gamma_{\mathbf{v} \rightarrow \mathbf{u}} = \frac{|\text{Vor}_{\mathbf{V}}(\mathbf{v}) \cap \text{Vor}_{\mathbf{U}}(\mathbf{u})|}{|\mathbf{v}|}, \quad \text{for all } \mathbf{u} \in \mathbf{U}, \mathbf{v} \in \mathbf{V}.$$

It is easy to see that for any  $\mathbf{v} \in \mathbf{V}$  the non-negative Voronoi coefficients  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  form a partition of unity, so that (11) holds.

Now the proposed utilization of Voronoi coefficients deserves a rather philosophical comment concerning *meshless* versus *mesh-based* methods. Note that the Voronoi coefficients are based on a very natural geometric concept which mainly relies on the distribution of the nodes in  $\mathbf{U}$  and  $\mathbf{V}$ , rather than being dominated by the topology of a mesh. Although the proposed construction is strictly speaking – through the required Voronoi diagrams of  $\mathbf{U}$  and  $\mathbf{V}$  – not entirely mesh-independent, it does not introduce any restriction to the spatial distribution of the moving particles in  $\mathbf{V}$ . This is particularly important for the construction of flexible adaption rules (as discussed in the following section), where the (unconnected) moving particles are subject to dynamic modifications during the simulation. Finally, it seems to be common practice in meshless methods to work with background Voronoi diagrams (or similar geometric data structures), e.g. to handle geometric queries (such as nearest neighbour search etc.), efficiently.

#### 4.4 Mass Conservative Particle Advection

We close this section by providing the following algorithm, which returns, at any time step  $t \rightarrow t + \tau$ , on given masses  $\{m_{\mathbf{v}}(t) : \mathbf{v} \in \mathbf{V}\}$ , updated mass values  $\{m_{\mathbf{v}}(t + \tau) : \mathbf{v} \in \mathbf{V}\}$  satisfying (5).

**Algorithm 1 (Mass Conservative Particle Advection).****INPUT:** Time step  $\tau$ , node set  $\mathbf{V} \subset \Omega$ , mass values  $\{m_{\mathbf{v}}(t) : \mathbf{v} \in \mathbf{V}\}$ .

- (1) Compute upstream nodes  $\mathbf{U} = \{\mathbf{u} : \mathbf{u} = \Psi^{t, t+\tau} \mathbf{v}\}$  from node set  $\mathbf{V}$ .
- (2) Compute coordinates  $\{\gamma_{\mathbf{v} \rightarrow \mathbf{u}} \in [0, 1] : \mathbf{v} \in \mathbf{V}, \mathbf{u} \in \mathbf{U}\}$ , satisfying (11);

**FOR** each  $\mathbf{v} \in \mathbf{V}$  **DO**

- (3a) Compute total mass  $m_{\mathbf{u}}(t)$  of  $\mathbf{u} = \Psi^{t, t+\tau} \mathbf{v}$  via (10);
- (3b) Let  $m_{\mathbf{v}}(t + \tau) = m_{\mathbf{u}}(t)$ .

**OUTPUT:** Mass values  $\{m_{\mathbf{v}}(t + \tau) : \mathbf{v} \in \mathbf{V}\}$ , satisfying (5).

## 5 Mass Conservative Adaption Rules

In order to balance the two conflicting requirements of good approximation quality and small computational costs, we need to combine the proposed particle advection scheme with a suitable strategy for particle adaption. Adaptivity requires customized rules for the modification of the node set  $\mathbf{V}$  after each time step  $t \rightarrow t + \tau$  of Algorithm 1. Indeed, for the sake of reducing the computational complexity we wish to reduce the size of the node set  $\mathbf{V}$ , whereas for the sake of good approximation quality we prefer to increase the density (and thus the size) of the node set  $\mathbf{V}$  in  $\Omega$ .

This section combines the robust and effective adaption strategy of our previous work [1, 2, 7] with the basic requirements of mass conservation. To this end, we explain how the conservative distribution of mass is accomplished during the adaptive modification of the node set  $\mathbf{V}$ . But we do not intend to explain all details of the utilized node adaption scheme. For the purposes of this work, it is sufficient to say that the node adaption is done by the removal of current nodes from  $\mathbf{V}$ , *coarsening*, and by the insertion of new nodes to  $\mathbf{V}$ , *refinement*. The decision on the node removal and insertion rely on a customized a posteriori error indicator. For details on this, we refer to the previous work [1].

In the remainder of this section, it is explained how these two operations, coarsening and refinement, are accomplished, such that the total mass is conserved.

### 5.1 Coarsening (Node Removal)

A node  $\mathbf{v}^* \in \mathbf{V}$  is *coarsened* by its removal from the current node set  $\mathbf{V}$ , i.e., in this case we let  $\mathbf{V} = \mathbf{V} \setminus \mathbf{v}^*$ . Moreover, the portion of mass  $m_{\mathbf{v}^*}$ , which is currently attached to  $\mathbf{v}^*$  is distributed to remaining nodes in the neighbourhood of  $\mathbf{v}^*$ . In order to further explain this, let  $\mathbf{V}_* \subset \mathbf{V} \setminus \mathbf{v}^*$  denote a set of neighbouring nodes to  $\mathbf{v}^*$ .

For the purpose of mass (re)distribution, we first compute coefficients  $\gamma_{\mathbf{v}^* \rightarrow \mathbf{v}} \in [0, 1]$  satisfying

$$\sum_{\mathbf{v} \in \mathbf{V}_*} \gamma_{\mathbf{v}^* \rightarrow \mathbf{v}} = 1.$$

Next, we update for any node  $\mathbf{v} \in \mathbf{V} \setminus \mathbf{v}^*$  its current mass by letting

$$m_{\mathbf{v}}^* = \begin{cases} m_{\mathbf{v}} + \gamma_{\mathbf{v}^* \rightarrow \mathbf{v}} \cdot m_{\mathbf{v}^*}, & \text{for } \mathbf{v} \in \mathbf{V}_*, \\ m_{\mathbf{v}}, & \text{for } \mathbf{v} \in \mathbf{V} \setminus (\mathbf{V}_* \cup \mathbf{v}^*), \end{cases}$$

so that the mass  $m_{\mathbf{v}}$ , attached to  $\mathbf{v}$ , is updated by  $m_{\mathbf{v}}^*$ .

This leads to a *mass conservative* removal of the node  $\mathbf{v}^*$  from  $\mathbf{V}$  by

$$\begin{aligned} M &= \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}} \\ &= \sum_{\mathbf{v} \in \mathbf{V} \setminus \mathbf{v}^*} m_{\mathbf{v}} + m_{\mathbf{v}^*} \\ &= \sum_{\mathbf{v} \in \mathbf{V} \setminus \mathbf{v}^*} m_{\mathbf{v}} + \sum_{\mathbf{v} \in \mathbf{V}_*} \gamma_{\mathbf{v}^* \rightarrow \mathbf{v}} \cdot m_{\mathbf{v}^*} \\ &= \sum_{\mathbf{v} \in \mathbf{V} \setminus (\mathbf{V}_* \cup \mathbf{v}^*)} m_{\mathbf{v}} + \sum_{\mathbf{v} \in \mathbf{V}_*} (m_{\mathbf{v}} + \gamma_{\mathbf{v}^* \rightarrow \mathbf{v}} \cdot m_{\mathbf{v}^*}) \\ &= \sum_{\mathbf{v} \in \mathbf{V} \setminus \mathbf{v}^*} m_{\mathbf{v}}^*. \end{aligned}$$

## 5.2 Refinement (Node Insertion)

A node  $\mathbf{v}^* \in \mathbf{V}$  is *refined* by the insertion of new nodes  $\hat{\mathbf{V}}$  in the neighbourhood of  $\mathbf{v}^*$ , so that  $\mathbf{V}$  is updated accordingly by letting  $\mathbf{V} = \mathbf{V} \cup \hat{\mathbf{V}}$ .

This modification requires (re)distributing the current masses of neighbouring nodes  $\mathbf{V}_*$  around  $\mathbf{v}^*$  to the new nodes in  $\hat{\mathbf{V}}$ . This is done by working with coefficients  $\gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \in (0, 1)$  satisfying

$$0 < \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} < 1, \quad \text{for } \mathbf{v} \in \mathbf{V}_*,$$

where each  $\gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}}$  yields the fraction of mass being distributed from  $\mathbf{v} \in \mathbf{V}_*$  to  $\hat{\mathbf{v}} \in \hat{\mathbf{V}}$ , so that by  $0 < m_{\mathbf{v}}^* \leq m_{\mathbf{v}}$ , for  $\mathbf{v} \in \mathbf{V}_*$ , the updated mass  $m_{\mathbf{v}}^*$  of  $\mathbf{v} \in \mathbf{V}_*$  is positive. For notational convenience, we let  $\gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} = 0$  for all  $\mathbf{v} \in \mathbf{V} \setminus \mathbf{V}_*$ .

Hence, for any  $\mathbf{v} \in \mathbf{V}$ , its current mass  $m_{\mathbf{v}}$  is updated by letting

$$m_{\mathbf{v}}^* = \left( 1 - \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \right) \cdot m_{\mathbf{v}}, \quad \text{for all } \mathbf{v} \in \mathbf{V}.$$

Moreover, the mass  $m_{\hat{\mathbf{v}}}^*$  of any new node  $\hat{\mathbf{v}} \in \hat{\mathbf{V}}$  (to be inserted) is given by

$$m_{\hat{\mathbf{v}}}^* = \sum_{\mathbf{v} \in \mathbf{V}_*} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \cdot m_{\mathbf{v}} = \sum_{\mathbf{v} \in \mathbf{V}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \cdot m_{\mathbf{v}} \quad \text{for } \hat{\mathbf{v}} \in \hat{\mathbf{V}}.$$

This leads to a *mass conservative* insertion of the new nodes  $\mathbf{V}_*$  by

$$\begin{aligned} M &= \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}} \\ &= \sum_{\mathbf{v} \in \mathbf{V}} \left[ \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \cdot m_{\mathbf{v}} + \left( 1 - \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \right) \cdot m_{\mathbf{v}} \right] \\ &= \sum_{\mathbf{v} \in \mathbf{V}} \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \cdot m_{\mathbf{v}} + \sum_{\mathbf{v} \in \mathbf{V}} \left( 1 - \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \right) \cdot m_{\mathbf{v}} \\ &= \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} \sum_{\mathbf{v} \in \mathbf{V}} \gamma_{\mathbf{v} \rightarrow \hat{\mathbf{v}}} \cdot m_{\mathbf{v}} + \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}^* \\ &= \sum_{\hat{\mathbf{v}} \in \hat{\mathbf{V}}} m_{\hat{\mathbf{v}}}^* + \sum_{\mathbf{v} \in \mathbf{V}} m_{\mathbf{v}}^* \\ &= \sum_{\mathbf{v} \in \mathbf{V} \cup \hat{\mathbf{V}}} m_{\mathbf{v}}^*. \end{aligned}$$

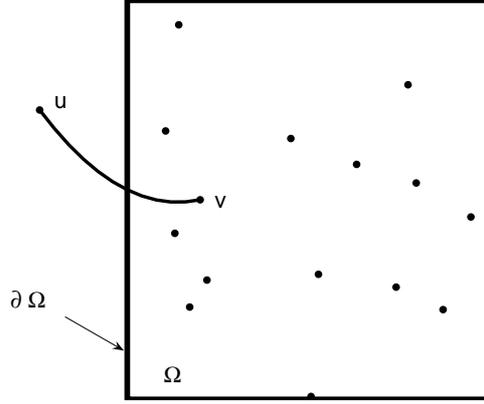
## 6 Implementation of Boundary Conditions

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.,  $\Omega = \mathbb{R}^2$ , so that  $(\mathbf{U} \cup \mathbf{V}) \subset \Omega$  at any time  $t$ . 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 set  $\mathbf{V} = \{\mathbf{v}\}_{\mathbf{v} \in \mathbf{V}}$  of finite nodes, and corresponding upstream nodes  $\mathbf{U} = \{\mathbf{u}\}_{\mathbf{u} \in \mathbf{U}}$ , satisfying  $\mathbf{u} = \Psi^{t, t+\tau} \mathbf{v}$  for all  $\mathbf{v} \in \mathbf{V}$ .

### 6.1 Incoming Flow

The following discussion is relevant, when an upstream node  $\mathbf{u} \in \mathbf{U}$  lies outside the domain  $\Omega$ , i.e.,  $\mathbf{u} \in \mathbf{U} \setminus \Omega$ , see Figure 2 for illustration. In this case, a fraction of mass,  $\gamma_{\partial\Omega \rightarrow \mathbf{u}} \in [0, 1]$ , is advected from  $\mathbf{u}$  to  $\mathbf{v} = \Psi^{t+\tau, t} \mathbf{u} \in \Omega$  across the boundary  $\partial\Omega$ , where the coefficients  $\gamma_{\partial\Omega \rightarrow \mathbf{u}}$  are required to form a partition of unity, i.e.,



**Fig. 2.** Incoming flow from upstream node  $\mathbf{u} \in \mathbf{U} \setminus \Omega$ , to corresponding downstream node  $\mathbf{v} = \Psi^{t+\tau, t} \mathbf{u} \in \Omega$  across the domain boundary  $\partial\Omega$ .

$$\sum_{\mathbf{u} \in \mathbf{U} \setminus \Omega} \gamma_{\partial\Omega \rightarrow \mathbf{u}} = 1.$$

In order to implement boundary conditions concerning the *incoming flow*, we assign a boundary value  $\gamma_{\partial\Omega \rightarrow \mathbf{u}} \cdot m_{\partial\Omega}$  to each  $\mathbf{u} \in \mathbf{U} \setminus \Omega$ , giving the portion of mass which is advected from  $\mathbf{u}$  into  $\Omega$  across the boundary  $\partial\Omega$ . Hence,

$$M_{\text{in}} = \sum_{\mathbf{u} \in \mathbf{U} \setminus \Omega} \gamma_{\partial\Omega \rightarrow \mathbf{u}} \cdot m_{\partial\Omega} \equiv m_{\partial\Omega}$$

is the total amount of incoming mass across the boundary  $\partial\Omega$ .

Moreover, for any  $\mathbf{u} \in \mathbf{U} \setminus \Omega$ , portions of mass from current nodes  $\mathbf{v} \in \mathbf{V}$  may also be distributed to  $\mathbf{u}$ . In this case, we obtain

$$m_{\mathbf{u}} = \gamma_{\partial\Omega \rightarrow \mathbf{u}} \cdot m_{\partial\Omega} + \sum_{\mathbf{v} \in \mathbf{V}} \gamma_{\mathbf{v} \rightarrow \mathbf{u}} \cdot m_{\mathbf{v}}, \quad \text{for } \mathbf{u} \in \mathbf{U} \setminus \Omega,$$

for the total mass which is assigned to the upstream node  $\mathbf{u}$ .

## 6.2 Outgoing Flow

Let us regard the situation, where for  $\mathbf{v} \in \mathbf{V}$  all its coordinates  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}}$  vanish, i.e.,  $\gamma_{\mathbf{v} \rightarrow \mathbf{u}} = 0$  for all  $\mathbf{u} \in \mathbf{U}$ . In this case, *no* mass is distributed from  $\mathbf{v}$  to any point in  $\mathbf{U}$ . The amount of mass  $m_{\mathbf{v}}$ , attached to  $\mathbf{v}$ , is then being advected from  $\mathbf{v} \in \Omega$  to the exterior of  $\Omega$  across the domain boundary  $\partial\Omega$ .

In consequence, the total outgoing mass is given by

$$M_{\text{out}} = \sum_{\mathbf{v} \in \mathbf{V}_{\text{out}}} m_{\mathbf{v}},$$

where we let

$$\mathbf{V}_{\text{out}} = \{\mathbf{v} \in \mathbf{V} : \gamma_{\mathbf{v} \rightarrow \mathbf{u}} = 0 \text{ for all } \mathbf{u} \in \mathbf{U}\} \subset \mathbf{V}$$

for the set of nodes whose masses are advected across  $\partial\Omega$  to the exterior of  $\Omega$ .

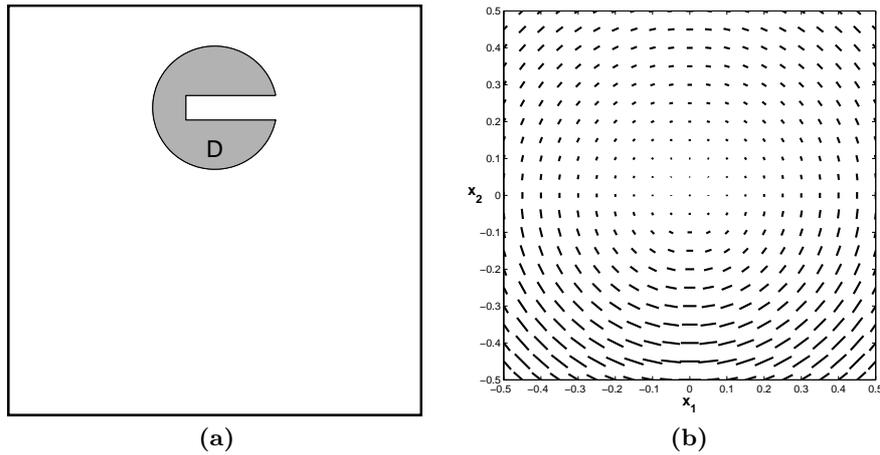
## 7 The Slotted Cylinder — A Test for Passive Advection

In this section, the performance of our advection scheme is evaluated by using one numerical experiment. In this experiment, taken from [7, Section 5], we consider the rotating *slotted cylinder*, a popular test case suggested by Zalesak [15].

Here,  $\Omega = [-0.5, 0.5]^2 \subset \mathbb{R}^2$  and the initial condition is given by

$$u(0, \mathbf{x}) = \begin{cases} 1, & \text{for } \mathbf{x} \in D, \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

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 3 (a).



**Fig. 3.** 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  $\mathbf{a}(\mathbf{x}) \sim (x_2, -x_1)$ , where  $\mathbf{x} = (x_1, x_2)$ . We decided to replace the velocity field in [15] by the somewhat more complicated velocity field

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

whose azimuth angle is given by

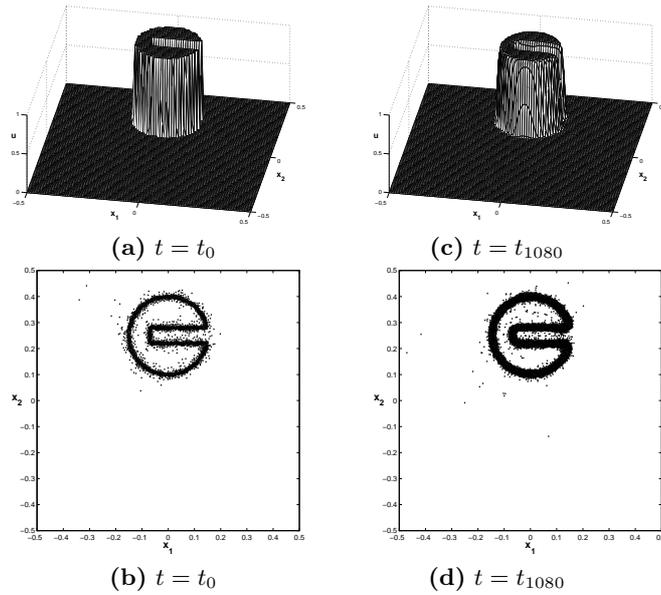
$$\varphi(\mathbf{x}) = \begin{cases} \arctan(-x_2/x_1), & \text{for } x_1 > 0, \\ \arctan(x_1/x_2) + \frac{\pi}{2}, & \text{for } x_1 \leq 0. \end{cases}$$

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 5. The maximum angular velocity  $\omega = 2$  is attained in the lower half of the coordinate system, namely at the points on the vertical line

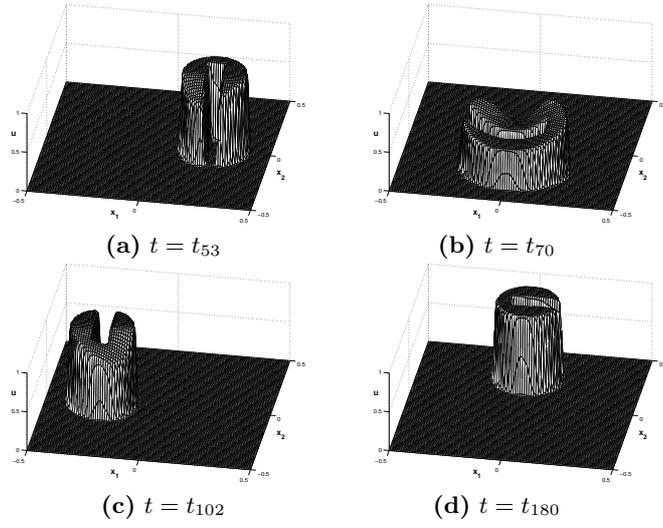
$$\{\mathbf{x} = (x_1, x_2) : x_1 = 0, x_2 < 0\}.$$

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.

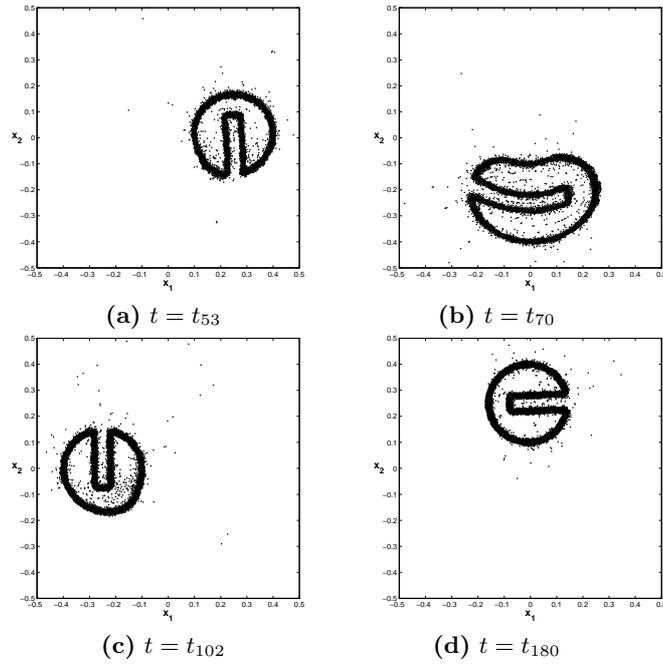
Initially, a set  $\mathbf{V} \subset \Omega$  of 1500 randomly distributed nodes is chosen. The initial condition (13) is used in order to assign a particle average value  $\bar{u}_{\mathbf{v}}(0)$  in (4) to each node  $\mathbf{v} \in \mathbf{V}$  at time  $t = t_0$ . The nodes in  $\mathbf{V}$  are automatically adapted to the discontinuities of the initial condition  $u_0$ , see Figure 4 (b).



**Fig. 4.** The slotted cylinder. (a) 3D view, (b) node distribution, of the initial condition (left column), and after six revolutions (right column), (c),(d).



**Fig. 5.** The slotted cylinder. 3D view on  $\bar{u}_v(t)$  during first revolution at four times.



**Fig. 6.** The slotted cylinder. Node distribution during first revolution at four times.

At each revolution of the slotted cylinder, the particle average values  $\bar{u}_v$  are decreasing, as soon as the cylinder enters the acceleration part of the velocity field, see Figure 5. This behaviour is due to the mass conservation of the scheme. In contrast to this, in the deceleration part, the particle average values  $\bar{u}_v$  are increasing. Moreover, in this region, the initial shape of slotted cylinder is gradually recovered, see Figure 5 (b)-(d).

Our simulation of this model problem comprises six full revolutions of the slotted cylinder. Figure 4 (a) shows the 3D view of the particle averages  $\bar{u}_v$ , and Figure 4 (b) shows the node distribution for the initial condition (13). In comparison, Figure 4 (c)-(d) shows to the corresponding numerical result after six full revolutions. Observe that the shape of the cylinder is accurately maintained during the simulation, and numerical diffusion is widely suppressed. For more details concerning the model problem and the discussion of the numerical results, we refer to [7].

## Acknowledgement

The fruitful joint collaboration with Martin Käser in previous work on related particle methods is greatly appreciated.

## References

1. 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, Berlin, 110–119.
2. 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.
3. P. Deuffhard and F. Bornemann (2002) *Scientific Computing with Ordinary Differential Equations*. Springer, New York.
4. D.R. Durran (1999) *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer, New York.
5. M.S. Floater, K. Hormann, and G. Kós (2006) A general construction of barycentric coordinates over convex polygons. *Advances in Comp. Math.* **24**, 311–331.
6. D. Hietel, K. Steiner, and J. Struckmeier: A finite-volume particle method for compressible flows. *Math. Mod. Meth. Appl. Sci.* **10**(9), 2000, 1363–1382.
7. A. Iske and M. Käser (2004) Conservative semi-Lagrangian advection on adaptive unstructured meshes. *Numerical Methods for Partial Differential Equations* **20**, 388–411.
8. M. Junk: Do finite volume methods need a mesh? In: *Meshfree Methods for Partial Differential Equations*, M. Griebel, M.A. Schweitzer (eds.), Springer, 2003, 223–238.
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. T.N. Phillips and A. J. Williams (2001) Conservative semi-Lagrangian finite volume schemes. *Numer. Meth. Part. Diff. Eq.* **17**, 403–425.
13. A. Priestley (1993) A quasi-conservative version of the semi-Lagrangian advection scheme. *Monthly Weather Review* **121**, 621–629.
14. 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.
15. S. T. Zalesak (1979) Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31**, 335–362.