# Do Finite Volume Methods Need a Mesh?

Michael Junk

Fachbereich Mathematik, Universität Kaiserslautern, 67663 Kaiserslautern, Germany

**Abstract.** In this article, finite volume discretizations of hyperbolic conservation laws are considered, where the usual triangulation is replaced by a partition of unity on the computational domain. In some sense, the finite volumes in this approach are not disjoint but are overlapping with their neighbors. This property can be useful in problems with time dependent geometries: while the movement of grid nodes can have unpleasant effects on the grid topology, the meshfree partition of unity approach is more flexible since the finite volumes can arbitrarily move ontop of each other. In the presented approach, the algorithms of classical and meshfree finite volume method are identical – only the geometrical coefficients (cell volumes, cell surfaces, cell normal vectors) have to be defined differently. We will discuss two such definitions which satisfy certain stability conditions.

## 1 Introduction

The finite volume method (FVM) is a standard approach to construct approximate solutions of hyperbolic conservation laws [4,8]. The basic idea is to split the computational domain into small cells - the finite volumes - and to enforce conservation by prescribing fluxes at the cell interfaces: if a certain amount of the conserved quantity leaves cell $C_i$ across a common boundary $\Gamma_{ij}$ with cell $C_j$, it has to reappear in $C_j$. In this way, the evolution of the conserved quantities can be approximated if the fluxes are suitable approximations of the fluxes given by the conservation laws.

At this level, the underlying mesh seems to be very important. However, if one looks at the finite volume method from a more abstract point of view, it appears as a system of ODEs with the following ingredients: a numerical flux function $g$ and parameters $V_i$ and $\boldsymbol{\beta}_{ij}$. Here, $V_i$ has the interpretation as volume of cell $C_i$, $|\boldsymbol{\beta}_{ij}|$ is the surface area of the interface $\Gamma_{ij}$, and $\boldsymbol{\beta}_{ij}/|\boldsymbol{\beta}_{ij}|$ is the corresponding normal vector, pointing from cell $C_i$ to cell $C_j$. Now the question, whether finite volume methods need a mesh, can be reformulated mathematically: under which conditions on $V_i$ and $\boldsymbol{\beta}_{ij}$ does the finite volume method produce reliable approximations to solutions of the conservation law? Is it really necessary that $V_i$ and $\boldsymbol{\beta}_{ij}$ are constructed from a mesh, or do they just have to satisfy some algebraic relations?

This interesting question naturally arises in connection with the finite volume particle method (FVPM) which has recently been proposed in [5]. At the core of this method is a partition of unity on the computational domain where the partition functions are used as test functions in the weak formulation of the conservation law. As a result, a system of ODEs is obtained which looks very much like a finite volume method but the parameters $V_i$ and $\boldsymbol{\beta}_{ij}$ are given by integrals over the partition functions and *not* as quantities derived from a mesh.

Nevertheless, the obtained geometric parameters satisfy all assumptions which are needed in the convergence proof of classical finite volume methods and numerical experiments show that FVPM yields reliable results. Hence, we can say that reasonable geometric parameters $V_i$, $\boldsymbol{\beta}_{ij}$ in finite volume methods can be generated without underlying mesh.

In the article, the requirements on the geometric parameters $V_i$ and $\boldsymbol{\beta}_{ij}$ are explained and numerical examples are presented which show the practical relevance of these conditions. Moreover, we show that FVPM can easily be coupled with classical finite volume methods.

Apart from FVPM we introduce a quite similar method called PUMESH. Starting from a partition of unity on a $d$-dimensional domain, we build an associated mesh on a $d+1$ dimensional cylinder with the original geometry as cross section. On this grid, a classical finite volume ansatz is used which gives rise to a scheme where the additional dimension is no longer visible and where the geometric coefficients are defined by integrals of the partition functions. Also in this case, the geometric parameters satisfy the required stability conditions.

## 2 The Finite Volume Method

Let us start by recalling the finite volume discretization of conservation laws (for a detailed discussion, we refer to [4,8]). As example, we consider the problem to find $u : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} + \mathrm{div}_{\boldsymbol{x}} \boldsymbol{F}(u) = 0, \qquad u(0,\boldsymbol{x}) = u^0(\boldsymbol{x}). \tag{1}$$

In order to construct an approximate solution for (1), we split the domain $\mathbb{R}^d$ into small, disjoint, polyhedral volumes $C_1, C_2, \ldots$ such that $\Omega = \cup_i C_i$. Integrating (1) over such a cell and using the divergence theorem, we obtain

$$\frac{d}{dt}\int_{C_i} u \, d\boldsymbol{x} + \sum_j \int_{\Gamma_{ij}} \boldsymbol{F}(u)\cdot \boldsymbol{n}_{ij}\, dS = 0. \tag{2}$$

Here, $\Gamma_{ij}$ is the common cell interface between cell $i$ and cell $j$ (see Fig. 1) and $\boldsymbol{n}_{ij}$ is the associated outer normal vector. If $C_i$ and $C_j$ are not adjacent, we define $\Gamma_{ij} = \emptyset$ and $\boldsymbol{n}_{ij}$ can be arbitrary. Introducing the cell volumes $V_i$ and the cell averages $u_i(t)$

$$V_i = \int_{C_i} d\boldsymbol{x}, \qquad u_i = \frac{1}{V_i}\int_{C_i} u\, d\boldsymbol{x}, \tag{3}$$

the first term in (2) can be written as $d/dt(u_i V_i)$ which we approximate, for simplicity, with an explicit Euler step

$$\frac{d}{dt}\int_{C_i} u\, d\boldsymbol{x}|_{t=n\Delta t} = \frac{d}{dt}(u_i V_i)|_{t=n\Delta t} \approx V_i \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

where $u_i^n = u_i(t_n)$, $t_n = n\Delta t$.

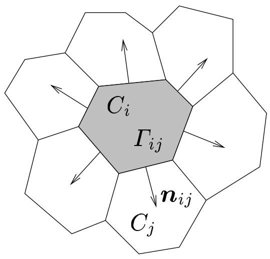


**Fig. 1.** Control volume $C_i$ with interfaces $\Gamma_{ij}$ and outer normals $\boldsymbol{n}_{ij}$

If we take the cell averages $u_i^n$ as our unknowns in the numerical scheme, we also have to approximate the flux integrals in (2) through these quantities. One approach is to replace the function $u$ by the piece-wise constant function

$$\tilde{u}(t, \boldsymbol{x}) = \sum_i u_i^n \mathbf{1}_{C_i}(\boldsymbol{x}) \mathbf{1}_{[t_n, t_{n+1})}(t), \qquad \boldsymbol{x} \in \mathbb{R}^d, t \in [0, T] \tag{4}$$

where $\mathbf{1}_A$ denotes the characteristic function of a set $A$. Doing this, we find the values $u_i^n$ and $u_j^n$ on the two sides of $\Gamma_{ij}$ and the flux throught the interface should be essentially determined by these values

$$\int_{\Gamma_{ij}} \boldsymbol{F}(u) \cdot \boldsymbol{n}_{ij}\, dS|_{t=n\Delta t} \approx |\Gamma_{ij}| g(u_i^n, u_j^n, \boldsymbol{n}_{ij}), \qquad |\Gamma_{ij}| = \int_{\Gamma_{ij}} dS.$$

The function $g$ is called *numerical flux function*. We remark that $g(v, w, \boldsymbol{n})$ can be constructed by solving problem (1) (exactly or approximately) with a particular initial value $u^0$ consisting of two constant values $v, w$ which are separated by an infinite plane with normal $\boldsymbol{n}$ (Riemann problem). For details, we again refer to [4,8]. Combining our approximations, we end up with the classical finite volume method

$$u_i^{n+1} V_i = u_i^n V_i - \Delta t \sum_j |\Gamma_{ij}| g(u_i^n, u_j^n, \boldsymbol{n}_{ij}), \qquad u_i^0 = \frac{1}{V_i}\int_{C_i} u^0\, d\boldsymbol{x} \tag{5}$$

Iterating (5), we can construct the piecewise constant function $\tilde{u}$, and, under suitable assumptions, one can show that for finer and finer discretizations of space and time, the approximation $\tilde{u}$ converges to the entropy solution of (1) (see, for example, [8,2,11] and the more recent references [1,12] for $(t, \boldsymbol{x})$ dependent fluxes and domains with boundary). The assumptions required for the convergence proof can be split into conditions on the flux function $g$ (like consistency, conservativity, Lipschitz continuity and monotonicity) and into conditions on the underlying mesh. In the following, we are particularly interested in these geometrical conditions which we formulate in terms of the quantities $V_i$ and $\boldsymbol{\beta}_{ij} = |\Gamma_{ij}|\boldsymbol{n}_{ij}$.

A first requirement is that a cell should not have too many neighbors, i.e. the number of indices in $\{j : \boldsymbol{\beta}_{ij} \neq \boldsymbol{0}\}$ should be uniformly bounded for every index $i$

$$\big|\{j : \boldsymbol{\beta}_{ij} \neq \boldsymbol{0}\}\big| \leq K, \qquad \forall i. \tag{6}$$

Moreover, if $h$ is the largest cell diameter then volumes and surface areas should satisfy

$$V_i \geq \alpha h^d, \qquad |\boldsymbol{\beta}_{ij}| \leq C h^{d-1}, \qquad \forall i \tag{7}$$

with $h$-independent constants $\alpha$ and $C$. Geometrically, (7) prevents narrow cells with very small volumes or very large surfaces. Apart from the rather technical conditions (6) and (7), two algebraic conditions on the coefficients are needed

$$\boldsymbol{\beta}_{ij} = -\boldsymbol{\beta}_{ji}, \qquad \forall i, j \tag{8}$$

$$\sum_j \boldsymbol{\beta}_{ij} = \boldsymbol{0}, \qquad \forall i \tag{9}$$

Note that (8) is a direct consequence of the fact that $\Gamma_{ij} = \Gamma_{ji}$ and $\boldsymbol{n}_{ij} = -\boldsymbol{n}_{ji}$ in connection with $\boldsymbol{\beta}_{ij} = |\Gamma_{ij}|\boldsymbol{n}_{ij}$. The second property (9) follows from the divergence theorem. Introducing $\boldsymbol{b} = \sum_j \boldsymbol{\beta}_{ij}$, we have

$$|\boldsymbol{b}|^2 = \sum_j \boldsymbol{\beta}_{ij} \cdot \boldsymbol{b} = \sum_j \int_{\Gamma_{ij}} \boldsymbol{n}_{ij} \cdot \boldsymbol{b}\, dS = \int_{\partial C_i} \boldsymbol{n} \cdot \boldsymbol{b}\, dS = \int_{C_i} \mathrm{div} \boldsymbol{b}\, d\boldsymbol{x} = 0.$$

We remark that the finite volume method applied to domains with boundaries contains extra terms due to boundary fluxes. In fact, if we integrate (1) over a cell $C_i$ whose boundary intersects $\partial\Omega$, then

$$\int_{\partial C_i} \boldsymbol{F}(u) \cdot \boldsymbol{n}\, dS \approx \sum_j g(u_i, u_j, \boldsymbol{n}_{ij}) + \int_{\partial\Omega \cap \partial C_i} \boldsymbol{F}(u) \cdot \boldsymbol{n}\, dS.$$

The remaining boundary integral has to be approximated using the boundary conditions and by extrapolating $u$ from the computational domain to the boundary where no information on $u$ is available. Also, the condition (9) changes in boundary cells. With the same argument as before, one can show that

$$\sum_j \boldsymbol{\beta}_{ij} = -\int_{\partial\Omega\cap\partial C_i} \boldsymbol{n}\, dS \tag{10}$$

where the right hand side obviously vanishes for all interior cells.

## 3 The Finite Volume Particle Method

The finite volume particle method (FVPM) has been developed a few years ago in an attempt to combine features of SPH (Smoothed Particle Hydrodynamics) with finite volume methods [5]. A similar approach has also been proposed in [3] for a particular equation from petroleum reservoir simulation.

To explain the idea, let us assume that the computational domain $\mathbb{R}^d$ is covered with the supports of "smoothed particles", i.e.

$$\bigcup_i \mathrm{supp} W_i = \mathbb{R}^d, \qquad W_i(\boldsymbol{x}) = W\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right).$$

Here, $W$ is a Lipschitz continuous, compactly supported function which is strictly positive on its support, for example, a radially symmetric cupic spline, or the $d$-fold tensor product of one-dimensional hat functions (in the first case, $\mathrm{supp} W_i$ are $d$-dimensional balls around the points $\boldsymbol{x}_i$, in the second case, the supports are axis parallel cubes). Then, using Shephard's method [9], a partition of unity is built

$$\psi_i(\boldsymbol{x}) = \frac{W_i(\boldsymbol{x})}{\sigma(\boldsymbol{x})}, \qquad \sigma(\boldsymbol{x}) = \sum_k W_k(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^d$$

and the partition functions $\psi_i$ are used as test functions for equation (1). Multiplying (1) with $\psi_i$ and integrating over $\mathbb{R}^d$, we obtain after integration by parts

$$\frac{d}{dt}\int_{\mathbb{R}^d} \psi u \, d\boldsymbol{x} - \int_{\mathbb{R}^d} \boldsymbol{F}(u)\cdot\nabla\psi_i \, d\boldsymbol{x} = 0.$$

In order to split the flux integral into pairwise flux contributions between particle $i$ and its neighboring particles $j$, we use the fact that $\sum_j \psi_j = 1$ and $\nabla(\sum_j \psi_j) = 0$ which leads to

$$\frac{d}{dt}\int_{\mathbb{R}^d} \psi u \, d\boldsymbol{x} - \sum_j \int_{\mathbb{R}^d} \boldsymbol{F}(u)\cdot(\psi_j\nabla\psi_i - \psi_i\nabla\psi_j)\, d\boldsymbol{x} = 0.$$

Assuming that $u$ varies only slightly around $\bar{u}$ on the intersection of the supports of $\psi_i$ and $\psi_j$, we have

$$-\sum_j \int_{\mathbb{R}^d} \boldsymbol{F}(u)\cdot(\psi_j\nabla\psi_i - \psi_i\nabla\psi_j)\, d\boldsymbol{x} \approx \boldsymbol{F}(\bar{u})\cdot\boldsymbol{\beta}_{ij},$$

where

$$\boldsymbol{\beta}_{ij} = \int_{\mathbb{R}^d} \psi_i \nabla \psi_j - \psi_j \nabla \psi_i \, d\boldsymbol{x}. \tag{11}$$

Now, we proceed as in the derivation of the finite volume method. Our discrete quantities are the averages

$$u_i^n = \frac{1}{V_i} \int_{\mathbb{R}^d} \psi_i u \, d\boldsymbol{x}|_{t=t_n}, \qquad V_i = \int_{\mathbb{R}^d} \psi_i \, d\boldsymbol{x} \tag{12}$$

and the flux

$$\boldsymbol{F}(\bar{u}) \cdot \boldsymbol{\beta}_{ij} = |\boldsymbol{\beta}_{ij}| \boldsymbol{F}(\bar{u}) \cdot \boldsymbol{n}_{ij}, \qquad \boldsymbol{n}_{ij} = \frac{\boldsymbol{\beta}_{ij}}{|\boldsymbol{\beta}_{ij}|} \text{ if } |\boldsymbol{\beta}_{ij}| \neq 0$$

is approximated in terms of the discrete values with the help of a numerical flux function

$$\boldsymbol{F}(\bar{u}) \cdot \boldsymbol{n}_{ij} \approx g(u_i, u_j, \boldsymbol{n}_{ij}).$$

Using again an explicit Euler discretization of the time derivative, the resulting scheme has the same structure as the finite volume method (5)

$$u_i^{n+1} V_i = u_i^n V_i - \Delta t \sum_j |\boldsymbol{\beta}_{ij}| g(u_i^n, u_j^n, \boldsymbol{n}_{ij}), \qquad u_i^0 = \frac{1}{V_i} \int_{\mathbb{R}^d} \psi_i u^0 \, d\boldsymbol{x}. \tag{13}$$

A natural reconstruction of a function from the discrete values is given by

$$\tilde{u}(t, \boldsymbol{x}) = \sum_{i=1}^N u_i^n \psi_i(\boldsymbol{x}) \mathbf{1}_{[t_n, t_{n+1})}(t), \qquad \boldsymbol{x} \in \mathbb{R}^d, t \in [0, T]. \tag{14}$$

We remark that the classical finite volume method (5) can be viewed as limiting case of (13) for $\psi_i \to \mathbf{1}_{C_i}$. Note that the characteristic functions $\mathbf{1}_{C_i}$ also form a partition of unity and that (3) and (4) are just (12) and (14) with $\psi_i$ replaced by $\mathbf{1}_{C_i}$. Moreover, if $\psi_i \to \mathbf{1}_{C_i}$, then $\nabla \psi_i$ converges to a surface delta distribution on $\partial C_i$ so that the definition (11) of $\boldsymbol{\beta}_{ij}$ degenerates to a combination of surface integrals. If, for example, the particles $W_i$ are regularly arranged on a square grid and possess a tensor product structure, one can show that $\boldsymbol{\beta}_{ij}$ converges to $|\Gamma_{ij}| \boldsymbol{n}_{ij}$ where $\Gamma_{ij}$ and $\boldsymbol{n}_{ij}$ are calculated from the dual grid of the particle positions.

Instead of considering (5) as a special case of (13), we could also say that (13) is a generalization of the usual finite volume method where the partition functions $\mathbf{1}_{C_i}$ with disjoint supports are replaced by partition functions $\psi_i$ with overlapping supports – or in other words – (13) is a finite volume method with *overlapping finite volumes*.

Before studying the geometric parameters $V_i$ and $\boldsymbol{\beta}_{ij}$ in (13) more closely, let us remark that the derivation works similarly in the case when the particle positions $\boldsymbol{x}_i$ are time dependent (moving particles). The only difference is that the test functions $\psi_i$ and thus also the parameters $V_i$ and $\boldsymbol{\beta}_{ij}$ depend on time. If we use such functions $\psi_i$ as test functions, we find an additional term

$$\int_{\mathbb{R}^d} \psi_i \frac{\partial u}{\partial t}\, d\boldsymbol{x} = \frac{d}{dt}\int_{\mathbb{R}^d} \psi_i u\, d\boldsymbol{x} - \int_{\mathbb{R}^d} u \frac{\partial \psi_i}{\partial t}\, d\boldsymbol{x}.$$

If particles move along a velocity field $\boldsymbol{a}$, i.e. $\dot{\boldsymbol{x}}_i = \boldsymbol{a}(t, \boldsymbol{x}_i)$, then the extra term $\int u \partial_t \psi_i \, d\boldsymbol{x}$ can be combined with the flux integral by replacing $\boldsymbol{F}(u)$ with the Lagrangian flux $\boldsymbol{G}(t, \boldsymbol{x}, u) = \boldsymbol{F}(u) - u\boldsymbol{a}(t, \boldsymbol{x})$ (for details, we refer to [6]). Note that during the movement of the particles, one only has to take care that the supports always cover the domain. Otherwise, there are no restrictions on the movement: particles can move arbitrarily ontop of each other and their "volumes" $V_i$, "interface areas" $|\boldsymbol{\beta}_{ij}|$, and "normals" $\boldsymbol{\beta}_{ij}/|\boldsymbol{\beta}_{ij}|$ are always determined by the formulas (11) and (12).

Let us now investigate these geometrical parameters. In order to show that they are reasonable, we check the conditions on the parameters that are needed in the convergence proof of classical finite volume methods. First, condition (6) can be ensured by setting up the particle positions in such a way that the points do not cluster too much. Assuming that the maximal number of overlapping particles is $K$, we have

$$\sigma(\boldsymbol{x}) = \sum_k W_k(\boldsymbol{x}) \le K\|W\|_\infty, \qquad \forall \boldsymbol{x} \in \mathbb{R}^d$$

and an estimate of $\sigma(\boldsymbol{x}) \ge \sigma_{min} > 0$ follows if we assume a certain minimal overlap of the particles. Then, conditions (7) follow by direct calculation. We have

$$V_i = \int_{\mathbb{R}^d} \psi_i \, d\boldsymbol{x} = \int_{\mathbb{R}^d} \frac{W_i}{\sigma}\, d\boldsymbol{x} \ge \frac{1}{K\|W\|_\infty} h^d \int_{\mathbb{R}^d} W(\boldsymbol{y})\, d\boldsymbol{y} = \alpha h^d$$

and since

$$|\boldsymbol{\beta}_{ij}| \le \int |\nabla \psi_i| + |\nabla \psi_j| \, d\boldsymbol{x}, \qquad \nabla \psi_i = \frac{\nabla W_i}{\sigma} - \psi_i \frac{\sum_k \nabla W_k}{\sigma},$$

we obtain

$$\int_{\mathbb{R}^d} |\nabla \psi_i| \, d\boldsymbol{x} \le \frac{K+1}{h\sigma_{min}} \int_{\mathbb{R}^d} |\nabla W|(\boldsymbol{x}/h)\, d\boldsymbol{x} = \frac{K+1}{\sigma_{min}} h^{d-1} \int_{\mathbb{R}^d} |\nabla W(\boldsymbol{y})|\, d\boldsymbol{y}$$

so that $|\boldsymbol{\beta}_{ij}| \le C h^{d-1}$. Next, the algebraic conditions (8) follows directly from the skew-symmetric definition of $\boldsymbol{\beta}_{ij}$ and (9) is a consequence of the fact that $\sum_j \psi_j = 1$, $\nabla(\sum_j \psi_j) = 0$, $\int \nabla \psi_i \, d\boldsymbol{x} = \boldsymbol{0}$. More generally, in domains $\Omega$ with boundary, we find

$$\sum_j \boldsymbol{\beta}_{ij} = \sum_j \int_\Omega \psi_i \nabla \psi_j - \psi_j \nabla \psi_i \, d\boldsymbol{x} = -\int_\Omega \nabla \psi_i \, d\boldsymbol{x} = -\int_{\partial\Omega} \psi_i \boldsymbol{n}\, dS$$

which parallels (10).

Hence, we can adopt the stability proofs from classical finite volume schemes: if we use a consistent, conservative, Lipschitz continuous, monotone numerical flux function $g$, a discrete $\mathbb{L}^\infty$ estimate, a weak BV estimate and also a discrete entropy inequality can be shown. We conclude that it is possible to set up a reasonable finite volume scheme with geometric coefficients which are not determined form a grid but from a partition of unity, or in other words,

*we can replace grid generation by integration.*

## 4 Comparing FVM and FVPM

In this section, we want to investigate whether the generation of geometric parameters by integration can be a promising alternative to the conventional grid-based approach. First, we note that in the case of regular grids, the generation of the parameters practically involves no computational costs at all, because the parameters can be determined in advance from certain reference cells. The same is true for the partition of unity approach, if we choose a regular distribution of points (for example, the cell centers of the regular grid). Then, all integrals can also be calculated in advance from some reference configurations of partition functions.

Therefore, let us now concentrate on situations where an irregular point distribution is given in the computational domain (for example, such a distribution could result from a particle movement). In order to set up a finite volume grid where the grid cells are determined by the given points, we use a Voronoi tesselation, i.e. the cell $C_i$ contains all points $\boldsymbol{x}$ which are closer to $\boldsymbol{x}_i$ than to any other point $\boldsymbol{x}_j$. For the partition of unity, we use Shephard's functions $W$ of tensorial structure because the rectangular support of these functions allows a fast determination of support intersections.

To have an explicit example, we consider a model problem on the unit square $\Omega$ in two space dimensions (see Fig. 2).
The entropy solution is indicated in Fig. 3 for several instances in time. The last figure shows the stationary solution with a compression fan and a jump discontinuity between two constant regions $u = 1.5$ and $u = -1.0$.
An example of an irregular point distribution is given in Fig. 4. In our calculation, we will use 10.000 points in the unit square. A small clipping of the Voronoi grid is also indicated in Fig. 4.
To get an idea of the computational costs, we note that the generation of the geometric parameters $V_i$, $\boldsymbol{\beta}_{ij}$ takes about 25 seconds on a PC by setting up a Voronoi grid. The partition of unity uses the product of hat functions

$$W(\boldsymbol{x}) = w(x_1)w(x_2), \qquad w(x) = (1 - |x|)_+.$$

A clipping of the cover of $\Omega$ is also shown in Fig. 4. The corresponding functions $\psi_i$ are piece-wise rational functions and we use numerical quadrature to

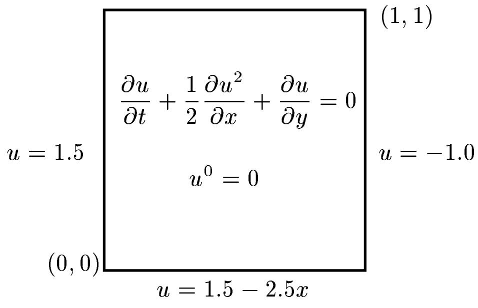


**Fig. 2.** The model problem with initial and boundary values.

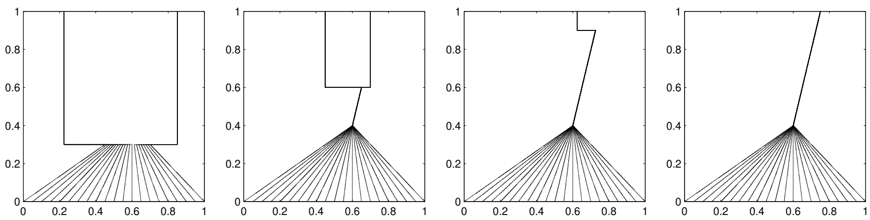

**Fig. 3.** The entropy solution of the model problem for $t \in \{0.3, 0.6, 0.9, 1.2\}$.

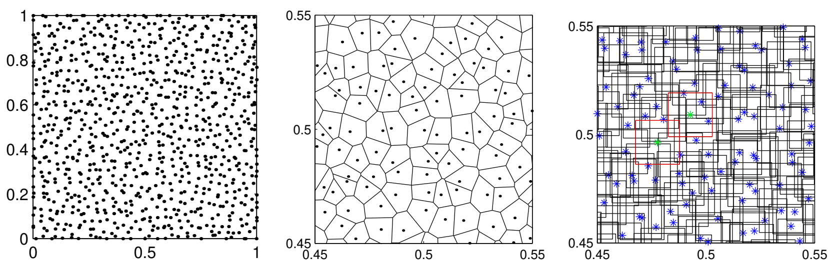

**Fig. 4.** Example of an irregular point distribution, a clipping of the Voronoi grid, and the supports of the particle cover.

evaluate the parameters $V_i$ and $\boldsymbol{\beta}_{ij}$ (Gauss-Legendre integration on patches where the integrands are $C^\infty$ functions). The time consumption then depends on the number of integration nodes we use. A coarse integration requires 27 seconds, and thus a little bit more than the grid generation. However, a very accurate integration easily leads to computation times of 300 seconds. The stationary solution of our model problem calculated on a Voronoi grid and with FVPM, using two levels of accuracy for the parameter integration, are shown in Fig. 5. The values of the left and right most level lines correspond to the boundary values. Obviously, the coarse integration of the parameters leads to oscillatory solutions: the values in the constant regions range between 1.4 and 1.6, respectively -1.1 and -0.9, and also the isolines in the compression region are quite wavy.

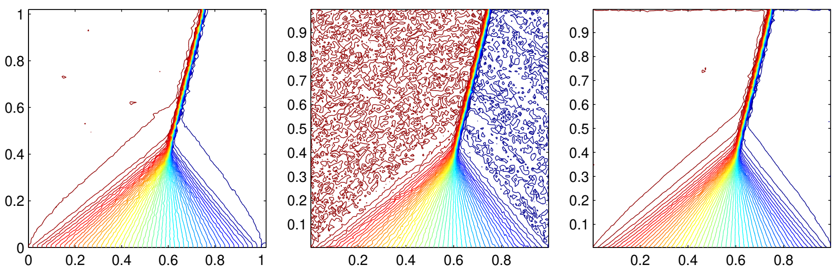


**Fig. 5.** Stationary solution using Voronoi grid (left), FVPM with coarse integration (middle), and FVPM with fine integration (right).

To understand the origin of these oscillations, let us consider the case of a conservation law on $\mathbb{R}^d$ with constant initial value $u^0(\boldsymbol{x}) = c$. Then, the algorithm (13) implies $u_i^0 = c$ for all $i$ and, using consistency of $g$, i.e. $g(c, c, \boldsymbol{n}_{ij}) = \boldsymbol{F}(c) \cdot \boldsymbol{n}_{ij}$, we find the first iterates as

$$u_i^1 = c - \frac{\Delta t}{V_i} \boldsymbol{F}(c) \cdot \sum_j \boldsymbol{\beta}_{ij}, \qquad \forall i.$$

We see that the constant state will, in general, only be preserved if $\sum_j \boldsymbol{\beta}_{ij} = \mathbf{0}$ which is exactly condition (9). If we determine $\boldsymbol{\beta}_{ij}$ by numerical integration, the sum $\sum_j \boldsymbol{\beta}_{ij}$ is of the order of the integration error and, if $\Delta t / V_i$ is of order one (such a CFL-type condition is used for stability reasons), exactly this integration error becomes visible in regions where the solution is constant.

In order to ensure condition (9) also when $\boldsymbol{\beta}_{ij}$ is calculated numerically, several correction procedures have been proposed. In [10], a coarse integration yields $\tilde{\boldsymbol{\beta}}_{ij}$ and then conditions (8) and (9) are enforced for $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}$ where $\hat{\boldsymbol{\beta}}$ is determined using a least squares method. A faster correction which does not require the solution of large linear systems has been developed in [7]. Here, the error in $\sum_j \boldsymbol{\beta}_{ij}$ is subtracted from some non-zero $\boldsymbol{\beta}_{ij_0}$ with $j_0 > i$ (and added to $\boldsymbol{\beta}_{j_0 i}$ to keep property (8)). In other words, the error is moved to a neighboring particle and one can show that it does not accumulate. In [13], this procedure has been extended to condition (10) which is the counterpart of (9) in domains with boundary. Using this correction procedure together with the coarse integration, the computational time for the calculation of geometric parameters increases from 27 seconds only to 28 seconds but the result improves drastically (see Fig. 6). In fact, a comparison of the $\mathbb{L}^1$ error shows that the error is slightly smaller than in the Voronoi-FVM solution.

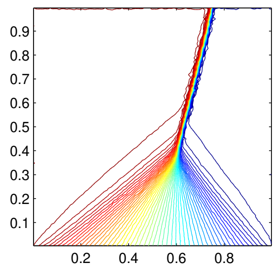


**Fig. 6.** Stationary solution using FVPM with coarse integration and parameter correction.

## 5 The PUMESH method

In this section, we present another approach how to calculate geometric parameters based on a partition of unity. The basic idea is to associate a mesh to the given partition of unity (a *PU-mesh*) and to use a grid based finite volume method on this mesh. The mesh construction is easily explained for the regular partition of unity shown in Fig. 7.

If the partition functions are denoted $\psi_i$, we now set up the functions (see Fig. 8)

$$\Phi_1 = \psi_1, \ \Phi_2 = \psi_1 + \psi_2, \ \ldots, \ \Phi_k = \sum_{i \le k} \psi_i.$$

If we plot all functions $\Phi_k$ together, we eventually obtain a structure as shown in Fig. 9 which can be considered as a mesh on $\Omega \times (0,1)$ where $\Omega = (0,1)$ is the domain where the partition of unity is given.

The grid cells are naturally defined by

$$C_k = \{(\boldsymbol{x}, y) : \Phi_{k-1}(\boldsymbol{x}) < y < \Phi_k(\boldsymbol{x})\}. \tag{15}$$

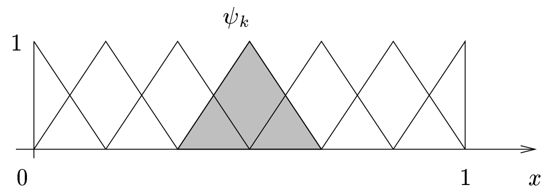


**Fig. 7.** A regular partition of unity with hat functions on $[0, 1]$.

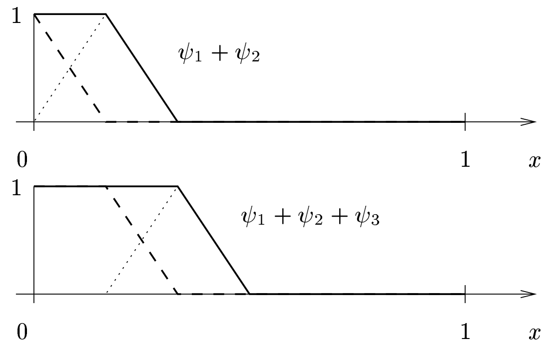


**Fig. 8.** Sum of partition functions.

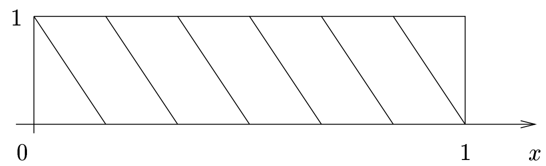


**Fig. 9.** A grid (PU-mesh) associated to the partition of unity from Fig. 7.

In particular, upper and lower boundary of $C_k$ are the graphs of $\Phi_{k-1}$ and $\Phi_k = \Phi_{k-1} + \psi_k$

$$\partial C_k = \{(\boldsymbol{x}, \Phi_{k-1}(\boldsymbol{x})) : \boldsymbol{x} \in \operatorname{supp}\psi_k\} \cup \{(\boldsymbol{x}, \Phi_k(\boldsymbol{x})) : \boldsymbol{x} \in \operatorname{supp}\psi_k\}. \tag{16}$$

Applying this construction to a partition of unity with irregularly distributed partition functions and either uniform or varying support size, we find grids as shown in Fig. 10. We remark that the numbering of the partition functions obviously influences the form of the grid: if we count the partition functions in Fig. 11 from left to right we get one possible grid - another possibility is obtained if we renumber by first taking the particles with even index from left to right and then the particles with odd index from right to left. Finally, a typical random permutation of the indices leads to grids which resemble the lower right situation in Fig. 11.

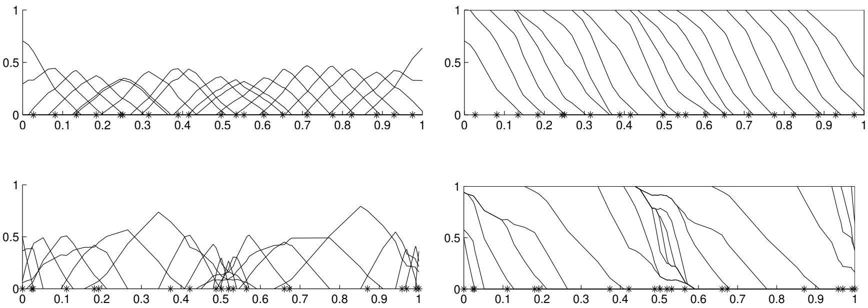

**Fig. 10.** Partitions of unity constructed from irregular point distributions and hat functions with uniform (upper left) and non-uniform (lower left) support sizes. The corresponding PU-meshes are shown in the right columns.

Now let us come back to the original goal to construct a finite volume type method for equation (1) on the domain $\Omega$. First, we lift this equation to the extended domain $\Omega_e = \Omega \times (0,1)$ by introducing the extended flux function $\boldsymbol{F}_e(u) = (\boldsymbol{F}(u), 0)$ and the extended initial value $u_e^0(\boldsymbol{x}, y) = u^0(\boldsymbol{x})$. Then, the problem

$$\frac{\partial u_e}{\partial t} + \operatorname{div}_{(\boldsymbol{x},y)} \boldsymbol{F}_e(u_e) = 0, \qquad u_e(0, \boldsymbol{x}, y) = u_e^0(\boldsymbol{x}, y). \tag{17}$$

is obviously equivalent to (1) and our idea is to approximate (17) using a finite volume approach on a PU-mesh. This plan requires the calculation of cell volumes, cell surface areas, and cell normals. To begin with, let us determine the volume $V_k$ of cell $C_k$ defined in (15). We have

$$V_k = \int_{C_k} dy d\boldsymbol{x} = \int_{\mathbb{R}^d} \int_{\Phi_{k-1}(\boldsymbol{x})}^{\Phi_k(\boldsymbol{x})} dy\, d\boldsymbol{x} = \int_{\mathbb{R}^d} \psi_k \, d\boldsymbol{x}$$

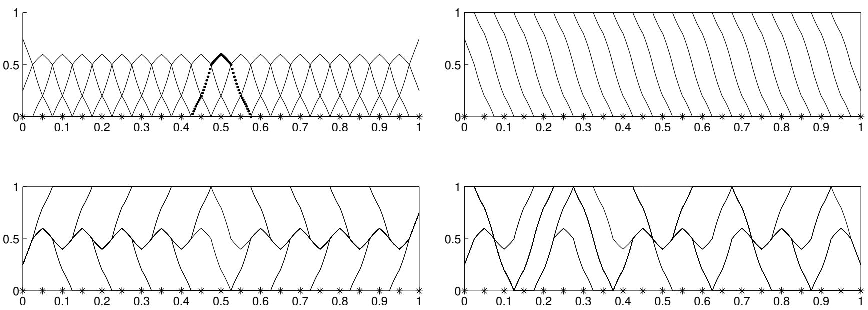


**Fig. 11.** Several PU-meshes constructed from the regular partition of unity shown in the upper left diagram. Counting the functions from left to right, the upper right PU-mesh is obtained. A renumbering by first taking even indices in increasing and then odd indices in decreasing order leads to the lower left PU-mesh. A random permutation of the indices yields the lower right mesh.

which is also used in (12) for the FVPM approach. Next, we calculate the cell averages of the initial value $u_e^0$

$$u_{ek}^0 = \frac{1}{V_k}\int_{C_k} u_e^0(\boldsymbol{x}, y)\,dyd\boldsymbol{x}$$

$$= \frac{1}{V_k}\int_{\mathbb{R}^d} u^0(\boldsymbol{x})\int_{\Phi_{k-1}(\boldsymbol{x})}^{\Phi_k(\boldsymbol{x})} dy\,d\boldsymbol{x} = \frac{1}{V_k}\int_{\mathbb{R}^d} u^0(\boldsymbol{x})\psi_k(\boldsymbol{x})\,d\boldsymbol{x}$$

which is again the same as in FVPM.

In order to determine the interface $\Gamma_{kj}$ between two cells $C_k$ and $C_j$, we note that the boundary of $C_k$ is given by (16). In particular, a cell $C_j$ with a larger index $j > k$ can only touch the upper boundary

$$\partial^+ C_k = \{(\boldsymbol{x}, \Phi_k(\boldsymbol{x})) : \boldsymbol{x} \in \mathrm{supp}\psi_k\}$$

with parts of its lower boundary

$$\partial^- C_j = \{(\boldsymbol{x}, \Phi_{j-1}(\boldsymbol{x})) : \boldsymbol{x} \in \mathrm{supp}\psi_j\}$$

if $\Phi_{j-1}(\boldsymbol{x})$ coincides with $\Phi_k(\boldsymbol{x})$ for at least one point $\boldsymbol{x} \in \mathrm{supp}\psi_k \cap \mathrm{supp}\psi_j$. Hence, the (relative interior of the) interface is given by

$$\Gamma_{kj} = \{(\boldsymbol{x}, \Phi_k(\boldsymbol{x})) : \boldsymbol{x} \in D_{kj}\}, \qquad j > k$$

where

$$D_{kj} = \int \{\boldsymbol{x} : \Phi_{k-1}(\boldsymbol{x}) < \Phi_k(\boldsymbol{x}) = \Phi_{j-1}(\boldsymbol{x}) < \Phi_j(\boldsymbol{x})\}.$$

Since the interface $\Gamma_{kj}$ is part of the graph of $\Phi_k$, we immediately find the normal vector (pointing out of $C_k$) and the surface measure in terms of $\Phi_k$ as

$$\boldsymbol{n}_{kj} = (-\nabla\Phi_k, 1)/\sqrt{1+|\nabla\Phi_k|^2}, \qquad dS = \sqrt{1+|\nabla\Phi_k|^2}d\boldsymbol{x}. \tag{18}$$

Using (18), we introduce an average interface area and an average normal vector, by setting for $j > k$

$$|\bar{\Gamma}_{kj}|\bar{\boldsymbol{n}}_{kj} = \int_{\Gamma_{kj}} \boldsymbol{n}_{kj}\, dS = \int_{D_{kj}} (-\nabla\Phi_k, 1)\, d\boldsymbol{x}$$

and $|\bar{\Gamma}_{jk}| = |\bar{\Gamma}_{kj}|$, $\bar{\boldsymbol{n}}_{jk} = \bar{\boldsymbol{n}}_{kj}$, $|\bar{\Gamma}_{kk}| = 0$. Finally, we have to specify a numerical flux function $g_e$ corresponding to $\boldsymbol{F}_e$ which we also obtain by lifting a suitable flux function $g$ for the original flux $\boldsymbol{F}$. If $\boldsymbol{n} = (\boldsymbol{n}_{\boldsymbol{x}}, n_y)$ is a unit vector with $\boldsymbol{n}_{\boldsymbol{x}} \neq \boldsymbol{0}$, we define

$$g_e(v, w, \boldsymbol{n}) = |\boldsymbol{n}_{\boldsymbol{x}}| g(v, w, \boldsymbol{n}_{\boldsymbol{x}}/|\boldsymbol{n}_{\boldsymbol{x}}|)$$

and set $g_e(v, w, \boldsymbol{n}) = 0$ if $\boldsymbol{n}_{\boldsymbol{x}} = \boldsymbol{0}$. (This construction reflects the fact that there is no flux in $y$-direction, i.e. $(\boldsymbol{F}_e)_y = 0$). Then, the classical finite volume ansatz (5)

$$u_i^{n+1}V_i = u_i^n V_i - \Delta t \sum_j |\bar{\Gamma}_{ij}| g(u_i^n, u_j^n, \bar{\boldsymbol{n}}_{ij}), \qquad u_i^0 = \frac{1}{V_i}\int_{C_i} u_e^0\, dy d\boldsymbol{x}$$

reduces to

$$u_i^{n+1}V_i = u_i^n V_i - \Delta t \sum_j |\boldsymbol{\beta}_{ij}| g(u_i^n, u_j^n, \boldsymbol{n}_{ij}), \quad u_i^0 = \frac{1}{V_i}\int_{\mathbb{R}^d} \psi_i u^0\, d\boldsymbol{x} \tag{19}$$

with $\boldsymbol{n}_{ij} = \boldsymbol{\beta}_{ij}/|\boldsymbol{\beta}_{ij}|$ and

$$V_i = \int_{\mathbb{R}^d} \psi_i\, d\boldsymbol{x}, \quad \boldsymbol{\beta}_{ij} = -\int_{D_{ij}} \nabla\Phi_i\, d\boldsymbol{x},\ \boldsymbol{\beta}_{ji} = -\boldsymbol{\beta}_{ij},\ j > i, \quad \boldsymbol{\beta}_{ii} = \boldsymbol{0}.$$

We remark that (19) has precisely the form of the FVPM algorithm (13) – only the definition of $\boldsymbol{\beta}_{ij}$ is different. In particular, the extra dimension of the PU-mesh is no longer visible in the method. Only the fact that integration has to be carried out to determine $V_i$ and $\boldsymbol{\beta}_{ij}$ can be viewed as remnant of the additional dimension. However, if we reconstruct a function from the discrete values $u_i^n$, the natural choice is, according to our derivation,

$$\tilde{u}_e(t, \boldsymbol{x}, y) = \sum_{i=1}^N u_i^n \mathbf{1}_{C_i}(\boldsymbol{x}, y)\mathbf{1}_{[t_n, t_{n+1})}(t).$$

Since the solution $u_e$ of (17) is independent of $y$, it is reasonable to suppress the additional dimension also in the reconstruction. This can be achieved by using the local $y$-average of $\tilde{u}_e$ which, in view of (15), can be written as

$$\tilde{u}(t, \boldsymbol{x}) = \int_0^1 \tilde{u}_e(t, \boldsymbol{x}, y)\, dy = \sum_i u_i^n \psi_i(\boldsymbol{x}) \mathbf{1}_{[t_n, t_{n+1})}(t).$$

Note that this is the same reconstruction formula as (14) for FVPM. In Fig. 12, we compare the stationary FVPM solution of our model problem with the PUMESH solution. The partition functions are numbered according to the lexicographical ordering of the given points $\boldsymbol{x}_i$. The parameter integrals in the PUMESH method can be carried out similar to the FVPM case.

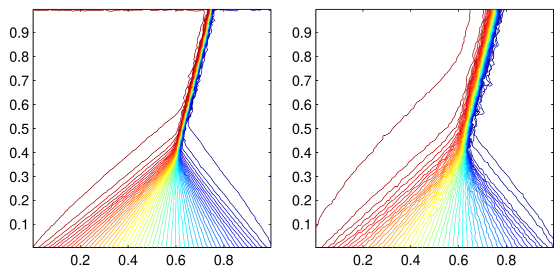


**Fig. 12.** Stationary solution using FVPM (left) and PUMESH (right).

One can see that FVPM has a better shock resolution than PUMESH, where the resolution appears to be dependend on the chosen numbering. However, PUMESH has the advantage that the number of relevant neighbors is smaller than in FVPM which leads to an increase in speed. Moreover, the calculation $\boldsymbol{\beta}_{ij}$ can be carried out in such a way that the stability conditions (8) and (9) are satisfied on a discrete level without additional correction.

## 6 Conclusion

We have presented two finite volume type schemes (FVPM and PUMESH) which are based on a partition of unity instead of a grid on the computational domain. The geometrical parameters (cell volumes, cell surface areas, and cell normal vectors) used in grid based finite volume methods (FVM) are replaced by integrals over partition functions and their derivatives. For the case of FVPM, we have shown that these geometrical parameters satisfy the conditions used in convergence proofs for classical FVM and a similar proof can be given for the PUMESH parameters. Since the numerical results obtained with both partition of unity methods are reliable and robust, we can conclude that finite volume methods do *not* necessarily require a grid - reasonable geometrical parameters can also be defined differently. In future works, advantages of FVPM and PUMESH over classical FVM due to the possible overlap of finite volumes will be investigated.

## Acknowledgement

This work has been carried out in the project *Particle Methods for Conservation Systems* NE 269/11-3 which is part of the DFG - Priority Research Program *Analysis and Numerics for Conservation Laws.*

## References

1. Chainais-Hillairet, C.: Finite volume schemes for a nonlinear hyperbolic equation. Convergence towards the entropy solution and error estimate. $M^2AN$ **33** (1999) 129-156.
2. Cockburn, B., Coquel, F., Lefloch, P.: An error estimate for finite volume methods for multidimensional conservation laws. Math. Comput. **63** (1994) 77-103
3. Eymard, R., Gallouët, T.: Convergence d'un schéma de type éléments finis - volumes finis pour un système formé d'une équation elliptique et d'une équation hyperbolique. $M^2AN$, Modélisation mathématique et analyse numerique, **27** (1993) 843-862
4. Godlewski, E., Raviart, P.-A.: Numerical Approximation of Hyperbolic Systems of Conservation Laws. Applied Mathematical Sciences **118**, Springer (1996)
5. Hietel, D., Steiner, K., Struckmeier, J.: A finite-volume particle method for compressible flows. Math. Models Methods Appl. Sci. **10** (2000) 1363–1382
6. Junk, M., Struckmeier, J.: Consistency analysis for mesh-free methods for conservation laws. AG Technomathematik, Universität Kaiserslautern, preprint **226** (2000)
7. Keck, R.: PhD Thesis, Universität Kaiserslautern, in preparation.
8. Kröner, D.: Numerical Schemes for Conservation Laws. Wiley Teubner (1997)
9. Shepard, D.: A two-dimensional interpolation function for irregularly spaced points. Proceedings of A.C.M National Conference (1968) 517-524
10. Teleaga, D.: Numerical studies of a finite-volume particle method for conservation laws. Master thesis, Universität Kaiserslautern, 2000.
11. Vila, J.-P.: Convergence and error estimates in finite volume schemes for general multidimensional scalar conservation laws. I. Explicite monotone schemes. $M^2AN$ **28** (1994) 267-295
12. Vovelle, J.: Convergence of finite volume monotone schemes for scalar conservation laws on bounded domains. Numer. Math., Online First Publications (2001)
13. Yang, Z.: Efficient Calculation of Geometric Parameters in the Finite Volume Particle Method. Master thesis, Universität Kaiserslautern, 2001.