# A Fast High-Resolution Algorithm for Linear Convection Problems: Particle Transport Method

Anton Smolianski* Olga Shipilova† Heikki Haario‡

**Abstract**

The paper is devoted to a novel explicit technique, the Particle Transport Method (PTM), for solving linear convection problems. While being a Lagrangian (characteristic based) method, PTM has the advantage of Eulerian methods to represent the solution on a fixed mesh. The proposed approach belongs to the class of monotone high-resolution numerical schemes, possesses the property of unconditional stability and works with structured and unstructured meshes. It is also demonstrated that the method has a linear computational complexity. The performance of the presented algorithm is tested on one- and two-dimensional benchmark problems. The numerical results confirm that the method has the 2nd-order spatial accuracy and can be significantly faster than the grid-based methods of the same order.

*Keywords:* linear convection, particle transport, monotone, high-resolution, unconditionally stable, linear complexity

## 1 Introduction

Convective transport is one of the most significant physical phenomena in the mechanics of fluids and in the heat and mass transfer processes. It also has a primary importance in many problems of population biology and plasma physics. The mathematical modelling of this phenomenon results in a hyperbolic equation whose numerical resolution may present a substantial difficulty due to possible discontinuities or high gradients of the exact solution. It is well known that special care must be taken in order to obtain a non-oscillatory and sufficiently accurate numerical scheme for a convection problem. Over the last three decades, the general principles of the design of numerical techniques for hyperbolic problems have been established as well as the main requirements to a proper numerical scheme: monotonicity, capability of high-resolution/adaptivity, good accuracy and stability, low computational cost. It is clear that some of these criteria are of purely qualitative nature, and there is a usual trade-off between many of them (e.g., between accuracy and computational cost). This fact led to creation of numerous competitive algorithms for convection problems (see, e.g., books [16], [18], [33], [36], [40]); the most general classification of such methods is borrowed from fluid mechanics and is based on a division into three groups: Eulerian, Lagrangian and mixed Eulerian-Lagrangian.

*Institute of Mathematics, Zurich University, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland; *E-mail:* antsmol@amath.unizh.ch

†Laboratory of Applied Mathematics, Lappeenranta University of Technology, P. Box 20, 53851 Lappeenranta, Finland; *E-mail:* olga.shipilova@lut.fi

‡Laboratory of Applied Mathematics, Lappeenranta University of Technology, P. Box 20, 53851 Lappeenranta, Finland; *E-mail:* haario@csc.fi

The idea of the Eulerian concept is to examine the solution changes within a fixed region of the space, therefore, in this description a computational mesh is fixed. This makes the visualization and interpretation of the numerical results convenient, and also allows for an easy coupling of convection with other physical processes (like, e.g., diffusion) within the operator-splitting framework (see, e.g., [41]). The main problem of explicit Eulerian methods is the stability constraint on the time-step size known as the Courant-Friedrichs-Levy (CFL) condition. This condition reflects the characteristic connection between time and space in hyperbolic problems and may be very restrictive, resulting in small time steps and, correspondingly, long computing times. However, the Eulerian schemes still remain extremely attractive, and a lot of works have been devoted to this approach. Here we could mention, for instance, the streamline-upwind Petrov-Galerkin method ([8], [25]), the Galerkin/least-squares method ([12], [26]), the Taylor-Galerkin method ([14], [15]), the finite-volume method (see [36] and the references therein) and the discontinuous Galerkin method ([1], [11], [23], [32], [50]).

The Lagrangian concept is based on the idea to follow the motion of the fluid particles along their trajectories and, thus, is well suited to reflect the nature of convective transport phenomena. In contrast to Eulerian schemes, the Lagrangian methods require less strict conditions on the time step. Unfortunately, severe problems occur when the discretization is performed on a moving Lagrangian mesh; it can become excessively distorted and hence yield a poor accuracy of the approximation. Due to this reason, the purely Lagrangian methods are rarely applied for the solution of the convection problems.

A good alternative to the purely Lagrangian and Eulerian approaches seems to be their combination, the mixed Eulerian-Lagrangian approach. In this field, several competitive methods have been developed, in particular one could mention the Eulerian-Lagrangian Localized Adjoint Method (ELLAM), see, e.g., [4], [17], [22], the characteristic Galerkin method ([31]), the Lagrange-Galerkin method ([43], [46]) as well as semi-Lagrangian ([28], [45]) and Free-Lagrangian methods ([13], [19]). While operating on the basis of the Lagrangian concept, this kind of schemes keeps the advantage of the presentation of the solution on a fixed Eulerian grid. Namely, the Eulerian-Lagrangian methods essentially rely on the idea of "exact transport + projection" (see [30]). The first step (the transport) may be accomplished with either forward or backward tracking of the characteristics; the second step (the projection onto a fixed grid), although may seem purely technical, is important and requires a special care. For example, the general way of the interpolation between two arbitrary meshes is an $L_2$-projection, but this approach can be overly diffusive and, moreover, not conservative (see the discussion in [19]). There exists a number of works devoted to the development of efficient mass-conservative projection, see, e.g., [19], [39], [47].

In spite of different concepts underlying the Eulerian and Lagrangian schemes, all numerical methods for convection problems can be equally judged with respect to the capability of accurate capturing the discontinuities of the exact solution while retaining the solution monotonicity. There are quite a few works on the numerical schemes possessing this property. The classical strategy to preclude spurious oscillations is to add an artificial ("shock-capturing") viscosity in the vicinity of singularities, see e.g., [44]. This technique can be easily applied to an arbitrary method. Unfortunately, the magnitude of the viscosity depends on the solution, that makes the whole scheme nonlinear and may cause the ineffective elimination of oscillations and the downgrade of the accuracy in the smooth part of the solution. Other approaches to the design of the high-resolution schemes have been proposed, starting from the work [5], where the concept of Flux-Corrected-Transport (FCT) was introduced, and ranging over the Total Variation Diminishing (TVD) schemes

(see, e.g., [20], [36]), the Essentially Non-Oscillatory (ENO) schemes (see, e.g., [9], [21], [48], [49]) and their numerous modifications. The finite-element FCT (FEM-FCT) (see [34], [35], [38] and also [40]) and the discontinuous Galerkin schemes (see, e.g., [1], [11], [23], [32], [50]) have been recently developed with particular emphasis on the computations with unstructured grids. The important feature of all existing high-resolution schemes is the use of flux limiters (see, e.g., [36]). The latter are needed to suppress nonphysical oscillations; however, the limiters depend nonlinearly on the solution, hence, making the whole numerical scheme nonlinear.

Besides being monotone and accurate in the smooth parts of the solution, a high-resolution numerical scheme should, as its name suggests, resolve sufficiently well the solution discontinuities. It is clear that this task can be accomplished only by virtue of an appropriate spatial (and, maybe, temporal) adaptivity. The commonly used mesh-adaptivity is a powerful tool for resolving singularities; however, it requires the mesh reconstruction, at least near the discontinuities, on every time step. The mesh change itself may be rather expensive, especially in 3D case, and it also implies the necessity of the data projection between the old and the new grids, see, e.g., [28]. In this respect, very promising approaches seem to be the meshless (the grid-free) and the particle concepts that are closely related to each other. The comprehensive overviews of both approaches can be found, e.g., in [3], [6], [7], [37]. As important particular cases of the particle methods, very popular in computational fluid mechanics community, we would mention the Vortex Method (VM) (see, e.g., [51]) and the Smooth Particle Hydrodynamics (SPH) method (see, e.g., [42]). The idea to eliminate the traditional mesh-based view on the computational domain is over twenty years old but has not attracted proper attention until recently. At the same time, meshless schemes are well suited for the problems with moving solution discontinuities, since they operate with particles, which dramatically simplifies the adaptivity procedure (see, e.g., [2]). Moreover, the particle based methods allow to work efficiently with free-boundary problems and problems in complex geometries, see, e.g., [24], [27].

In this work, we utilize the particle meshless concept to create an explicit, unconditionally stable, semi-Lagrangian scheme, called the Particle Transport Method (PTM). The scheme is based on the classical method of characteristics that amounts to the exact solution transport on each time step. Instead of a moving mesh, we use a system of particles which can be projected onto any fixed mesh with a special, fast and monotone projection technique. It is worth noticing that a simple linear interpolation constitutes the core of the projection procedure, unlike the "radial basis functions" approximation typical for the meshless methods (see [2]). The transfer from the particle system to the stationary mesh may be done on every time step or just when it is required. It is important to note that no attention is paid to the particle connectivity, which greatly simplifies the numerical scheme. Using the almost unlimited capabilities of the spatial adaptivity within the particle meshless framework, it turns out to be possible to develop a high-resolution numerical scheme (at least, for the linear convection problems) without resorting to nonlinear tools like shock-capturing viscosity or flux limiters. Since there is no need to assemble matrices and solve any system of equations, the scheme is computationally cheap, and we demonstrate that it has optimal linear complexity with respect to the number of particles as well as the number of nodes of the fixed grid used for the projection.

The rest of the paper is organized as follows. A linear convection problem is formulated in Section 2 along with some preliminary definitions. In Section 3, the detailed description of PTM is given; such features of the proposed method as linear complexity, accuracy and monotonicity are also discussed. Section 4 is devoted to computational experiments.

A step-function transport, a wave-packet convection, a rigid-body rotation of a slotted cylinder and a transport of smooth *sine*-surface are presented. The results for the wave-packet convection are compared with the solution obtained by the high-resolution finite volume method ([36]), while the *sine*-surface transport test is chosen for the comparison of the computing time and accuracy with those of the discontinuous Galerkin method ([50]). Some conclusions are drawn in Section 5.

## 2 Problem setting and preliminaries

In this paper we consider a linear convection problem

$$u_t + (\mathbf{v} \cdot \nabla)u = 0 \qquad \text{in} \quad \Omega \times (0,T)\,, \tag{1}$$

where $u = u(\mathbf{x},t)$ and the space variable $\mathbf{x}$ belongs to the bounded domain $\Omega \subset \mathbb{R}^d$ $(d = 1,2,3)$. It is assumed that the convective velocity field $\mathbf{v} = \mathbf{v}(\mathbf{x},t)$ is given in $\Omega \times (0,T)$ and is solenoidal, i.e. $\nabla \cdot \mathbf{v} = 0$ in $\Omega$ at any moment of time.

Equation (1) is supplemented by the initial and boundary conditions

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

$$u(\mathbf{x},t) = u_{in}(\mathbf{x},t) \qquad \text{on} \quad \Sigma_{in} \times (0,T)\,, \tag{3}$$

where $\Sigma_{in}$ is the inflow part of the domain boundary $\Sigma = \partial\Omega$.

It is well known that the solution of the linear initial boundary-value problem (1)–(3) has the property to retain its value along characteristic lines (trajectories) which are defined by the ordinary differential equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x},t). \tag{4}$$

This equation forms one of the main parts of all Lagrangian and semi-Lagrangian algorithms.

As the starting point of our algorithm, we use a stationary mesh in the domain $\Omega$ that will further be called *the grid*. We suppose that the grid is conforming and shape-regular (see [10]) but can be unstructured and composed of both triangular and quadrilateral elements. The grid is not used for solving the problem but only for the representation of the solution, which is especially important when the proposed scheme is combined with the Eulerian methods, e.g., within the operator-splitting framework. The transport of the exact solution is carried out by moving the system of particles, further called *the particles*. The grid nodes define the initial locations of the particles; the particles do not have to be placed at all nodes but, for example, at each $m$-th node, where the number $m$ can be rather large. In fact, only a small initial amount of particles is needed, while the adaptive algorithm discussed below will automatically add more particles in the regions that can be crucial for numerical solution.

## 3 The particle transport method

The particle transport method is based on two main ideas. First of all, it is well known that the singularities of the solution function passively convected by an *incompressible* flow are completely determined by initial and inflow boundary conditions. Thus, the approximation should be correspondingly improved ("adapted") at the initial step and, then, near the inflow part of the boundary only. Secondly, by controlling the density of the particles by

means of an adaptivity procedure we can reach any order of approximation accuracy in the given regions of the domain. These ideas are extensively used in the proposed algorithm. The method consists of three key components: (i) adaptive distribution of the particles, (ii) particles movement and (iii) projection of the solution from the particle system onto the grid.

Further, for detailed consideration we confine ourselves mostly to two-dimensional case; however, the treatment of one- and three-dimensional cases is absolutely analogous. We also consider only the grids composed of triangular elements, although the extension to the grids consisting of quadrilaterals and to the mixed (hybrid) grids does not present any difficulty.

### 3.1 Adaptive distribution of the particles

As mentioned above, the set of grid nodes is the basis for creating the particles. In order to balance the requirements of good approximation quality and small computational cost, one certainly needs an adaptivity. The main idea is to make the system of particles more dense in the "dangerous" regions of the solution, in particular, in the vicinities of its steep fronts and discontinuities. For this, one can use the solution gradient as a good indicator of the singularities. On the other hand, the areas where the solution is smooth also require some attention, if we later want to achieve a reasonable accuracy using linear interpolation to project the solution from the particle system onto the grid (the 3rd step of the algorithm). Here, the second derivatives of the solution are quite decisive, as well known from the interpolation estimates (see, e.g., [10]).

Thus, two different adaptivity procedures are proposed, each based on its own indicator ("signal value") for the addition of new particles. The adaptivity for the resolution of steep fronts, so-called *sharp front adaptivity*, operates with the absolute value of the function gradient on a grid element. The other, so-called *smooth adaptivity*, is based on the absolute values of the second derivatives of the solution. In practice, a linear (resp. quadratic) interpolation is used to approximate the gradient (resp. the second derivatives) of the solution on each grid element; we discuss the details below.

Since, at the initial moment, the exact solution is known (the function $u_0$), we can interpolate it on every grid element, compute the corresponding signal values and generate the initial distribution of particles by the following simple algorithm.

**Algorithm for initial-step adaptivity**

Let $G_i$ be the signal value of the solution function on the $i$-th grid element and $G_{mean}$ and $G_{max}$ respectively the mean and the maximum of the considered signal value on the whole grid. Add $N_{add}$ new points to the system of particles on the $i$-th element using the rule:

$$N_{add} = \begin{cases} 0\,, \text{ if } G_i \le G_{mean}, \\ N_{max}\,, \text{ if } G_i \equiv G_{max}, \\ \varphi(G_i), \text{ otherwise}\,, \end{cases} \tag{5}$$

where $N_{max}$ is a user-defined number, normally between 5 and 10, $\varphi$ is a continuous function monotonically increasing from 0 up to $N_{max}$ within the interval $[G_{mean}, G_{max}]$. In our numerical tests, the linear function $\varphi$ was always used.

Now we address the computation of the signal values $G_i$ for the $i$-th grid element. As mentioned above, the indicator for the "sharp" adaptivity is computed as the absolute

value of the gradient of the linearly interpolated initial function $u_0$. Since the gradient of a linear interpolation is a constant vector on each triangle of the grid, the absolute value of it yields a unique value of the "sharp" indicator on the considered triangle.

For the computation of the "smooth" indicator we tested two different methods. In the first one, we construct a quadratic ($\mathbb{P}_2$ in the finite element terminology, see, e.g., [10]) interpolation of the solution function on each triangle and then compute the maximum of the absolute values of the second derivatives of the interpolation. We call this method "the 6-point interpolation", since six points, namely the triangle's vertices and midpoints of its edges, are used in the interpolation procedure. The second method is based on the directional derivatives along the medians of the triangle. First, a one-dimensional quadratic polynomial depending on the distance measured along the median is constructed by the 1D interpolation on the median and, then, differentiated twice; next, we compute the maximum of the absolute values of such directional second derivatives for two arbitrary medians of the triangle. This maximum brings us the corresponding "smooth" indicator $G_i$ on the $i$-th grid element.

To demonstrate the performance of the adaptivity indicators we consider a Gaussian hill

$$u_0(\mathbf{x}) = \frac{1}{\sigma\pi\sqrt{2}} \exp\left(-\frac{(x_1-a)^2+(x_2-b)^2}{2\sigma^2}\right), \tag{6}$$

where $\mathbf{x} = (x_1, x_2)$, $(a, b)$ and $\sigma$ are the center and standard deviation, respectively. The parameters are chosen as $(a, b) = (1, 1)$ and $\sigma = 0.3$. The plot of the Gaussian hill on the domain $\Omega = [0, 2] \times [0, 2]$ is presented in Figure 1.

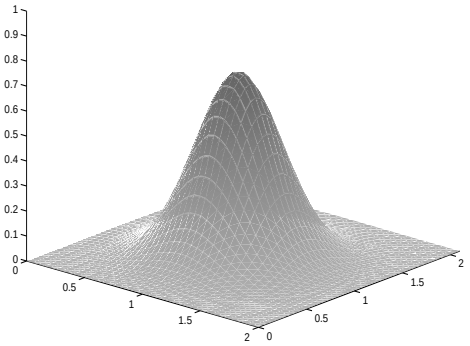



Figure 1: Gaussian hill with the center $(1, 1)$ and deviation $\sigma = 0.3$.

Consider now the initial set of particles consisting of 547 points, see Figure 2(a). The "sharp" adaptivity introduces new particles in the region of hillsides, that is demonstrated in Figure 2(b). Applying this kind of adaptivity we obtain 2251 points in the particle set. The effect of "smooth" adaptivities based on 6-points interpolation and directional derivatives can be seen in Figures 2(c),(d). Given the same parameters ($N_{max} = 10$, the initial number of particles equals 547) the first procedure brings more additional points than the second one: the final number is 1330 for the 6-point interpolation, and 1215 for the directional derivatives based adaptivity. To make a comparison we perform adaptivity using the maximums of the exact absolute values of the second derivatives for the Gaussian hill function, see Figure 2(e). This "exact smooth" adaptivity results in the set of 1736 particles. Although the 6-points interpolation is more exact than the one based on directional derivatives, the latter is computationally cheaper. At the same time, both techniques

adapt the top of the hill, i.e. the region of the maximal error for linear interpolation. In Figure 2(f) the example of joint action of "sharp" and "smooth" adaptivities is given; here the number of particles is 2483, which shows that the application of the combined "sharp + smooth" adaptivity is not much more expensive than the use of the "sharp" adaptivity alone. It is worth noting that, for the combined adaptivity, the adaptivity algorithm must be applied with "sharp" and "smooth" indicators simultaneously; every triangle receives $N_{add}$ new particles, where $N_{add}$ is the maximum of the particle numbers that would have been added by each of the adaptivity procedures.

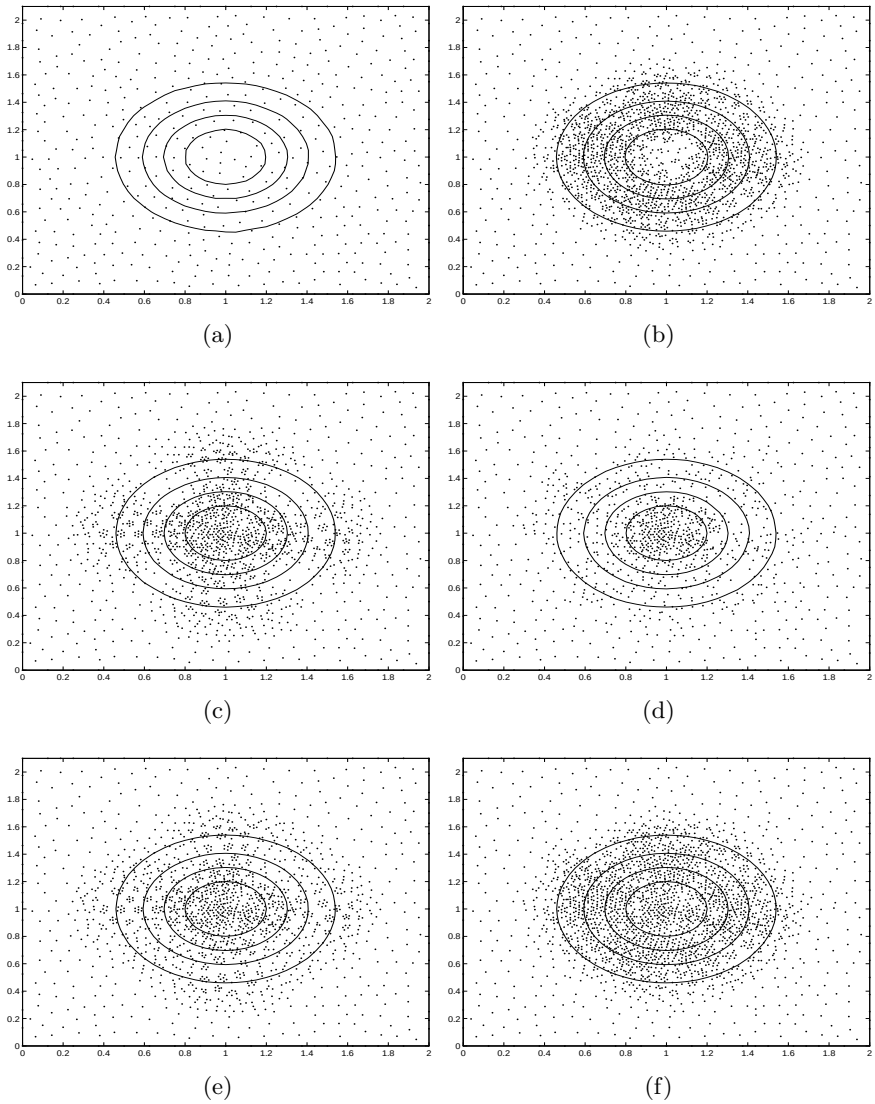



Figure 2: Particle distribution for the Gaussian hill (solid lines are the isolines of the Gaussian surface): (a) – initial distribution; (b) – sharp front adaptivity; (c) – smooth adaptivity based on 6-points interpolation; (d) – smooth adaptivity based on directional derivatives; (e) – smooth adaptivity based on exact 2nd derivatives; (f) – "sharp + smooth" adaptivity.

The above adaptivity is implemented as the particle set is initialized. Thus, at the end of this step, the particle system includes (some) grid-nodes and the set of points added by the adaptivity procedure. The type of the adaptivity (smooth or sharp) depends on the behavior of the initial function $u_0$; in general, the usage of combination of both adaptivity procedures can be recommended.

**Inflow adaptivity**

In addition to initial adaptivity, the particle system is subjected to inflow adaptivity during the solution of the problem. Due to the convective transport of the particle system a region of the domain $\Omega$ lacks the particles after each time step. This region is adjacent to the inflow part of the domain boundary and certainly requires an adaptive supply of new particles. The nodes and corresponding elements of the grid in this region form the base for creation of new inflow particles: the nodes lying on the inflow boundary receive the values of the boundary function $u_{in}$, and the elements are adapted similarly to the initialization step discussed above. Thus, inflow adaptivity that consists of "sharp front" and "smooth" variants always adds new points to the system of particles.

As an alternative to the adaptation procedure discussed above, we also used another algorithm for the inflow adaptivity; this algorithm delivered slightly better results in some cases. We illustrate it with an example of a one-dimensional problem. Here, the inlet portion of the domain is a point $p_0$ that, after each time step, generates a new inflow particle with the corresponding value of the boundary function $u_{in}$. To create an appropriate particle distribution between this new particle and its closest one, $p_1$, the following adaptive algorithm is used. Using the known function values at the points $p_0$ and $p_1$ we linearly interpolate on the interval $[p_0, p_1]$ and compute the signal value, i.e. the absolute value of the interpolation's gradient. If the signal value exceeds the threshold $1/h$, where $h$ is the distance between $p_0$ and $p_1$, then a new point $p_{1/2}$ is introduced in the middle of the interval $[p_0, p_1]$. The particle $p_{1/2}$ is assigned the value of the boundary function $u_{in}$ with the corresponding time-shift. Then, this procedure is repeated for intervals $[p_0, p_{1/2}]$ and $[p_{1/2}, p_1]$ and so on, until the length of intervals becomes very small or the signal value becomes less than the reciprocal of the interval length. Usually, only 5–7 iterative steps of this adaptivity are needed, but the ultimate number of steps definitely depends on the inflow boundary condition. The "smooth" adaptivity based on the directional derivative can be implemented in a very similar manner, starting with the three points $p_0$, $p_{1/2}$, $p_1$ to obtain a quadratic interpolation.

The result of the "sharp front" inflow adaptivity is demonstrated in Figure 3 for periodic inflow of a wave-packet function to be considered in test 2 of the section devoted to numerical examples.

### 3.2 Movement of the particles

On the initialization step, all the particles are assigned the values of the initial-condition function $u_0$ at the respective locations. It is important to note that these values are *exact*, because the initial condition $u_0$ is the given data of the problem. The particles must carry these values along the characteristics (trajectories).

Denote by $\mathbf{X} := \mathbf{X}(\mathbf{X_0}, t)$ the set of particle coordinates at time $t$, where $\mathbf{X_0}$ is the initial particle distribution. To find the new particle positions one has to solve equation (4) on the time-interval $[t_{n-1}, t_n]$, $n = 1, 2, ...$, with the initial condition at moment $t_{n-1}$

$$\mathbf{X}(\mathbf{X_0}, t_{n-1}) \,=\, \mathbf{X}^{(n-1)}\,, \tag{7}$$

where $\mathbf{X}^{(n-1)}$ is the vector of the coordinates of the particles computed on the previous time step ($\mathbf{X}^{(0)} \equiv \mathbf{X_0}$).

The Cauchy problem (4), (7) can be solved by any ODE solver, e.g. by the explicit Runge–Kutta method. The order of the method is usually dictated by accuracy considerations. It is worth noticing that the particles are completely independent of each other, and, hence, the motion of each particle can be traced with its own accuracy. For instance,

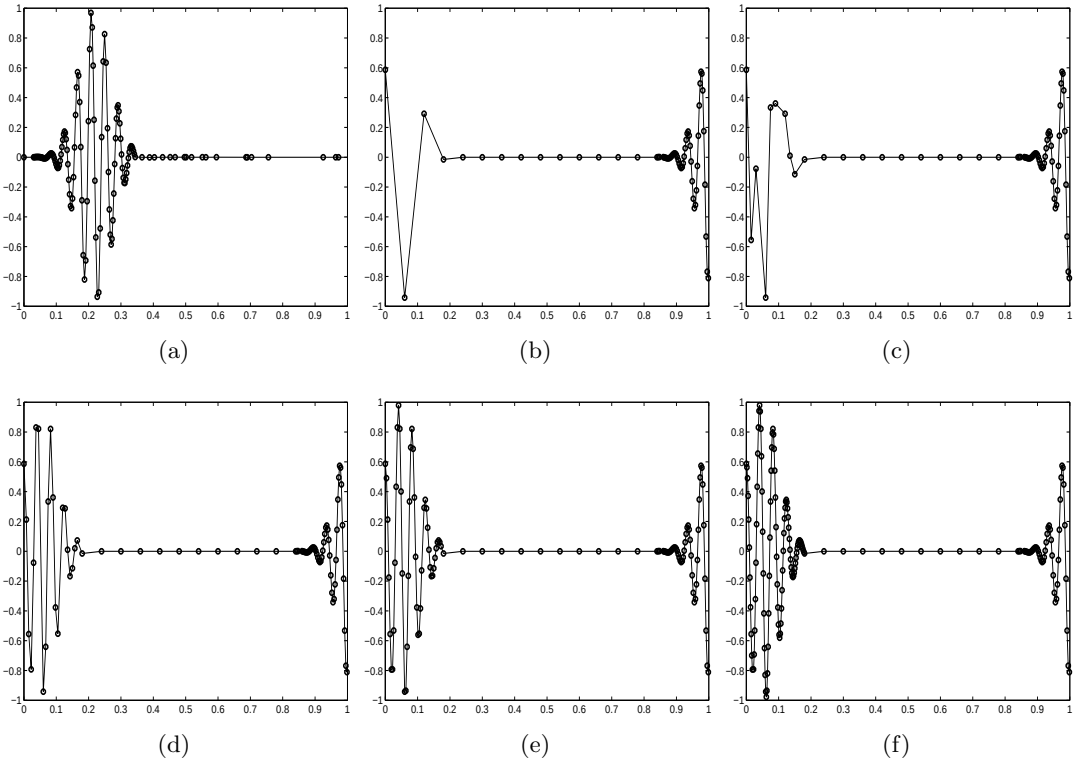


Figure 3: Particle set for the wave-packet function: (a) – initial particle distribution; (b) – inflow boundary condition without adaptivity; (c) – one step of inflow adaptivity; (d) – two steps of inflow adaptivity; (e) – five steps of inflow adaptivity; (f) – result of inflow adaptivity.

the transport in the regions with high solution-gradients or rapidly changing velocity field may require a better temporal accuracy than on the rest of the domain.

**Remark.** It is well known that any small volume formed by the particles passively convected by an *incompressible* flow is conserved. Thus, one does not have to worry about a clustering or rarefaction of particles in some regions of $\Omega$ due to particles' movement.

### 3.3 Projection onto the grid

In the PTM algorithm the projection is based on linear interpolation and is performed sequentially for each grid node $p$. This is done by finding the triangle that contains $p$ and is formed by three particles closest to the node (the "minimal triangle" for node $p$); then, the value at $p$ is straightforwardly computed by virtue of the linear interpolation on the found triangle. In order to have an efficient algorithm, it is necessary to optimize the search for the minimal triangle. To achieve this, we create a virtual Cartesian grid that fully covers the domain $\Omega$ (the so-called Global Discrete Coordinate System (GDCS)) and is composed of square cells $h \times h$, where $h$ is the width of our fixed grid, see Figure 4.

The location of any point in $\Omega$ with respect to GDCS can be found simply via integer division of the point's coordinates by $h$; the two resulting integers, the row and column numbers of the corresponding cell in the 2D array of GDCS, are the "global discrete coordinates" of the point. Based on these coordinates, every particle (namely, its number) is added to the list of the particles contained in the particular cell of GDCS. This information on the cell distribution of the particles is refreshed and saved after each time step. Thus, for a grid node $p$, all the particles lying in its cell and in the adjacent ones

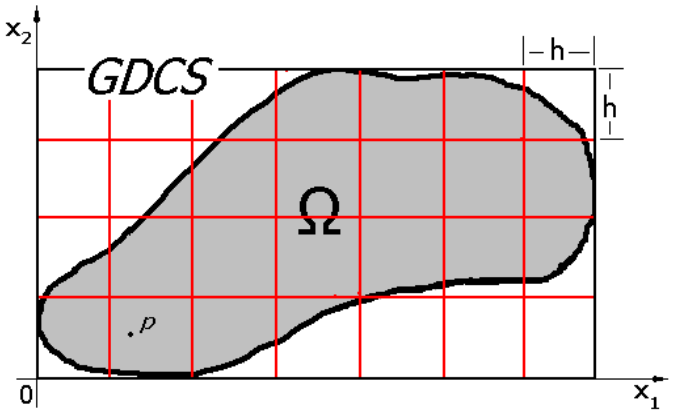



Figure 4: Global Discrete Coordinate System for the domain Ω.

can be found very fast; the search for the particles forming the minimal triangle becomes, thus, restricted to these particles only. It must be pointed out that for one- and three-dimensional space "minimal interval" and "minimal tetrahedron" are respectively needed. Obviously, a cell of GDCS is an interval in one-dimensional case and a parallelepiped in three-dimensional one.

Creating additional global coordinates allows us to achieve a genuine inter-independence of the particles, as far as there is no need of their ordered storage. Since for each grid node its particle vicinity is known in GDCS in advance, the complexity of projection procedure is $O(n)$, where $n$ is the number of grid nodes (here we have assumed that each GDCS cell contains $O(1)$ particles, which is, in average, true, because the adaptivity adds $O(1)$ new particles on each grid element of diameter $h$). Taking into account movement of the particles, time of the PTM execution can be estimated as $O(n) + O(N)$, where $N$ is the number of particles. Thus, PTM has optimal, linear (with respect to the number of particles and nodes) complexity, which is also demonstrated in the numerical tests below.

By construction, PTM is a combination of exact transport and projection. Since the former does not change the solution value and the latter operates with a convex combination of exact transported values in the triangle's vertices, maxima increase and minima decrease are precluded. Moreover, new local extrema cannot appear either. Thus, PTM satisfies the Local Extremum Diminishing (LED) criterion (see [29]). Owing to the linear interpolation, we can expect the 2nd-order spatial accuracy for the smooth solutions in the maximum norm. The combination of these properties and the adaptive mechanism for sharp resolution of solution singularities make PTM a high-resolution scheme, which is clearly confirmed by the numerical results.

**Remark.** The linear interpolation on a "minimal triangle" is, clearly, especially advantageous for assigning the values to the grid nodes located near the solution discontinuities: the local character of the interpolation allows one to maintain a good resolution of the singularities.

## 4 Numerical results

In this section we consider four problems: the step-function transport, the wave-packet convection, the rigid-body rotation of a slotted cylinder and the transport of smooth *sine*-surface. All experiments have been conducted on a LINUX PC (Intel Pentium IV 2.5 GHz).

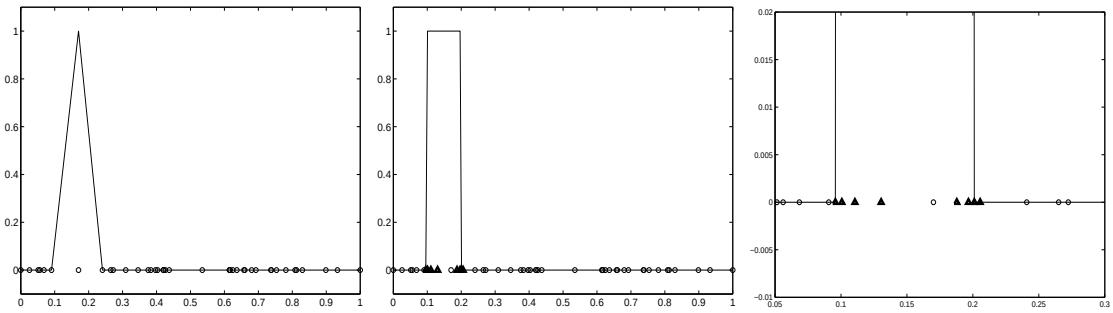
### 4.1 Experiment 1: step-function transport

Consider the problem of step-function transport on the domain $\Omega = [0, 1]$, given the velocity $v = 0.3$ and the initial condition

$$u_0(x) = \begin{cases} 1\,, \text{ if } 0.1 \leq x \leq 0.2, \\ 0\,, \text{ otherwise.} \end{cases} \tag{8}$$

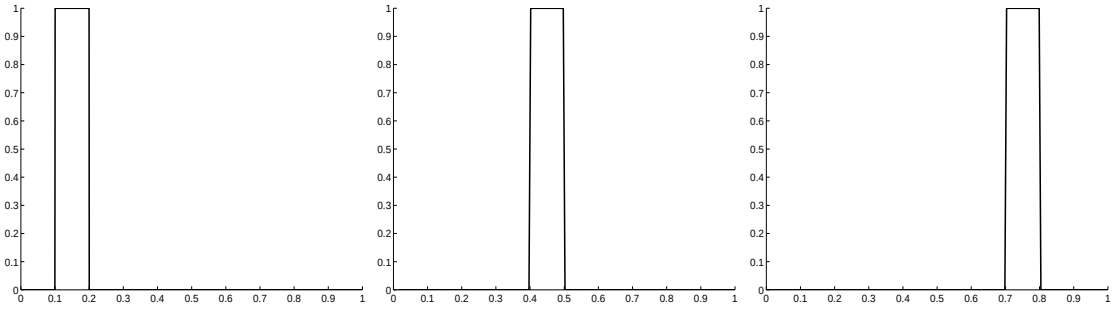
A zero inflow boundary condition is set at the left end of the domain, $x = 0$. We simulate the transport on the time interval $[0, 2]$ with the time step $\triangle t = 0.02$.

The starting grid is a collection of 40 points randomly distributed on $\Omega$, see Figure 5(a). After the sharp front adaptive initialization with $N_{max} = 1$ the system of particles numbers 48 points, Figure 5(b); eight additional points are introduced in the vicinity of discontinuities, Figure 5(c).



(a) (b) (c)

Figure 5: Particle set for the problem of step-function transport: (a) – initial distribution; (b) – distribution after adaptivity; (c) – zoom of the distribution in vicinity of discontinuities (triangles denote the new particles).



(a) (b) (c)

Figure 6: Step-function transport: (a) – initial condition; (b) – solution at $t = 1.0$; (c) – solution at $t = 2.0$.

It takes about 9.0e-02 seconds of CPU time to execute 100 time steps of PTM. Figure 6 shows the initial step-function and the solution after 50 and 100 time steps. The projection onto the uniform 100-nodes grid is based on linear interpolation between two nearest particles for each grid node.

### 4.2 Experiment 2: wave-packet transport

Next, consider the solution of equation (1) on $\Omega = [0, 1]$ for the wave-packet function with the periodic inflow boundary condition at the left end of the domain. The convective velocity $v = 0.3$ and the initial condition is given by

$$u_0(x) = \exp(-200(x - 0.5)^2) \sin(150x) \qquad \text{in } \Omega\,. \tag{9}$$

The transport of wave-packet is simulated on the time interval $[0, 6]$ with the time step $\triangle t = 0.02$.

The initial wave and the solution after 86 and 300 time steps computed by PTM are depicted in Figure 7. The system of 120 particles was adaptively generated starting from a uniform grid with 30 nodes. Here, the sharp front adaptivity is used with $N_{max} = 1$; this procedure is also implemented after each time step as the inflow adaptivity. The projection onto the uniform 120-nodes grid is based on linear interpolation between two closest particles for each grid node.

For a comparison, a high-resolution finite volume method of [36] is considered. We use its realization in the package CLAWPACK[1]. The method utilizes a flux-limiter for eliminating the numerical oscillations; among different limiters – minmod, superbee, van Leer and monotonized centered (MC) – the latter has been selected, since it yields the best results for this particular problem and makes the scheme total-variation-diminishing (TVD). In this test, a uniform grid is employed again, with the number of nodes equal to the number of the PTM particles, i.e. 120. It is seen in Figure 8 that the flux-limiter exerts a hard damping on the solution, suppressing all the physical sinusoidal oscillations. A more realistic wave-packet can be obtained on a finer grid with 500 nodes, see Figure 9. It is worth noticing that CPU time needed to execute CLAWPACK was 4.1e-01 sec. for the 120-nodes grid and 1.8e+00 sec. for the 500-nodes grid. PTM needs only 1.2e-01 sec. to bring the results presented in Figure 7.

In this case, with a constant convective velocity, the ODE (particle movement) step of PTM is trivial. What plays the main role is the sharp front adaptive placement of the particles in the regions of high gradient, this procedure taking place on each inflow step. Such an adaptivity also helps to keep the linear interpolation step accurate.

### 4.3 Experiment 3: slotted-cylinder rotation

Consider a rigid-body rotation of a slotted cylinder. Here, $\Omega = [0, 3] \times [0, 4] \subset \mathbb{R}^2$ and the initial condition is given by

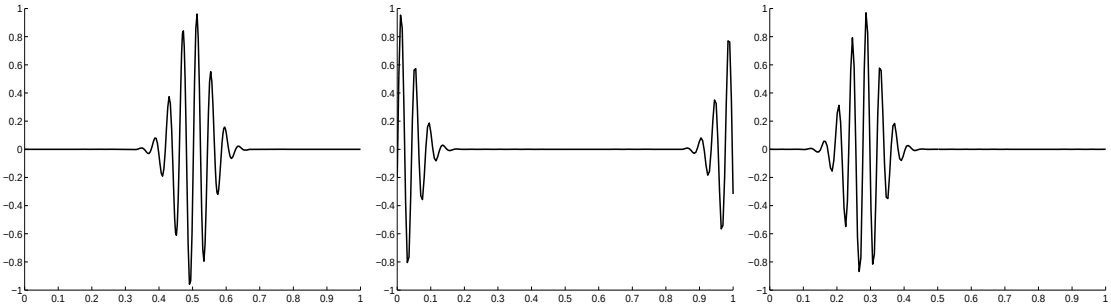
$$u_0(\mathbf{x}) = \begin{cases} 1\,, \text{ if } \mathbf{x} \in D, \\ \\ 0\,, \text{ otherwise,} \end{cases} \tag{10}$$

where $D \subset \Omega$ is a slotted disk with the radius $r = 0.6$, centered at $(1.5, 2.0)$ and having the slot of the width $2r/3$ and the length $3r/2$. The slotted cylinder is rotated by the steady flow field $\mathbf{v}(\mathbf{x}) = \{x_2 - 2.0,\ -x_1 - 1.5\}$, where $\mathbf{x} = (x_1, x_2)$.

An unstructured Delaunay triangulation was generated by the package Triangle[2]; the starting grid has 1398 nodes, see Figure 10(a). Using the sharp front adaptivity with $N_{max} = 10$, the system of particles is concentrated in the region of high gradient yielding 2254 points, Figure 10(b). The evolution of the particles is computed by the 3rd-order Runge–Kutta method with the time step $\triangle t = \pi/10 = 0.3142$.

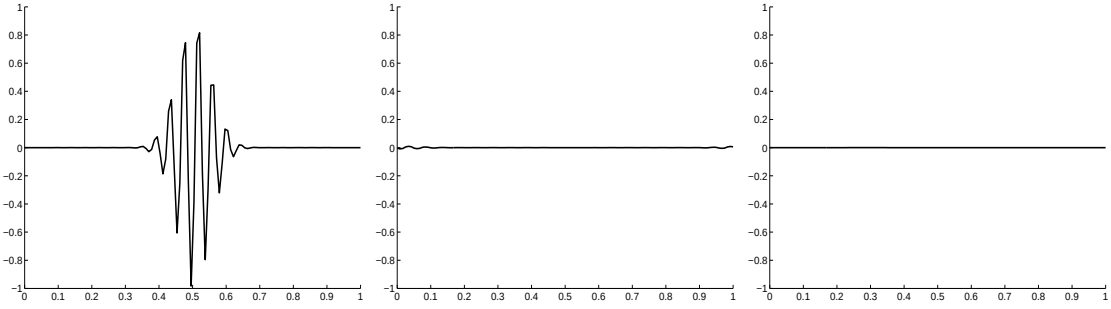
[1] LeVeque R.J., Conservation Law Package, http://www.amath.washington.edu/∼claw/

[2] Triangle mesh generator, http://www-2.cs.cmu.edu/∼quake/triangle.html



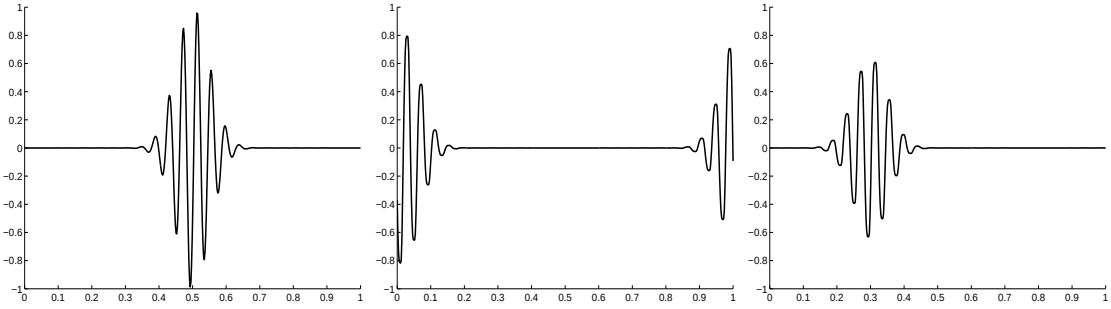
(a) (b) (c)

Figure 7: Wave-packet transport. PTM, 120 particles: (a) – initial condition; (b) – solution at $t = 1.72$; (c) – solution at $t = 6.0$.



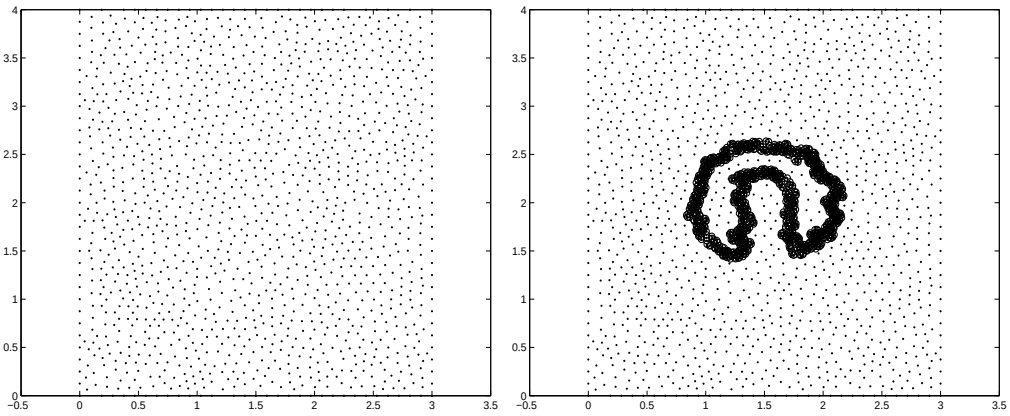
(a) (b) (c)

Figure 8: Wave-packet transport. CLAWPACK, 120 nodes: (a) – initial condition; (b) – solution at $t = 1.72$; (c) – solution at $t = 6.0$.



(a) (b) (c)

Figure 9: Wave-packet transport. CLAWPACK, 500 nodes: (a) – initial condition; (b) – solution at $t = 1.72$; (c) – solution at $t = 6.0$.



(a) (b)

Figure 10: Particle set for the slotted cylinder: (a) – initial set, 1398 points; (b) – adaptive set, 2254 points.

The initial condition and the solution after two full rotations projected onto the 9454-nodes unstructured grid are shown in Figure 11. It is interesting to note that, given these problem parameters, the equivalent Courant number $Co = \triangle t \|\mathbf{v}\| \sqrt{N}$, where $N$ is the total number of particles, equals 7.46 (i.e. this would be the value of the Courant number, if we solved the problem by some standard method with the same $\triangle t$ and on a uniform mesh with $N$ nodes).

To test the complexity of the method two batches of computational experiments have been conducted. In each set of experiments, one full revolution of the cylinder has been computed in 20 time steps using the 3rd-order Runge–Kutta method. The first set of simulations has been carried out to reveal the time dependence on the number of particles given two fixed numbers of grid nodes, namely 1398 and 1937. During the second set of tests we have varied the number of grid nodes keeping the same number of particles, namely 1139 and 2224. Tables 1 and 2 demonstrate the overall execution time of PTM and the time of the projection procedure, which has been implemented after each time step, with respect to the number of particles and the number of nodes, respectively. It is clear that the projection constitutes more than 80% of the total computational time; thus, for instance, an increase of the order of accuracy of the Runge–Kutta scheme would not result in a significant rise of the total computational cost. We have, however, found that in all numerical experiments the 3rd-order accurate Runge–Kutta method was sufficient.

The dependencies of CPU time on the number of grid nodes and particles are depicted in Figure 12 (the time is multiplied by 1000 to balance the scales on the axes). Here, there are several issues to be addressed. At first, it is obvious that the computing time depends linearly on both the particle and the node numbers. Secondly, one can notice that an increase of the number of particles results in a much smaller rise of the computational time than that due to an increase of the number of nodes. Hence, the particle adaptivity (particularly, an addition of new particles to improve the resolution in some subdomains) is a computationally cheap procedure. Finally, our goal was to show that the PTM complexity is really of the form $an + bN$ with some constants $a$ and $b$ and does not contain a term like $n^\alpha N^\beta$ (with some positive $\alpha$ and $\beta$; here, as before, $n$ is the number of grid nodes, $N$ is the number of particles). Since for two different $n$ (respectively, $N$) we obtain the lines with the same slope, one can infer that the PTM complexity has, in fact, the form $O(n) + O(N)$ and is, thus, optimal.

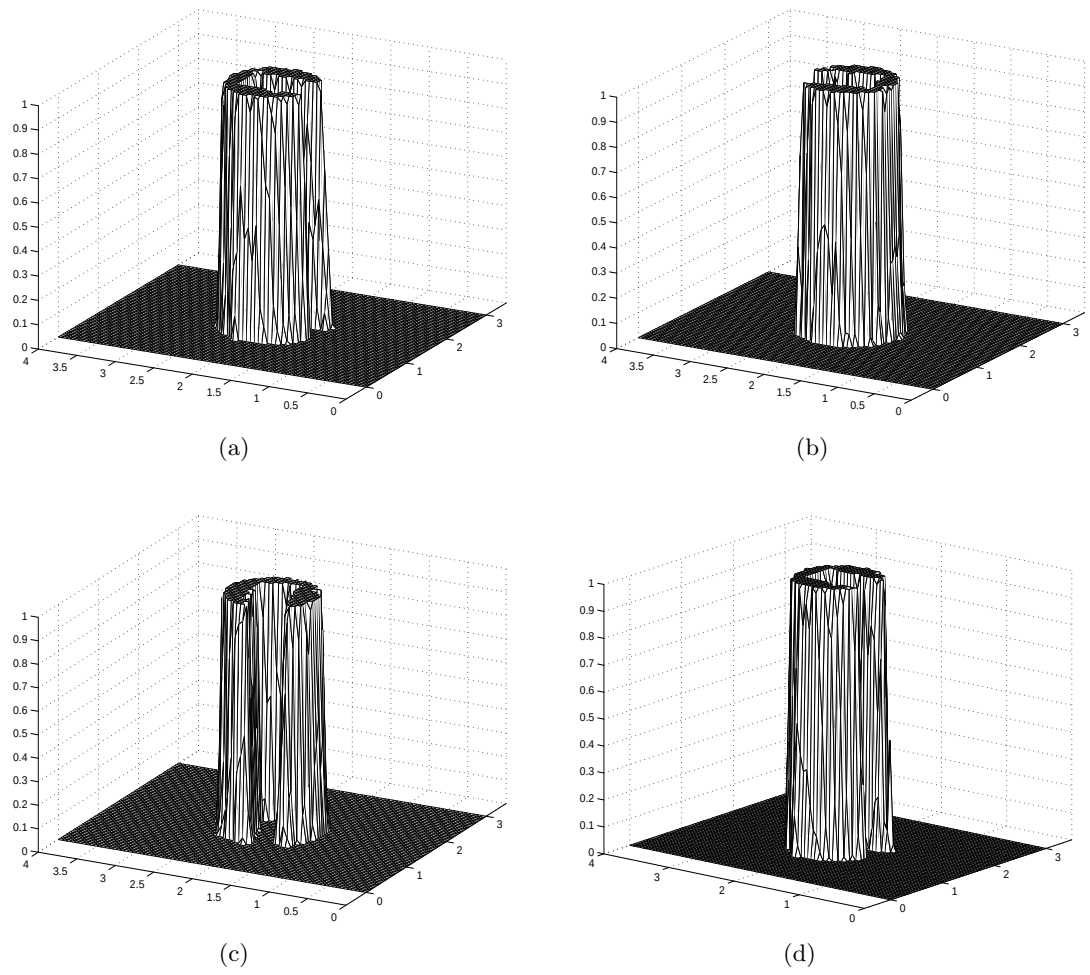


Figure 11: Two full rotations of the slotted cylinder, 2254 particles, $t \in [0, 4\pi]$, 40 time steps: (a) – initial condition; (b) – solution at $t = \pi$; (c) – solution at $t = 5\pi/2$; (d) – solution at $t = 4\pi$.

### 4.4 Experiment 4: transport of a smooth function

In order to check the pointwise accuracy of PTM, we consider the problem proposed in [50]. Here, the domain $\Omega = [0, 2] \times [0, 2]$, the velocity $\mathbf{v} = \{1, 1\}$ and the initial condition is given by

$$u_0(\mathbf{x}) = \sin(\pi(x_1 + x_2)). \tag{11}$$

It is assumed that the inflow boundary condition is periodic. The initial function is shown in Figure 13.

The goal of the simulation is to compute the solution at the time $t = 1.0$. Although in the case of a constant convective velocity PTM does not have any restrictions on the time-step size and the accuracy of the ODE solver, we use, for the sake of consistency with [50], the 3rd-order Runge–Kutta scheme and $\triangle t = 0.1$. However, we take the advantage of our approach and project the solution onto the fixed grid only once, after the final time step. An unstructured grid with 547 nodes is used for the projection.

The starting set of 2111 particles is depicted in Figure 14(a). For the problem at hand we used two kinds of adaptivity. In Figures 14(b),(c) one can see the effects of the sharp front and smooth adaptivities, respectively. The constant $N_{max}$ was equal to

Table 1: Computational time, sec., for one full revolution of a slotted cylinder (in dependence on the number of particles).

| | grid fixed | | | |
|---|---|---|---|---|
| | 1398 nodes | | 1937 nodes | |
| #particles | projection | PTM | projection | PTM |
| 2224 | 1.21 | 1.33 | 1.67 | 1.93 |
| 4363 | 1.16 | 1.4 | 1.79 | 1.98 |
| 5515 | 1.25 | 1.53 | 1.9 | 2.02 |
| 7717 | 1.41 | 1.82 | 1.96 | 2.24 |
| 9510 | 1.49 | 1.89 | 2.02 | 2.54 |

Table 2: Computational time, sec., for one full revolution of a slotted cylinder (in dependence on the number of nodes).

| | particle set fixed | | | |
|---|---|---|---|---|
| | 1139 points | | 2224 points | |
| #nodes | projection | PTM | projection | PTM |
| 1398 | 0.83 | 1.03 | 1.21 | 1.33 |
| 1937 | 1.16 | 1.29 | 1.67 | 1.93 |
| 3190 | 2.96 | 3.03 | 3.36 | 3.5 |
| 4785 | 4.37 | 4.46 | 4.68 | 5.25 |
| 9454 | 8.59 | 8.67 | 8.9 | 9.2 |

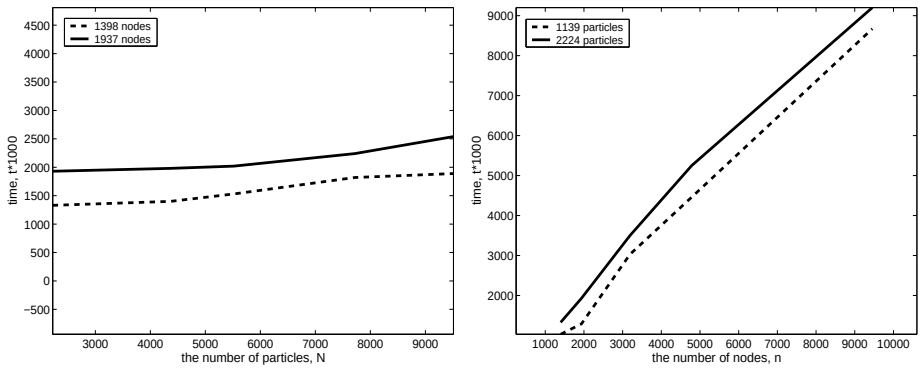
5 in both procedures; the sharp front adaptivity resulted in 10323 particles, the smooth adaptivity based on directional derivatives in 4746 particles. The particle system obtained by the combination of both adaptivity procedures consisted of 12491 points; it is shown in Figure 14(d).

To estimate the order of method's accuracy we measure the error in the $l_\infty$ norm:

$$\|u^{ex} - u^{ap}\| = \max_{i=1,n} \mid u_i^{ex} - u_i^{ap} \mid,$$

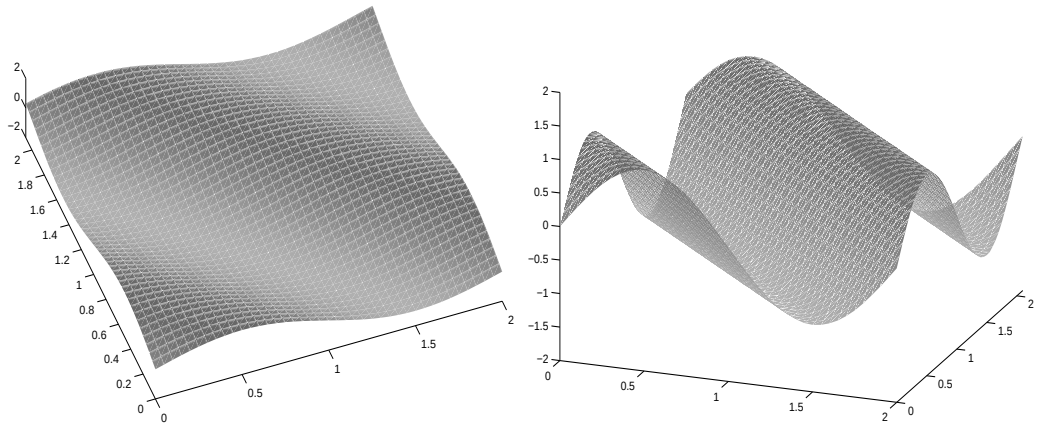
where $u^{ex}$ and $u^{ap}$ are the exact and the approximate solutions of the problem, respectively, $n$ is the number of grid nodes. Table 3 demonstrates the error and CPU time of PTM in comparison with those of the 2nd-order discontinuous Galerkin (DG) method considered in [50]. Our computational experiments have been conducted with different sets of particles, the number of them being always approximately equal to the number of grid nodes (i.e. the number of degrees of freedom) in the DG method. We have also to point out that in Table 3 we adopted the same notation for the number of grid nodes as in Table V of [50]; it does not, however, mean that the grids in [50] were uniform but is just a way of specifying (approximately) the number of grid nodes.

The results of Table 3 clearly indicate that both methods achieve the expected, second order accuracy in the $l_\infty$ norm with respect to $h = 1/\sqrt{N}$, where $N$ is the total number of




(a) (b)

Figure 12: Execution time of PTM for the slotted-cylinder rotation: (a) – in dependence on the number of particles; (b) – in dependence on the number of nodes.



(a) (b)

Figure 13: Initial *sine*-surface: (a) – view from above; (b) – side view.

degrees of freedom (see also in Figure 15 the logarithmic plot of the error in dependence on the number of particles). However, while both methods have nearly the same magnitude of the error, the PTM execution time is more than 30 times shorter than that of the DG method. This is rather remarkable, taking into account that the computations in [50] were carried out on a PC with almost the same configuration, and no code optimization has been used in our case.

# 5 Concluding remarks

In the paper, a novel explicit, unconditionally stable algorithm, the Particle Transport Method (PTM) for linear convection problems has been presented. The scheme is a mixture of the Lagrangian method of characteristics and the projection based on linear interpolation. Such a combination allows us to obtain a monotone method with linear computational complexity. PTM is a high-resolution method due to the spatial adaptivity for the moving particle system at the initialization stage and at the inflow boundary. The particle concept is the main issue of the method, and we have shown that the particle adaptivity is algorithmically simple and inexpensive. The transport of the particles along

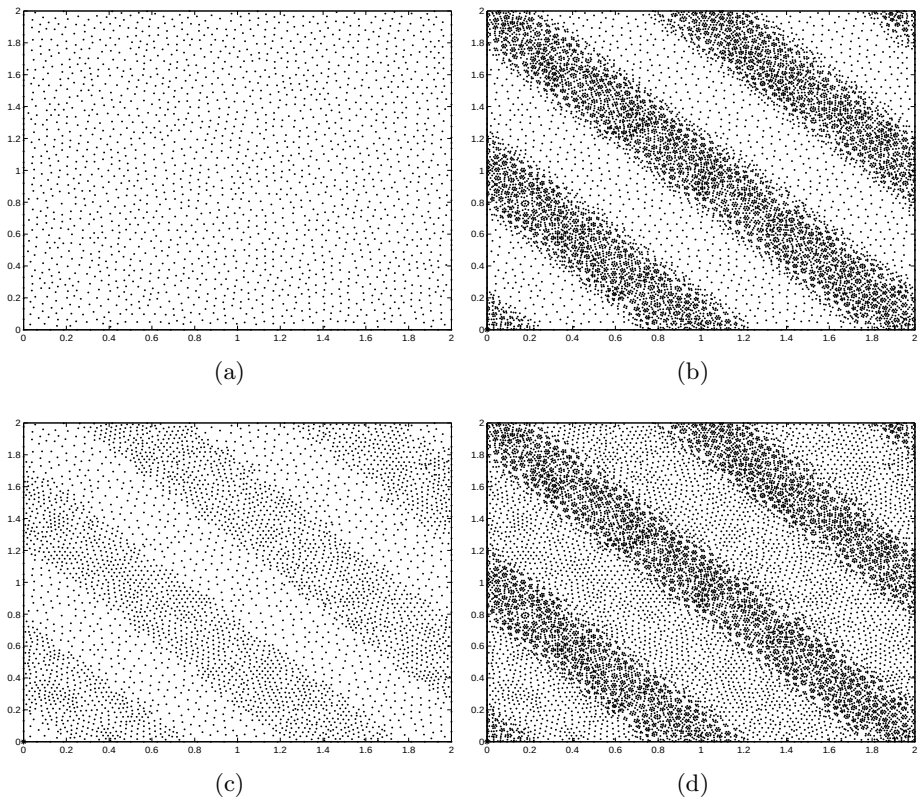



Figure 14: Particle set for the *sine*-surface: (a) – initial distribution; (b) – sharp front adaptivity; (c) – smooth adaptivity; (d) – sharp + smooth adaptivity.

the characteristics is performed by solving the first order Cauchy problem with the Runge–Kutta method. The order of the latter can be arbitrary, depending on the prescribed time step and the required accuracy; moreover, each particle can be traced with its own accuracy, thus, giving rise to the possibility of the temporal adaptivity. PTM has the 2nd-order accuracy in space but can be significantly faster than the grid-based methods of the same order, as the numerical comparisons show.

Our current research is focused on the combination of the proposed scheme with the fixed-grid finite element method for the diffusion problem within the operator-splitting framework as well as on the extension of the approach to the convection-reaction problems.

We also suppose that the method can be very efficient for the problems on complicated domains, especially in three-dimensional case, where the grid-based methods face apparent difficulties. The possibility of an almost unlimited particle-resolution might be important in such problems due to the presence of numerous small geometrical details that, however, cannot be neglected. The problems of this type often arise, e.g., in diverse biological and medical applications. The work in this direction is underway, and the results will be reported soon.

**Acknowledgements:** This work was supported by research project no.31335 (MANDI) of the Finnish Technology Development Agency.

Table 3: Error and computational time for smooth-function transport by DG method ([50]) and PTM on unstructured meshes.

| **DG** | | | **PTM** | | |
|---|---|---|---|---|---|
| #nodes | error, $l_\infty$ | time, sec. | #particles | error, $l_\infty$ | time, sec. |
| $10 \times 10$ | 6.05e-02 | 3.77e-01 | 145 | 9.3e-02 | 1.0e-01 |
| $20 \times 20$ | 1.64e-02 | 3.26e+00 | 472 | 3.2e-02 | 2.6e-01 |
| $40 \times 40$ | 4.17e-03 | 2.65e+01 | 1773 | 3.7e-03 | 9.9e-01 |
| $80 \times 80$ | 1.05e-03 | 2.13e+02 | 7073 | 1.4e-03 | 6.1e+00 |
| $160 \times 160$ | 2.62e-04 | 1.75e+03 | 25304 | 2.3e-04 | 5.2e+01 |

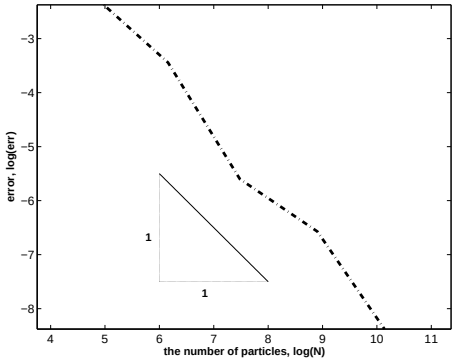



Figure 15: PTM error for *sine*-surface convection: the dependence of the $l_\infty$ norm on the number of particles, logarithmic scale.

# References


[1] Baumann C.E., Oden J.T. A discontinuous hp finite element method for convection-diffusion problems. *Comp. Meth. Appl. Mech. Engrg.*, **175** (1999), 311–341.

[2] Behrens J., Iske A. Grid-free adaptive semi-Lagrangian advection using radial basis functions. *Computers and Math. with Appl.* **43** (2002), 319–327.

[3] Belytshko T., Krongauz Y., Organ D., Fleming M., Krysl P. Meshless methods: an overview and recent developments. *Comp. Meth. Appl. Mech. Engrg.* **139** (1996), 3–37.

[4] Binning P., Celia M.A. A forward particle tracking Eulerian-Lagrangian localized adjoint method for solution of the contaminant transport equation in three dimensions. *Advances in Water Resources*, **25** (2002), 147–157.

[5] Boris J.P., Book D.L. Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. *J. Comp. Phys.* **11** (1973), 38–69.

[6] Brackbill J.U. Particle methods. *Int. J. Numer. Meth. Fluids* **47**, (2005), 693–705.

[7] Breitkopf P., Huerta A. *Meshfree and Particle Based Approaches in Computational Mechanics.* Kogan Page Science, 2004.

[8] Brooks A.N., Hughes T.J.R. Stream-line Upwind/Petrov-Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engrg.* **32** (1982), 199–259.

[9] Casper J., Atkins H.L. A finite-volume high-order ENO scheme for two-dimensional hyperbolic systems. *J. Comp. Phys.* **106** (1993), 62–76.

[10] Ciarlet P.G. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.

[11] Cockburn B., Shu C.-W. The Runge–Kutta discontinuous Galerkin method for conservation laws. I: Multidimensional systems. *J. Comp. Phys.* **141**, no.2, (1998), 199–224.

[12] Codina R. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Comp. Meth. Appl. Mech. Engrg.*, **188** (2000), 61–82.

[13] Crowley W.P. FLAG: A free-Lagrangian method for numerical simulating hydrodynamics flows in two dimensions. In *Lecture Notes in Physics* **8**, Springer, New York, 1971, pp. 37–43.

[14] Donea J. A Taylor-Galerkin method for convective transport problems. *J. Numer. Meth. Engrg.*, **20** (1984), 101–120.

[15] Donea J., Quartapelle L. An introduction to finite element methods for transient advection problems. *Comp. Meth. Appl. Mech. Engrg.*, **95** (1992), 169–203.

[16] Donea J., Huerta A. *Finite Element Methods for Flow Problems.* Wiley, Chichester, 2003.

[17] Ewing R.E., Wang H. Eulerian-Lagrangian localized adjoint methods for linear advection or advection-reaction equations and their convergence analysis. *Comput. Mech.* **12**, no. 1-2, (1993), 97–121.

[18] Finlayson B.A. *Numerical Methods for Problems with Moving Fronts.* Ravenna Park Publishing, Seattle, 1992.

[19] Hansbo P. A free-Lagrangian finite element method using space-time elements. *Comp. Meth. Appl. Mech. Engrg.* **188** (2000), 347–361.

[20] Harten A. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.* **49** (1983), 357–393.

[21] Harten A., Engquist B., Osher S., Chakravarthy S. Uniformly high order essentially non-oscillatory schemes, III. *J. Comp. Phys.* **71** (1987), 231–303.

[22] Healy R.W., Russell T.F. Solution of the advection-dispersion equation in two dimensions by a finite-volume Eulerian-Lagrangian localized adjoint method. *Advances in Water Resources*, **21**, no.1, (1998), 11–26.

[23] Hu F.Q., Hussaini M.Y., Rasetarinera P. An analysis of the discontinuous Galerkin method for wave propagation problems. *J. Comp. Phys.*, **151** (1999), no.2, 921–946.

[24] Huerta A., Fernández-Méndez S. Time accurate consistently stabilized mesh-free methods for convection dominated problem. *Int. J. Numer. Meth. Engrg.* **56** (2003), 1225–1242.

[25] Hughes T.J.R., Franca I.P., Mallet M. A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advection-diffusion System. *Comp. Meth. Appl. Mech. Engrg.* **63** (1987), 97–112.

[26] Hughes T.J.R., Mallet M. A new finite element formulation for computational fluid dynamics. III. The generalized streamline operator for multidimentional advective-diffusive Systems. *Comput. Methods Appl. Mech. Engrg.*, **58** (1986), 305–328.

[27] Idelsohn S.R., Oñate E., Del Pin F. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Numer. Meth. Engrg.* **61** (2004), 964–989.

[28] Iske A., Käser M. Conservative semi-Lagrangian advection on adaptive unstructured meshes. *Num. Meth. Partial Diff. Eq.* **20** (2004), 388–411.

[29] Jameson A. Positive schemes and shock modelling for compressible flows. *Int. J. Numer. Meth. Fluids* **20**, (1995), 743–776.

[30] Johnson C. A new approach to algorithms for convection problems which are based on exact transport + projection. *Comp. Meth. Appl. Mech. Engrg.* **100**, no.1, (1992), 45–62.

[31] Kaazempur-Mafrad M.R., Ethier C.R. An efficient characteristic Galerkin scheme for the advection equation in 3D. *Comp. Meth. Appl. Mech. Engrg.* **191** (2002), 5345–5363.

[32] Krivodonova L., Xin. J., Remacle J.-F., Chevaugeon N., Flaherty J.E. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.*, **48** (2004), 323–338.

[33] Kröner D. *Numerical Schemes for Conservation Laws.* Wiley and Teubner, 1996.

[34] Kuzmin D., Möller M. Algebraic flux correction I. Scalar conservation laws. In *Flux-Corrected Transport: Principles, Algorithms, and Applications.* (eds. D. Kuzmin, R. Löhner, S. Turek), Springer-Verlag, 2005, pp. 155–206.

[35] Kuzmin D., Möller M., Turek S. High-resolution FEM-FCT schemes for multidimensional conservation laws. *Comp. Meth. Appl. Mech. Engrg.* **193**, no. 45-47, (2004), 4915–4946.

[36] LeVeque R.J. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, Cambridge, 2003.

[37] Li S., Liu W.K. *Meshfree Particle Methods.* Springer, Berlin, 2004.

[38] Löhner R., Morgan K., Peraire J., Vahdati M. Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7**, (1987), 1093–1109.

[39] Löhner R. Robust vectorized search algorithms for interpolation on unstructured grid. *J. Comp. Phys.* **118** (1996), 380–387.

[40] Löhner R. *Applied CFD Techniques: An Introduction Based on FEM.* John Wiley and Sons, 2002.

[41] Marchuk G.I. Splitting and alternating direction methods. In *Handbook of Numerical Analysis I*, Noth-Holland, Amsterdam, 1990, pp. 197–462.

[42] Monaghan J.J. Simulating free surface flows with SPH. *J. Comp. Phys.* **110**, no.2, (1994), 399–406.

[43] Morton K.W., Priestley A., Süli E. Stability of the Lagrange-Galerkin method with non-exact integration. *Math. Model. Numer. Anal.* **22** (1988), 625–653.

[44] Nithiarasu P., Zienkiewicz O.C., Satya Say B.V.K., Morgan K., Codina R., Vázquez M. Shock capturing viscosities for the general fluid mechanics algorithm. *Int. J. Numer. Meth. Fluids.* **28** (1998), 1325–1353.

[45] Oliviera A., Fortunato A.B. Toward an oscillation-free, mass-conservative, Eulerian-Lagrangian transport method. *J. Comp. Phys.* **183** (2002), 142–164.

[46] Priestley A. Exact projection and the Lagrange-Galerkin method: a realistic alternative to quadrature. *J. Comp. Phys.* **112** (1994), 547–571.

[47] de Sampaio P.A.B., Lyra P.R.M., Morgan K., Weatherill N.P. Petrov-Galerkin solutions of the incompressible Navier-Stokes equations in primitive variable with adaptive remeshing. *Comp. Meth. Appl. Mech. Engrg* **106** (1993), 143–178.

[48] Serna S., Marquina A. Power ENO methods: a fifth-order accurate weighted power ENO method. *J. Comp. Phys.* **194** (2004), 632–658.

[49] Shu C.-W., Osher S. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comp. Phys.* **77** (1988), 439–471.

[50] Sun Y., Wang Z.J. Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids. *Int. J. Numer. Meth. Fluids*, **45** (2004), 819–838.

[51] Ying L., Zhang P. *Vortex methods.* Kluwer Academic Publishers, Dordrecht, 1997.