# Accurate, non-oscillatory, remeshing schemes for particle methods

Adrien Magni, Georges-Henri Cottet *

*Université de Grenoble and CNRS, Laboratoire Jean Kuntzmann, BP 53, 38041 Grenoble Cédex, France*

## ARTICLE INFO

## ABSTRACT

In this article we propose and validate new remeshing schemes for the simulation of transport equations by particle methods. Particle remeshing is a common way to control the regularity of the particle distribution which is necessary to guarantee the accuracy of particle methods in presence of strong strain in a flow. Using a grid-based analysis, we derive remeshing schemes that can be used in a consistent way at every time-step in a particle method. The schemes are obtained by local corrections of classical third order and fifth order interpolation kernels. The time-step to be used in the resulting push-and-remesh particle method is determined on the basis of rigorous bounds and can significantly exceed values obtained by CFL conditions in usual grid-based Eulerian methods. In addition, we extend the analysis of [5] to obtain TVD remeshing schemes that avoid oscillations of remeshing formulas near sharp variations of the solution. These methods are illustrated in several flow conditions in 1D, 2D and 3D.

© 2011 Elsevier Inc. All rights reserved.

## 1. Introduction

The accurate simulation of transport equations is a central issue in many problems in physics or engineering. The transported quantity can either be a passive scalar or vector (concentration, density, marker of a Lagrangian interface) or produce a feedback to the velocity vector and other quantities of interest (momentum or energy in fluid mechanics, charge density in plasma physics). The case of a Lagrangian interface, in particular inside a variable density fluid, is of particular interest. An interface can be just a marker indicating a free surface or interact with the fluid itself, for example in a curvature driven motion or in reacting fluids where reactions take place at the interface between different species.

The production of small scales is a typical feature of transport in many important applications e.g in turbulent, free surface or reactive flows. A great deal of work has therefore been devoted to the derivation of methods that are at the same time accurate, oscillation-free and conservative. In particular, recent works have been done where Lagrangian markers are used to improve or correct grid-based methods to capture interface dynamics in a conservative and accurate way. In [12] sub-grid particles are introduced near the interface to improve the accuracy of finite-difference level set methods. Vincent [25] describes a Volume-of-Fluid method for advection problems designed in the same spirit for general transport equations. These works take advantage of the *exact* Lagrangian solution to advection equation in order to limit the numerical diffusion of grid-based method and enforce mass conservation. In [15,10], on the other hand, the use of *regular* remeshed particle methods was advocated to ensure an efficient treatment of advection equations. By remeshed particles methods, we mean particle methods where particles are simply advected and remeshed on a regular grid through high order interpolation kernels.

Particle methods are Lagrangian techniques that have been designed for advection-dominated physical problems. Among the features that are generally acknowledged for these methods are the lack of numerical dissipation and the robustness due to the absence of Courant type stability condition. A well recognized drawback is the potential accuracy deterioration

---

\* Corresponding author.
   *E-mail address:* georges-henri.cottet@imag.fr (G.-H. Cottet).

resulting form high distortion in the flow. In particle methods macroscopic quantities are recovered by regularizing particle into overlapping cores, and the overlapping ratio should in principle increase as the particle spacing is reduced. In practice this condition is rarely satisfied and flow distortions result in a deterioration of the accuracy of the method. A now commonly used remedy to this problem is the periodic remeshing of particles in a way that conserves as much as possible the physical invariants of the flow. Although particle remeshing was already used in some of the early simulations using particles, systematic and generic use of remeshing was first used in [16] to deliver high resolution analysis of a flow past a two-dimensional obstacle. Three-dimensional simulations using remeshed particle methods followed [7,21,6,24]. A parallel library of particle-mesh programs has been designed [22] and used for extensive simulations of vortex dynamics [3,4]. Remeshed particle methods also enable the use of Adaptive Mesh Refinement [1,2] and particle-grid domain decomposition [20,4].

Obviously, remeshing comes at the expense of introducing numerical errors in methods that, for advection problems with constant coefficient, are otherwise exact. The truncation errors related to remeshing can be measured on the basis of the conservation of the invariants of the advected quantities [9]. Particle remeshing is performed through interpolation formulas that can be derived to conserve as many of these invariants as desired. In practical implementation of the method it turns out that the time-scales which control the particle advection schemes and on which particle remeshing is done are both given in terms of the flow shear. Although in principle remeshing should be done on a time scale larger than the time step used to advect particles, with a ratio increasing as the particle resolution increases, in practice it is performed every "few" time steps, and often at every time step. In such implementations of particle methods, which we will call remeshed particle methods, it is clearly necessary to revisit the numerical analysis of the method.

In [11,26] remeshed particle methods were rewritten as finite-difference methods and analyzed as such. This analysis in particular led to a new understanding of remeshed particle methods for nonlinear conservation laws. In [13] a similar analysis is used to take into account remeshing in the solution of diffusion for viscous terms. The work of [11] was further pursued in [5] with the objective of understanding and overcoming oscillations generated by remeshing formulas near discontinuities. TVD remeshing formulas were derived for both linear and nonlinear advection equations. The present paper is a continuation of this work, with a particular emphasis on the design of non-oscillatory remeshing formulas for linear advection problems with large time-steps.

The outline of the paper is as follows. In Section 2 we recall our previous work on the finite-difference interpretation of remeshed particle methods. In Section 3 we derive corrections formulas to ensure consistency of second and fourth order remeshing formulas for large time steps. In Section 4 we derive limiting techniques to avoid oscillations generated by sharp variations of the solution. In Section 5 we describe alternate direction techniques to extend the 1D formulas to multi-dimensional problems and we sketch the overall algorithm of the method. Section 6 is devoted to numerical illustrations of the methods designed in Sections 3–5. Finally, conclusions and directions for future work are given in Section 7.

## 2. Previous work

Throughout this work, we assume that particles are initialized and remeshed on a uniform grid of grid-size $h$. In [5,26] an analysis of remeshed particle methods in terms of grid-based schemes was given, and is summarized below.

Consider the constant coefficient linear advection equation in 1 space dimension: $\partial u/\partial t + a \partial u/\partial x = 0$, with $a > 0$. The advection of particles over a time-step $\Delta t$ followed by a remeshing over the nearest grid points conserving the $m$ first moments of the solution leads to the following formula for each grid point $i$:

$$u_i^{n+1} = \sum_{-[\frac{m-1}{2}] \leqslant j \leqslant [\frac{m}{2}]} w_j u_{i-j-l}^n, w_j = c_j \prod_{-[\frac{m-1}{2}] \leqslant k \neq j \leqslant [\frac{m}{2}]} (\lambda' - k), \tag{1}$$

where the integer $l$ is defined by $l = [a\Delta t/h]$, $\lambda' = a\Delta t/h - l$ and

$$c_k = (-1)^{[\frac{m-1}{2}]+k} \left( \left[ \frac{m-1}{2} \right] + k \right)! \left( \left[ \frac{m}{2} \right] + k \right)!.$$

Not surprisingly, if $\lambda = a\Delta t/h \leqslant 1$ then $\lambda' = \lambda$ is the usual CFL number, and we obtain general finite-difference formulas for the linear advection equation [23]. For $m = 3$ we obtain $w_0 = 1 - \lambda^2$, $w_{\pm 1} = \mp\lambda(1 \mp \lambda)/2$-the Lax–Wendroff scheme. If $\lambda > 1$, the resulting solution consists of the exact solution of the equation over a time-step $\Delta t' = lh/a$ followed by a finite difference scheme of order $m - 1$ over the time interval $\Delta t - \Delta t'$.

More generally, consider the advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(au) = 0. \tag{2}$$

where $a = a(x,t)$. Assume that at a given time-step $t_n$ particles carrying the local values of $u$ move with a velocity $\tilde{a}(x)$ and then are remeshed by a formula which conserves the $m$ first moments of $u$. Then it is proved in [26] that, if $\tilde{a}$ and $u$ are smooth enough and $\lambda \leqslant 1/2$, the values $u_j^{n+1}$ obtained at the grid points satisfy

$$u_j^{n+1} = \sum_{i=0}^{m-1} (-1)^i \frac{\Delta t^i}{i!} \frac{\partial}{\partial x^i} (\tilde{a}^i u)(x_j, t_n) + O(h^m). \tag{3}$$

In particular, if particles are advected by a second order Runge–Kutta scheme, with $\tilde{a}(x) = a(x + \delta t \, a(x)/2)$ and $m = 3$, the scheme is second order.

Consider finally the case of non-linear systems of the form

$$U_t + \operatorname{div}(G(U) \otimes U) = 0. \tag{4}$$

In that case, particle methods consist of sampling $U$ on particles advected with velocity $G(U)$ and constant strength:

$$U(x) \simeq \sum_p \alpha_p \delta(x - x_p), \dot{x}_p = G(U_p). \tag{5}$$

The strengths of particles combine local volumes $v_p$ and local values $U_p$ of $U$: $\alpha_p = v_p U_p$. From the conservative nature of (5), particle strengths are constant, but their volumes and local values evolve according to

$$\dot{v}_p = \operatorname{div} G(U_p) v_p, \dot{U}_p = -\operatorname{div} G(U_p) U_p. \tag{6}$$

A second order in time and space remeshed particle method is obtained as follows. At time $t_n$ particles $x_p$ are located on a regular grid. Particle velocities at time $t_n + \Delta t/2$ are evaluated on the basis of (6) by the following formula:

$$G_p^{n+1/2} = G\left[U_p^n\left(1 - \frac{\Delta t}{2}\operatorname{div} G_p\right)\right]. \tag{7}$$

In the above formula, $\operatorname{div} G_p$ is evaluated by second order centered finite-differences of the quantity $G(U)$ at the particle (coinciding with a grid point) $x_p$. Then, according to (5), particles move with the velocities $G_p^{n+1/2}$:

$$x_p^{n+1} = x_p^n + \Delta t G_p^{n+1/2}$$

and are finally remeshed with a three points formula preserving the three first moments of the particle distribution. The resulting scheme is second order in time and space.

For systems of conservation laws, $U$ is a vector whose components are the density, the momentum and the energy. The system (4) is complemented by a pressure gradient in the right hand side. The complete system can then be solved in an Euler-Lagrange fashion by alternating the particle scheme just described and a finite-difference scheme for the gradient part of the system on the grid [26].

In [5] this analysis is further used to derive limiters in order to remove oscillations that are produced in remeshed particle methods near discontinuities. In this reference, limiters inspired by finite-difference formulas are derived under CFL conditions for constant coefficient linear equations and for the Burgers equation, and turned into non linear remeshing schemes.

Fig. 1 illustrates the effect of the remeshing schemes with limiters for the Burgers equation. The initial condition is a Heaviside function
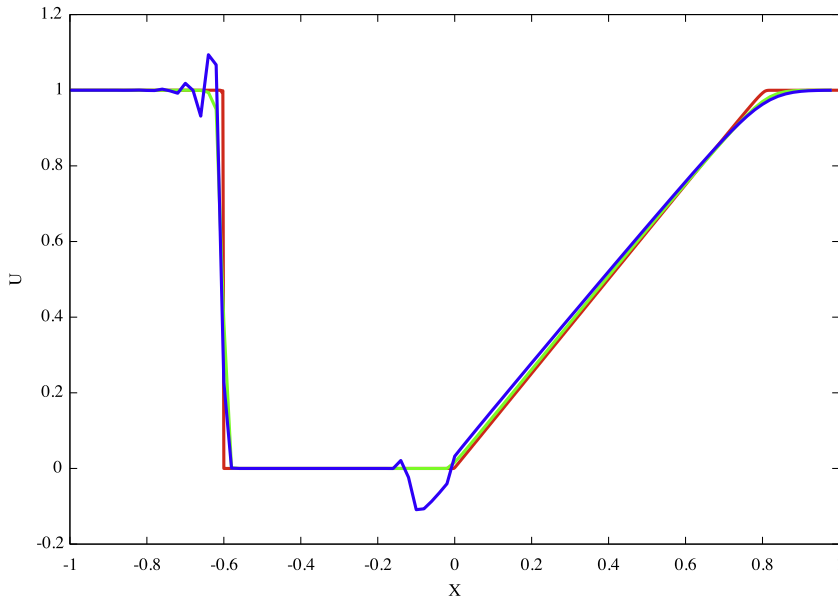


**Fig. 1.** From [5]. Simulation of the Burgers equation with $h = 0.02$ and CFL number 2/3. Particle method with a classical 3-points remeshing scheme (blue line) or a TVD 3 points remeshing scheme (green line), compared to the exact solution (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$u_0(x) = 0$   if $x \leqslant 0, u_0(x) = 1$ otherwise.

It develops a shock and a wave, both propagating to the right. The boundary conditions are periodic. The remeshing scheme is the three-points piecewise quadratic function preserving the 3 first moments described in Section 3. TVD remeshing formula were derived with a Van–Leer limiter. The solution is shown at time $t = 0.8$ for $h = 0.02$ and a CFL number 2/3. The TVD remeshing formula removes the oscillations that are visible when a classical remeshing scheme is used. One goal of the present work is to extend the work in [5] in the case of linear equations by relaxing the constraint on the time-step.

## 3. Consistent 1D remeshing formulas for large CFL numbers

In this section we focus on the linear advection Eq. (2) and we analyze, in the case of large CFL numbers, particle methods which consist for each time-step of a motion of particles with their local velocities followed by a remeshing step. The underlying grid is assumed to have a fixed grid size $h$.

To begin with, let us consider the case when particles are advected with an Euler scheme and remeshed with a linear interpolation. For a sake of simplicity in the notations, we assume that the advection velocity $a$ is time-independent. We denote by $x_j = jh$ the particles lying on the grid after remeshing, by $a_j$ their velocity and we set $v = \Delta t/h$ and $\lambda_j = a_j v$.

If we assume that for all particle $0 \leqslant \lambda_j \leqslant 1$ then it is readily that the corresponding finite-difference scheme is

$$u_i^{n+1} = u_i^n + v\left(a_i u_i^n - a_{i-1} u_{i-1}^n\right)$$

which is the first order upwind scheme. If now at a given particle $i$ we have a local CFL number satisfying $\lambda_i < 0 \leqslant \lambda_{i+1} \leqslant 1$ then we obtain $u_i^{n+1} = u_i^n + v a_i u_i^n$, which is not consistent with the advection equation. The same difficulty arises if $\lambda_i < 1 \leqslant \lambda_{i+1}$.

Higher order remeshing schemes can be obtained by ensuring the conservation of the 3 first moments of the particles. One then obtains the so-called $\Lambda_2$ interpolation kernel,

$$\Lambda_2(x) = \begin{cases} 1 - |x|^2 & \text{if } |x| \leqslant 1/2, \\ (1-|x|)(2-|x|) & \text{if } 1/2 < |x| \leqslant 3/2, \\ 0 & \text{elsewhere.} \end{cases} \tag{8}$$

It is readily seen that if remeshing is performed at every time-step, the resulting finite-difference schemes will be consistent (and second order) under the conditions $|\lambda_i| \leqslant 1/2$. A one-sided version of this remeshing formula, which will be defined in the next section, is consistent if $0 \leqslant \lambda_i \leqslant 1$.

We wish to stress the fact that in the discussion above it is not the value itself of the local CFL numbers that matters, but rather its variations from one particle to the next. For instance the centered-$\Lambda_2$ scheme will be consistent and second order if all particles have a local CFL number value between $N - 1/2$ and $N + 1/2$ for a given integer number $N$. Likewise, the one-sided $\Lambda_2$ scheme will be second order if all particles have a local CFL number value between $N$ and $N + 1$ for a given integer number $N$. The same properties are valid with all centered or one-sided remeshing formulas of even order.

A smoother version of the $\Lambda_2$ kernel, the so-called $M_4'$ scheme originally proposed in [19], is often used in practice. It uses one additional grid point and is given by:

$$M_4'(x) = \begin{cases} 1 - 5|x|^2/2 + |x|^2/2 & \text{if } |x| \leqslant 1, \\ (1-|x|)(2-|x|)^2/2 & \text{if } 1 < |x| \leqslant 2, \\ 0 & \text{elsewhere.} \end{cases} \tag{9}$$

This kernel does not suffer the consistency problem of $\Lambda_2$ but one can check that it is only first order at sonic points, independently of the local time-step value, and more generally when the CFL number crosses an integer value.

### 3.1. Correction formulas for second order remeshing schemes

In this section and the following we focus on the case of the $\Lambda_2$ remeshing schemes, and we derive formulas to enforce a consistent treatment of CFL variations between neighboring particles. Let us first define some notations. We denote by $y$ the distance between the particle after an advection step and its nearest grid point $i$ on the left, normalized by the grid-size. We thus have $y = \lambda - [\lambda]$. The left-$\Lambda_2$ scheme assigns to the grid points $i - 1$, $i$ and $i + 1$ weights respectively defined by (see Fig. 2)

$$\alpha(y) = y(y-1)/2, \quad \beta(y) = 1 - y^2, \quad \gamma(y) = y(1+y)/2. \tag{10}$$

Similarly the right-$\Lambda_2$ scheme uses the grid points $i$, $i + 1$ and $i + 2$ with the following weights

$$\alpha'(y) = (y-1)(y-2)/2, \quad \beta'(y) = y(2-y), \quad \gamma'(y) = y(y-1)/2. \tag{11}$$

We can observe that the following relationships hold:

$$\alpha'(y) = \alpha(y-1), \quad \beta'(y) = \beta(y-1), \quad \gamma'(y) = \gamma(y-1). \tag{12}$$

The center-$\Lambda_2$ remeshing scheme uses the weights $\alpha$, $\beta$, $\gamma$ if $y \leqslant 1/2$ and $\alpha'$, $\beta'$, $\gamma'$ if $y \geqslant 1/2$.
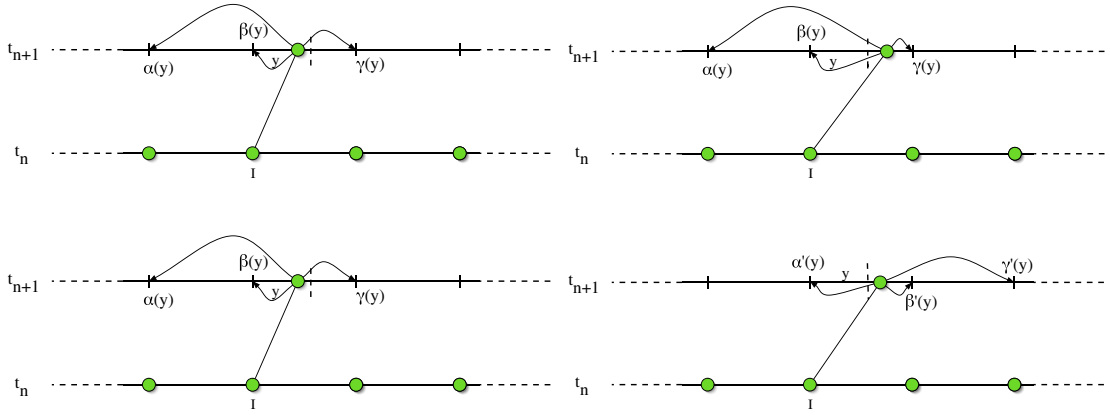
**Fig. 2.** Left and center $\Lambda_2$ remeshing schemes. Top row: left remeshing, bottom row: center remeshing. Left (respectively right) pictures correspond to the case $\lambda \leqslant 1/2$ (respectively $\lambda \geqslant 1/2$). The middle of the cells are materialized by a dashed line. Weight values are given in (10) and (11).

The solution we propose to overcome inconsistencies due to CFL variations is to combine centered and left remeshing schemes. The idea is to split particles in patches where either centered or left formulas can be used in a consistent way. More precisely, at the beginning of a given time step, we split the particles into blocks of size $Mh$. Each box thus contains $M+1$ particles. We first choose the time step on the basis of the strain rate of the flow, as it is usual for particle methods, and on the integer $M$, with the formula

$$\Delta t = \frac{1}{2(M+1)|a|_{1,\infty}} \tag{13}$$

where $|a|_{1,\infty}$ denotes the maximum of the absolute values of the derivatives of the advection field $a$. In the sequel $M$ can be any positive integer (and in practice it can be chosen equal to 1).

Let us consider two particles $x_i = ih$ and $x_j = jh$ lying in the same block. We can write

$$|\lambda_i - \lambda_j| \leqslant |a|_{1,\infty} \Delta t Mh/h \leqslant 1/2.$$

As a result we can state that, for a given block $B$, there exists an integer $N$ such that, either

$$\forall i \text{ such that } x_i \in B, |\lambda_i - N| \leqslant 1/2, \tag{14}$$

or

$$\forall i \text{ such that } x_i \in B, |\lambda_i - N - 1/2| \leqslant 1/2. \tag{15}$$

The determination of the case to be considered, and thus of the formula to be used, can be done by computing the minimal displacement of the particles within a block. We set

$$\bar{\lambda} = \min_{x_i \in B'} \lambda_i, \tag{16}$$

where $B'$ is the block $B$ extended to the next particle to the right (the reason for choosing $B'$ instead of $B$ will be apparent below). If $\mathcal{N}(\lambda)$ denotes the nearest integer to $\lambda$, it is straightforward to check that, for a certain value of $N$

- if $\bar{\lambda} \leqslant \mathcal{N}(\bar{\lambda})$, then all particles in $B$ satisfy (14),
- if $\bar{\lambda} > \mathcal{N}(\bar{\lambda})$, then all particles in $B$ satisfy (15).

Using the appropriate remeshing formulas, depending on which case one falls, guarantees the consistency of the push-and-remesh scheme *inside* each block.

Let us denote by $B_m$ the blocks as defined above and by $N_m$ the corresponding integers defined by (14) or (15). We will call this integer the index of the block. We will say that a given block $B_m$ is of type $C$ if (14) is satisfied, in which case a center-$\Lambda_2$ scheme is used for its particles, and of type $L$ if (15) is satisfied, in which case the left-$\Lambda_2$ formula is used. The question is now to determine the formula to be used for the last and first particles in each block.

Let us first show that if two blocks of the same type ($C$ or $L$) follow each other, then they can be merged into a single block where the center or left $\Lambda_2$ formula is consistent. Consider the case of two $C$-blocks $B_m$ and $B_{m+1}$ and denote by $N_m$ the index of $B_m$. There exists $x_j \in B_m$ such that $\lambda_j \leqslant N_m$. Let us denote by $x_l$ the first particle in $B_{m+1}$. We have $|x_j - x_l| \leqslant (M+1)h$ and, by (13), $|\lambda_i - \lambda_l| \leqslant (M+1)/2(M+1) = 1/2$. Therefore $\lambda_l \leqslant N_m + 1/2$. This implies that $N_{m+1} = N_m$. All particles in blocks $B_m \cup B_{m+1}$ have a value of $\lambda$ in $[N_m - 1/2, N_m + 1/2]$ and thus the center-$\Lambda_2$ remeshing is consistent in $B_m \cup B_{m+1}$. A similar argument

shows that if two type L blocks are next to each other, these two blocks can be merged into a single block where the left-$\Lambda_2$ scheme is consistent.

It remains now to consider the case of neighboring blocks of different types.

Assume two blocks $B_m$ and $B_{m+1}$ of different type are next to each other. Denote by $i$ the index of the last particle in $B_m$. If $B_m$ is of type $L$ and $B_{m+1}$ is of type $C$ we can write

$$|N_m + 1/2 - N_{m+1}| \leqslant |N_m + 1/2 - \lambda_i| + |N_{m+1} - \lambda_{i+1}| + |\lambda_i - \lambda_{i+1}| \leqslant 1/2 + 1/2 + 1/2(M+1).$$

Since we have assumed $M \geqslant 1$ we thus have $|N_m + 1/2 - N_{m+1}| < 5/4$ which implies that $N_m - N_{m+1} \in \{-1, 0\}$. Similarly, If $B_m$ is of type $C$ and $B_{m+1}$ is of type $L$ then $N_m - N_{m+1} \in \{0, +1\}$.

For a sake of simplicity we fix $m = 1$ and we denote by $I$ the index of the last particle in $B_1$. Consider first the case when $B_1$ is of type $L$ and $B_2$ of type $C$. We can assume without restricting the generality that $N_1 = 0$. We represent in Fig. 3 the possible configurations. The two first correspond to $N_2 = 0$, respectively with $\lambda_{I+1} \leqslant 0$ (we will denote this configuration as case (a)) and $\lambda_{I+1} \geqslant 0$ (case b) and the others to $N_2 = 1$, respectively with $\lambda_{I+1} \leqslant 1$ (case c) and $\lambda_{I+1} \geqslant 1$ (case d).

We need to determine how to modify the remeshing schemes for particles $I$ and $I + 1$ in order to recover a consistent scheme at time $t_{n+1}$ at all grid points.

For case (a), we observe that, due to (13) with $M \geqslant 1$, we have $0 \leqslant \lambda_{I-1} \leqslant 1/2$ and $0 \leqslant \lambda_I \leqslant 1/2$. Thus for particles $I - 1$ and $I$ the left and center remeshing schemes are equivalent and the remeshing formulas give a consistent 2nd order value to the grid point $I$ and $I + 1$. In other words no corrections are needed at this block interface. Similarly, for case (b), we observe that the center and left remeshing schemes are equivalent for particle $I + 1$ and that the left scheme is consistent including for the particle $I + 1$ (recall that this scheme is inconsistent only when $\lambda$ crosses an integer value). Therefore, no modification is needed in this case for particles $I$ and $I + 1$.

Let us next consider the case (c). In that case the center-$\Lambda_2$ scheme used for the particle $I + 1$ uses the right formulas (11). First we observe that if we choose the weights $\alpha(y_I)$, $\beta(y_I)$ for the particle $I$ respectively on grid points $I - 1$ and $I$, and the weight $\alpha(y_{I+1})$ for the particle $I + 1$ on the grid point $I$ we obtain a finite-difference scheme on grid points $I - 1$ and $I$ equivalent to that obtained if a left-$\Lambda_2$ remeshing scheme was used up to the particle $I + 1$, that is a second order scheme. Similarly, assigning a weight $\gamma'(y_I)$ from the particle $I$ on the grid point $I + 2$ and the weights $\beta'(y_{I+1})$ and $\gamma'(y_{I+1})$ from particle $I + 1$ onto grid points $I + 2$ and $I + 3$ gives on the grid points $I + 2$ second order finite difference formulas corresponding to the right-$\Lambda_2$ schemes for the particles $I$ and $I + 1$. It thus only remains to determine the contributions of particles $I$ and $I + 1$ to the grid-point $I + 1$. To do this, we remark that to ensure conservation of the mass of each particle, since $\alpha + \beta + \gamma = \alpha' + \beta' + \gamma' = 1$ the weight assigned to the grid point $I$ from the particle needs to be $\gamma(y_I) - \gamma'(y_I) = y_I$, and the weight assigned from the particle $I + 1$ on the same grid point has to be $\alpha'(y_{I+1}) - \alpha(y_{I+1}) = 1 - y_{I+1}$. Particles with indices below $I$ do not contribute to the grid point $I + 1$: due to (13), $y_{I-1} \leqslant 1$ and this particle remeshed with the left-$\Lambda_2$ scheme does not contribute to the grid point $I + 1$. Similarly, particles with indices beyond $I + 2$ do not contribute to the grid point $I + 1$: because the particle $I + 2$ is in a block of type $C$, it satisfies $y_{I+2} \geqslant 1/2$ and the center-$\Lambda_2$ formula does not assign weight to the grid
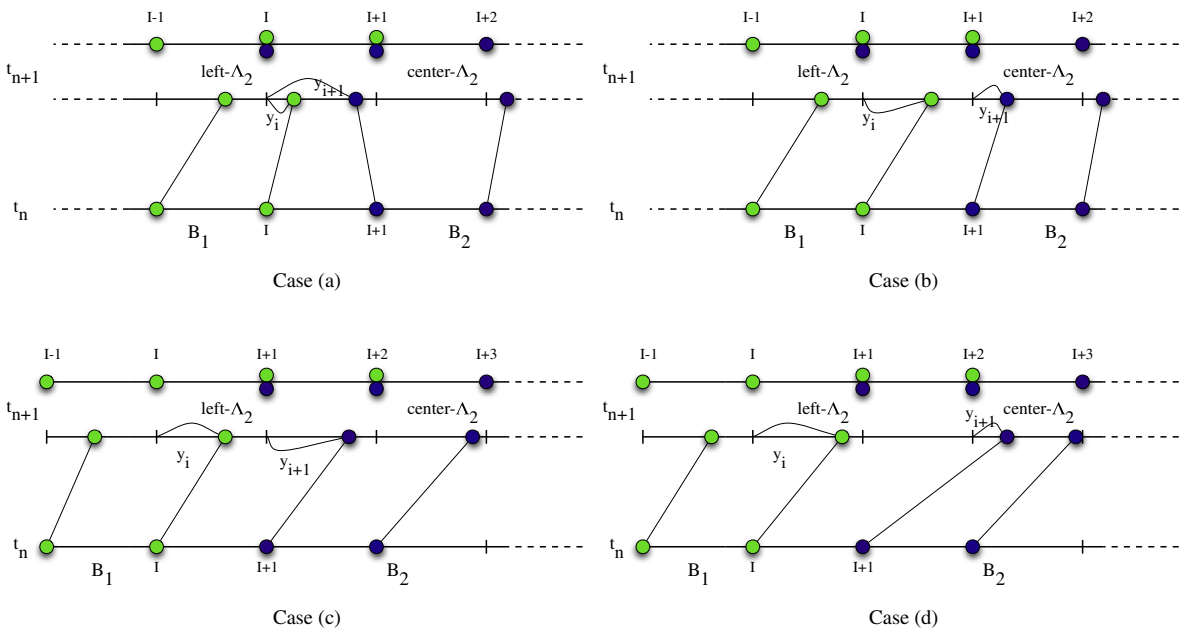


**Fig. 3.** The different possible configurations at the interface between a block of type $L$ ($B_1$) and a block of type $C$ ($B_2$). Particles of $B_1$ (resp. $B_2$) are in green (resp. blue). On the top rows, the colors indicate what type of particle contributes to the grid values after remeshing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

point $I + 1$. As a result the final contribution to the grid point $I + 1$ is $y_I u_I + (1 - y_{I+1})u_{I+1}$ which corresponds to the upwind scheme. We have therefore derived a correction formula which is locally first order.

We now turn to the last case (d). Particle $I$ is dealt with the same way as in case (c): it is remeshed with weights $\alpha(y_I)$, $\beta(y_I)$, $y_I$ and $\gamma'(y_I)$ respectively for grid points $I - 1$, $I$, $I + 1$ and $I + 2$. To determine the weights to remesh the particle $I + 1$, we observe that $\lambda_{I+1} = y_{I+1} + 1$. Therefore, to obtain a second order finite-difference scheme at the grid point $I$, this particle must assign there a weight $\alpha(y_{I+1} + 1)$. To obtain a second order scheme on grid points $I + 2$ and $I + 3$, it must assign weights $\beta(y_{I+1})$ and $\gamma(y_{I+1})$ on these grid points. Finally, to ensure mass conservation, the weight assigned by particle $I + 1$ to the grid point $I + 1$ must be $\alpha(y_{I+1}) - \alpha(y_{I+1} + 1) = -y_{I+1}$. Since $-y_{I+1} = 1 - \lambda_{I+1}$, it is now readily seen that at the grid-point $I + 1$ we recover the first order upwind scheme. Note that in cases (c) and (d), the remeshing formulas spread on a stencil of 4 grid points.

The general case (when $N \neq 0$) is simply recovered by shifting appropriately the grid indices. The corrections derived above are summarized in Table 1. We now consider the case of a block $B_2$ of type $L$ following a block $B_1$ of type $C$. We denote by $N_1$ and $N_2$ the indices of $B_1$ and $B_2$. For simplicity, we again assume that $N_1 = 0$, which implies that $N_2 = 0$ or $N_2 = -1$. Again, 4 cases have to be considered, as depicted in Fig. 4. The two first cases correspond to $N_2 = 0$.

In case (a'), which corresponds to $\lambda_I \leqslant 0$ we observe that, by (13), $\lambda_{I+2} \leqslant 1/2$ and the remeshing formula at particles $I$ and $I + 1$ are equivalent to the centered one, which ensures a consistent second order scheme at grid points $I$ and $I + 1$. If $\lambda_I \geqslant 0$ (case (b')), center and left remeshing formulas are equivalent for particles $I$ which again ensures a consistent second order scheme at grid points $I$ and $I + 1$. Therefore, cases (a') and (b') do not require any special treatment.

We now consider case (c') in Fig. 4, which corresponds to $\lambda_I \leqslant 0$ and $N_1 = -1$. To ensure a consistent second order treatment at grid points $I - 1$ and $I + 2$, it is readily seen that the particle $I$ must assign a weight $\alpha'(y_I)$ to the grid point $I - 1$ and the particle $I + 1$ must assign a weight $\gamma(y_{I+1})$ to the grid point $I + 1$. To conserve their mass, the particles $I$ and $I + 1$ must then

**Table 1**
Remeshing weights at the interface between a block of type $L$ and a block of type $C$. The cases are illustrated in Fig. 3. The weights $\alpha, \beta, \ldots$ are given in (10) and (11). $I'$ denotes the nearest grid-point on the left of particle $I$ after one advection step.

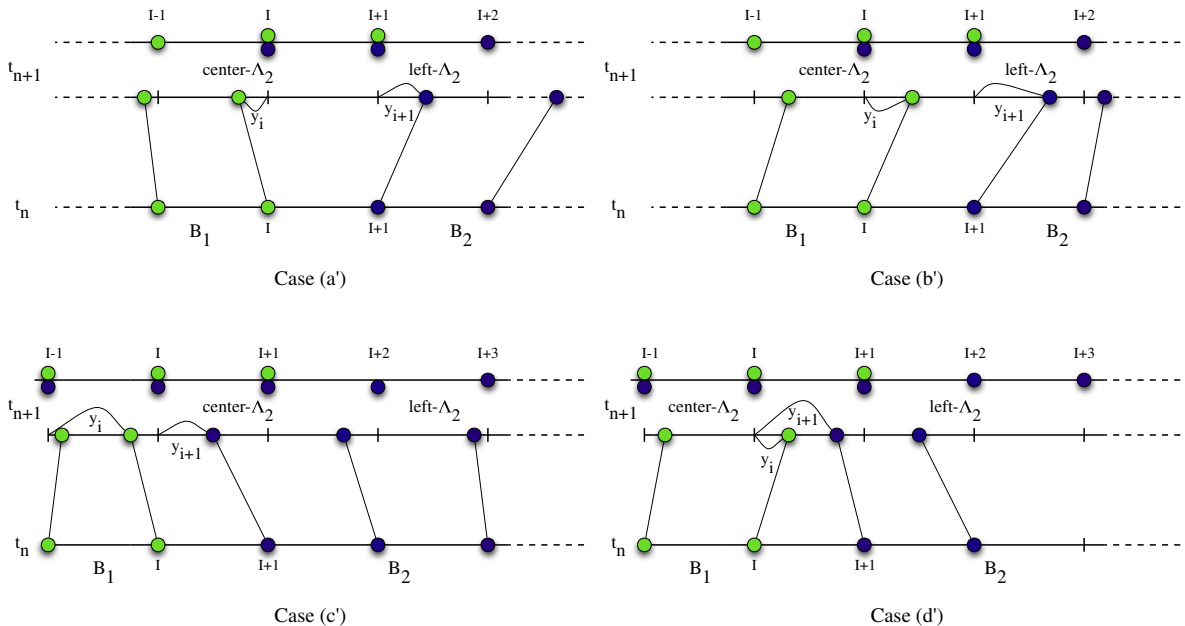| Grid points | Particles | | | | | |
|---|---|---|---|---|---|---|
| | Cases (a) / (b) | | Case (c) | | Case (d) | |
| | $I$ | $I + 1$ | $I$ | $I + 1$ | $I$ | $I + 1$ |
| $I' - 1$ | $\alpha(y_I)$ | 0 | $\alpha(y_I)$ | 0 | $\alpha(y_I)$ | 0 |
| $I'$ | $\beta(y_I)$ | $\alpha' / \alpha(y_{I+1})$ | $\beta(y_I)$ | $\alpha'(y_{I+1} + 1)$ | $\beta(y_I)$ | $\alpha(y_{I+1} + 1)$ |
| $I' + 1$ | $\gamma(y_I)$ | $\beta' / \beta(y_{I+1})$ | $y_I$ | $1 - y_{I+1}$ | $y_I$ | $-y_{I+1}$ |
| $I' + 2$ | 0 | $\gamma' / \gamma(y_{I+1})$ | $\gamma'(y_I)$ | $\beta'(y_{I+1})$ | $\gamma'(y_I)$ | $\beta(y_{I+1})$ |
| $I' + 3$ | 0 | 0 | 0 | $\gamma'(y_{I+1})$ | 0 | $\gamma(y_{I+1})$ |



**Fig. 4.** The different possible configurations at the interface between a block of type $C$ ($B_1$) and a block of type $L$ ($B_2$). Particles of $B_1$ (resp. $B_2$) are in green (resp. blue). On the top rows, the colors indicate what type of particle contributes to the corresponding grid values after remeshing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

assign respectively a weight $\beta'(y_I) + \gamma'(y_I)$ and a weight $\alpha(y_{I+1}) + \beta(y_{I+1})$ to the grid point $I$. The proof of the consistency of the resulting finite-difference scheme is postponed to the Appendix A.

We finally consider the case (d′) which corresponds to $\lambda_I \geqslant 0$ and $N_1 = -1$. To obtain a consistent second order upwind scheme on grid-points satisfying $J \leqslant I - 1$ or $J \geqslant I + 1$ we assign a weight $\gamma(y_{I+1})$ from particle $I + 1$ onto the grid point $I + 1$ and a weight $\alpha(y_I)$ from particle $I$ onto the grid-point $I - 1$. To ensure mass conservation, the particles $I$ and $I + 1$ must assign respectively the weights $\beta(y_I) + \gamma(y_I)$ and $\alpha(y_{I+1}) + \beta(y_{I+1})$ onto the grid-point $I$. The proof that this results in a first order scheme at the grid-point $I$ is given in the Appendix A.

The remeshing formulas just derived are summarized in Table 2.

### 3.2. Higher order schemes

Higher order remeshing formulas for particle methods are obtained by redistributing the weights of the particles on a larger number of grid-points to enforce the conservation of additional moments. We consider here the case of the fourth order $\Lambda_4$ remeshing scheme. This scheme uses 5 points. Using the same notations as above, for the left-$\Lambda_4$ scheme the weights associated to the grid points $i - 2$, $i - 1$, $i$, $i + 1$ and $i + 2$ are respectively

$$
\begin{aligned}
\alpha(y) &= \frac{1}{24}(y+1)y(y-1)(y-2), \quad \beta(y) = -\frac{1}{6}(y+2)y(y-1)(y-2) \\
\gamma(y) &= \frac{1}{4}(y+2)(y+1)(y-1)(y-2), \quad \delta(y) = -\frac{1}{6}(y+2)(y+1)y(y-2) \\
\eta(y) &= \frac{1}{24}(y+2)(y+1)y(y-1).
\end{aligned}
\tag{17}
$$

For the right-$\Lambda_4$ scheme, the weights associated to the grid points $i - 1$, $i$, $i + 1$, $i + 2$, and $i + 3$ are respectively given by the following relationships

$$
\alpha'(y) = \alpha(y-1), \quad \beta'(y) = \beta(y-1), \quad \gamma'(y) = \gamma(y-1), \quad \delta'(y) = \delta(y-1), \quad \eta'(y) = \eta(y-1).
\tag{18}
$$

**Table 2**
Same as in Table 1 for a block of type $L$ following a block of type $C$. The cases correspond to Fig. 4. $I'$ denotes the nearest grid-point to particle $I$ after one advection step.

| Grid points | Particles | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| | Cases (a′) and (b′) | | Case (c′) | | Case (d′) | |
| | $I$ | $I + 1$ | $I$ | $I + 1$ | $I$ | $I + 1$ |
| $I' - 1$ | $\alpha'/\alpha(y_I)$ | 0 | $\alpha'(y_I)$ | 0 | $\alpha(y_I)$ | 0 |
| $I'$ | $\beta'/\beta(y_I)$ | $\alpha(y_{I+1})$ | $(\beta' + \gamma')(y_I)$ | $(\alpha + \beta)(y_{I+1})$ | $(\beta + \gamma)(y_I)$ | $(\alpha + \beta)(y_{I+1})$ |
| $I' + 1$ | $\gamma'/\gamma(y_I)$ | $\beta(y_{I+1})$ | 0 | $\gamma(y_{I+1})$ | 0 | $\gamma(y_{I+1})$ |
| $I' + 2$ | 0 | $\gamma(y_{I+1})$ | 0 | 0 | 0 | 0 |

**Table 3**
Fourth order remeshing weights at the interface between a block of type $L$ and a block of type $C$. The weights $\alpha, \beta, \ldots$ are given in (17) and (18). Cases are defined in the text below (18) and in Fig. 3.

| Grid points | Particles | |
| --- | --- | --- |
| | $I - 1$ | $I$ |
| $I' - 3$ | $\alpha(y_{I-1})$ | 0 |
| $I' - 2$ | $\beta(y_{I-1})$ | $\alpha(y_I)$ |
| $I' - 1$ | $\gamma(y_{I-1})$ | $\beta(y_I)$ |
| $I'$ | $\delta(y_{I-1})$ | $\gamma(y_I)$ |
| $I' + 1$ | $\eta(y_{I-1}) - \eta(y_{I-1} - 1)$ | $(\delta + \eta)(y_I) - (\delta + \eta)(y_I - 1)$ |
| $I' + 2$ | $\eta(y_{I-1} - 1)$ | $\delta(y_I - 1)$ |
| $I' + 3$ | 0 | $\eta(y_I - 1)$ |

| Grid points | Particles | | | |
| --- | --- | --- | --- | --- |
| | $I + 1$ | | $I + 2$ | |
| | Case ($c_1$) / ($c_2$) | Case ($d_1$) / ($d_2$) | Case ($c_1$) / ($d_1$) | Case ($c_2$) / ($d_2$) |
| $I' - 1$ | $\alpha'(y_{I+1} + 1)$ | $\alpha(y_{I+1} + 1)$ | 0 | 0 |
| $I'$ | $\beta'(y_{I+1}+1)$ | $\beta(y_{I+1}+1)$ | $\alpha'(y_{I+2} + 1)$ | $\alpha(y_{I+2} + 1)$ |
| $I'+1$ | $(\alpha' + \beta')(y_{I+1})$ $-(\alpha' + \beta')(y_{I+1} + 1)$ | $(\alpha + \beta)(y_{I+1})$ $-(\alpha + \beta)(y_{I+1} + 1)$ | $\alpha'(y_{I+2})$ $-\alpha'(y_{I+2} + 1)$ | $\alpha(y_{I+2})$ $-\alpha(y_{I+2} + 1)$ |
| $I' + 2$ | $\gamma'(y_{I+1})$ | $\gamma(y_{I+1})$ | $\beta'(y_{I+2})$ | $\beta(y_{I+2})$ |
| $I' + 3$ | $\delta'(y_{I+1})$ | $\delta(y_{I+1})$ | $\gamma'(y_{I+2})$ | $\gamma(y_{I+2})$ |
| $I' + 4$ | $\eta'(y_{I+1})$ | $\eta(y_{I+1})$ | $\delta'(y_{I+2})$ | $\delta(y_{I+2})$ |
| $I' + 5$ | 0 | 0 | $\eta'(y_{I+2})$ | $\eta(y_{I+2})$ |

**Table 4**
Fourth order remeshing weights at the interface between a block of type $C$ and a block of type $L$. The cases are illustrated in Fig. 3. The weights $\alpha, \beta, \ldots$ are given in (17) and (18). Cases are defined in the text below (18) and in Fig. 4.

| Grid points | Particles | | | |
|---|---|---|---|---|
| | $I-1$ | | $I$ | |
| | Case $(c_1')/(d_1')$ | Case $(c_2')/(d_2')$ | Case $(c_1')/(c_2')$ | Case $(d_1')/(d_2')$ |
| $I'-3$ | $\alpha'(y_{I-1})$ | $\alpha(y_{I-1})$ | $0$ | $0$ |
| $I'-2$ | $\beta'(y_{I-1})$ | $\beta(y_{I-1})$ | $\alpha'(y_I)$ | $\alpha(y_I)$ |
| $I'-1$ | $\gamma'(y_{I-1})$ | $\gamma(y_{I-1})$ | $\beta'(y_I)$ | $\beta(y_I)$ |
| $I'$ | $\delta'(y_{I-1})$ | $\delta(y_{I-1})$ | $(\gamma'+\delta'+\eta')(y_I)$ | $(\gamma+\delta+\eta)(y_I)$ |
| | $+\eta'(y_{I-1})$ | $+\eta(y_{I-1})$ | $-\eta'(y_{I+1})$ | $-\eta(y_{I+1})$ |
| $I'+1$ | $0$ | $0$ | $\eta'(y_{I+1})$ | $\eta(y_{I+1})$ |
| | | Particles | | |
| Grid points | | $I+1$ | | $I+2$ |
| $I'-1$ | | $\alpha(y_{I+1}-1)$ | | $0$ |
| $I'$ | | $(\alpha+\beta+\gamma)(y_{I+1})-\alpha(y_{I+1}-1)$ | | $(\alpha+\beta)(y_{I+2})$ |
| $I'+1$ | | $\delta(y_{I+1})$ | | $\gamma(y_{I+2})$ |
| $I'+2$ | | $\eta(y_{I+1})$ | | $\delta(y_{I+2})$ |
| $I'+3$ | | $0$ | | $\eta'(y_{I+2})$ |

As for the $\Lambda_2$ remeshing schemes, the centered-$\Lambda_4$ (respectively left-$\Lambda_4$) scheme will be consistent if all particles have a local CFL number value between $N-1/2$ and $N+1/2$ (respectively $N$ and $N+1$) for a given integer number $N$. Blocks where centered and left formulas can be used are therefore determined in the same way as previously, and corrections in the remeshing weights are needed at the interface between two blocks of different types and different indices.

Assume two blocks $B_m$ and $B_{m+1}$ of different type are next to each other. As for the $\Lambda_2$-remeshing scheme, $N_m - N_{m+1} \in \{0,1\}$ if $B_m$ is of type C, $B_{m+1}$ of type L and $N_m - N_{m+1} \in \{-1,0\}$ otherwise. The same cases – (a) to (d) if a block of type $C$ follows a block of type $L$, (a') to (d') otherwise – have to be considered. Corrections have to be derived only in the cases (c), (d), (c') and (d'). As for the second order remeshing formulas, no correction is needed in the cases (a), (b), (a'), (b'). For the other cases, because the stencil is larger for the remeshing function $\Lambda_4$, one needs to distinguish several sub-cases depending on the CFL number of the particle $I+2$. These cases are denoted by $c_1, c_2, d_1, d_2, c_1', c_2', d_1', d_2'$ and are defined as follows. Cases $(c_1)$ and $(d_1)$ (resp $(c_2)$ and $(d_2)$) correspond to particles $I$ and $I+1$ in the configuration of the case (c) or (d) in Fig. 3, and $\lambda_{I+2} \leqslant N$ (resp $\lambda_{I+2} > N$) where $N$ is the index of the block containing the particle $I+2$. Similarly, cases $(c_1')$ and $(d_1')$ (resp $(c_2')$ and $(d_2')$) correspond to particles $I$ and $I+1$ in the configuration of the case (c') or (d') in Fig. 4, and $\lambda_{I-1} \leqslant N$ (respectively $\lambda_{I-1} > N$) where $N$ is the index of the block containing the particle $I-1$. For blocks of size $Mh$ with $M \geqslant 3$, and time-steps satisfying the condition (13), the corrected remeshing schemes which need to be used for particles $I-1$ to $I+2$ in each of these cases are given in the Table 3 for cases (c) and (d) and Table 4 for cases (c') and (d'). The proof that these remeshing schemes lead to grid formulas of third order under the condition on the time-step is given in [18]. In this reference these correction formulas are also extended to the case of smaller blocks ($M \geqslant 1$).

## 4. Limiting techniques for TVD remeshing formulas

High order remeshing formulas, like any non-dissipative grid-based methods, are likely to produce oscillations wherever the advected quantity undergoes strong variations. In this section we extend the approach of [5] to derive limiting techniques for remeshing formulas in the case when the CFL number is larger than 1.

We restrict here our discussion to the case of the 3-points $\Lambda_2$ remeshing formula considered in Section 3.1. Limiting techniques for the 4th order formulas derived in Section 3.2 is studied elsewhere [18]. In [5] we derived limiters that were based on the linear first order remeshing scheme. As we have already seen, this scheme is not appropriate when the advection velocity changes sign. To be able to combine limiting techniques to the corrections just derived, we will instead use two different first order formulas based on quadratic splines and using three grid points. The first formula is obtained by taking successive convolutions of the top hat function. This is the formula used in traditional Particle In Cell methods for the so-called TSC (for *Triangular Shape Clouds*) assignment schemes. Following the terminology of [9], we will call $M_3$ this remeshing technique, with left and right variants as for the $\Lambda_2$ formulas. With the same notations as in (10), the weights assigned from a particle to the grids points for the left-$M_3$ formula are, from left to right

$$\alpha(y) = \frac{1}{2}\left(\frac{1}{2}+y\right)^2, \quad \beta(y) = 3/4 - y^2, \quad \gamma(y) = \frac{1}{2}\left(\frac{1}{2}-y\right)^2. \tag{19}$$

As for the case of the $\Lambda_2$ formulas, the weights corresponding to the right formula are obtained by the formulas

$$\alpha'(y) = \alpha(y-1), \quad \beta'(y) = \beta(y-1), \quad \gamma'(y) = \gamma(y-1). \tag{20}$$

The center $M_3$ formula consists of taking the $\alpha$, $\beta$, $\gamma$ if $y \leqslant 1/2$ and $\alpha'$, $\beta'$, $\gamma'$ otherwise.

For a constant velocity field $a$, it is readily seen that if $\lambda = a\Delta t/h$ is such that $|\lambda| \leqslant 1/2$ the center-$M_3$ remeshing formula (19) is equivalent the following finite-difference scheme

$$u_j^{n+1} = u_j^n \left(\frac{3}{4} - \lambda^2\right) + \frac{1}{2}\left(\frac{1}{2} + \lambda\right)^2 u_{j-1}^n + \left(\frac{1}{2} - \lambda\right)^2 u_{j+1}^n. \tag{21}$$

This scheme can be rewritten as a central finite-difference scheme corrected by an artificial viscosity term:

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}\left(u_{j+1}^n - u_{j-1}^n\right) + \left(\frac{\lambda^2}{2} + \frac{1}{8}\right)\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right). \tag{22}$$

From this expression, one can see that this scheme is TVD provided that $\lambda^2 + 1/4 \leqslant 1$, a condition which is satisfied when $0 \leqslant \lambda \leqslant 1/2$.

This calculation actually shows that the center-$M_3$ remeshing scheme is TVD for any value of the CFL number, and that the left-$M_3$ scheme is TVD for $\lambda - [\lambda] \leqslant \sqrt{3/4}$. In the case the center-$M_3$ scheme, if $n$ is the nearest integer to $a\Delta t/h$, the method can be seen as an exact solution over a time $nh/a$, a step which is clearly TVD, followed by the above finite difference scheme over a time $\Delta t' = \Delta t - nh/a$. Since $|a\Delta t'/h| \leqslant 1/2$ this method is TVD. In the case of the left-$M_3$ scheme, this remeshing scheme is equivalent to the exact solution to the advection equation over a time $\Delta t' = [\lambda h/a]$ followed by the finite-difference scheme (21), over a time-step equal to $\Delta t - \Delta t'$. Since $a(\Delta t - \Delta t') \leqslant \sqrt{3/4}h$ this scheme is TVD.

In the general case of a varying velocity field, if $|\nu a_j| \leqslant 1/2$ for all $j$ the finite-difference Eq. (22) extends in the following way:

$$u_j^{n+1} = u_j^n - \frac{\nu}{2}\left(a_{j+1}u_{j+1}^n - a_{j-1}u_{j-1}^n\right) + \frac{\nu^2}{2}\left(a_{j+1}u_{j+1}^n - 2a_j u_j^n + a_{j-1}u_{j-1}^n\right) + \frac{1}{8}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right). \tag{23}$$

This proves that, in all blocks of type $C$, as defined in the previous section, the center-$M_3$ scheme is consistent with the advection equation, with a truncation error of the order of $((a^2 u)'' + u''/4)h^2/\Delta t$. Similarly, it can be checked that the left-$M_3$ scheme is consistent in all blocks of type $L$.

We now show how these first-order remeshing formulas can be used to limit the left and center-$\Lambda_2$ schemes devised in the previous section in order to give a TVD remeshing scheme. Let us go back to the case of an advection equation with constant velocity field (note that for a varying velocity field the advection equation in conservation form is not TVD so there is no clear cut criterion to limit oscillations in this case). We recall that the choice of the left or center remeshing schemes is based on the value of $\bar{\lambda}$ as defined in (16) compared to its nearest integer. If we consider the linearized advection equation we thus have two cases to consider for $\lambda$, up to the addition of an integer: either $0 \leqslant \lambda = a\Delta t/h \leqslant 1/2$, in which case left-formulas are used, or $-1/2 \leqslant \lambda = a\Delta t/h \leqslant 0$, in which case the center formulas are used. Let us first consider the case $0 \leqslant \lambda = a\Delta t/h \leqslant 1/2$. We proceed like in [5] and write the finite-difference forms of the remeshing formulas in incremental form. The left-$\Lambda_2$ scheme is then equivalent to the Lax–Wendroff scheme and can be written as

$$u_j^{n+1} = u_j^n + C_{j+1/2}^{(2)}\Delta u_{j+1/2}^n - D_{j-1/2}^{(2)}\Delta u_{j-1/2}^n, \tag{24}$$

where we used the classical notation $\Delta u_{j+1/2} = u_{j+1} - u_j$ and the coefficient $C$ and $D$ take the following values:

$$C_{j+1/2}^{(2)} = -\frac{\lambda}{2}(1 - \lambda), \quad D_{j-1/2}^{(2)} = \frac{\lambda}{2}(1 + \lambda).$$

Similarly, the left-$M_3$ remeshing scheme yields the following incremental form

$$u_j^{n+1} = u_j^n + C_{j+1/2}^{(1)}\Delta u_{j+1/2}^n - D_{j-1/2}^{(1)}\Delta u_{j-1/2}^n, \tag{25}$$

where:

$$C_{j+1/2}^{(1)} = \frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2, \quad D_{j-1/2}^{(1)} = \frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2.$$

We thus have

$$C_{j+1/2}^{(2)} = C_{j+1/2}^{(1)} - \frac{1}{8}, \quad D_{j-1/2}^{(2)} = D_{j-1/2}^{(1)} - \frac{1}{8}.$$

Following the approach of [14] we now look for a limiter of the $\Lambda_2$ formula by the $M_3$ formula under the form:

$$\begin{aligned} u_j^{n+1} &= u_j^n + \left(C_{j+1/2}^{(1)} - \frac{\phi_{j+1/2}}{8}\right)\Delta u_{j+1/2}^n - \left(D_{j-1/2}^{(1)} - \frac{\phi_{j-1/2}}{8}\right)\Delta u_{j-1/2}^n \\ &= u_j^n + \frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2 \Delta u_{j+1/2}^n - \Delta u_{j-1/2}^n\left[\frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 + \frac{\phi_{j-1/2}}{8} - \frac{\phi_{j+1/2}}{8r_{j+1/2}}\right]. \end{aligned} \tag{26}$$

In the above equation we have set $r_{j+1/2} = \Delta u_{j-1/2}^n/\Delta u_{j+1/2}^n$ and $\phi_{j+1/2} = \phi(r_{j+1/2})$, where $\phi$ is the limiter function to be determined. Eq. (26) defines a TVD scheme provided the following conditions are satisfied [14]:

$$\phi \geqslant 0, \quad \phi(r) = 0 \quad \text{if } r \leqslant 0, \quad \text{and} \quad |\phi(s) - \phi(r)/r| \leqslant \Phi$$

where the constant $\Phi$ is such that the coefficients of $\Delta u_{j\pm 1/2}^n$ in (26) are positive and have a sum less than or equal to 1. This yields the following conditions:

$$\frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 - \frac{\Phi}{8} \geqslant 0$$

and

$$\frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2 + \frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 + \frac{\Phi}{8} = \lambda^2 + \frac{1}{4} + \frac{\Phi}{8} \leqslant 1.$$

If $0 \leqslant \lambda = a\Delta t/h \leqslant 1/2$, these conditions are satisfied as soon as $\Phi \leqslant 1$. One may then use the classical limiter formulas to derive eligible function $\phi$. For a given limiter choice, the finite difference formula becomes:

$$\begin{aligned}
u_j^{n+1} &= u_j^n + \left[\frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2 - \frac{\phi_{j+1/2}}{8r_{j+1/2}}\right]\Delta u_{j+1/2}^n - \left[\frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 - \frac{\phi_{j-1/2}}{8}\right]\Delta u_{j-1/2}^n \\
&= u_j^n\left[\frac{3}{4} - \lambda^2 + \frac{\phi_{j+1/2} + \phi_{j-1/2}}{8}\right] + u_{j-1}^n\left[\frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 - \frac{\phi_{j-1/2}}{8}\right] + u_{j+1}^n\left[\frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2 - \frac{\phi_{j+1/2}}{8}\right].
\end{aligned} \tag{27}$$

This scheme can in turn be interpreted as a particle scheme using a left remeshing formula. Using the same notations as in (10), (19), it is readily seen that for a given particle $j$ the weights $\alpha, \beta, \gamma$ assigned to the grid points $i - 1, i, i + 1$ are given by the formula:

$$\beta(y) = \frac{3}{4} - y^2 + \frac{\phi_{j+1/2} + \phi_{j-1/2}}{8}, \quad \gamma(y) = \frac{1}{2}\left(y + \frac{1}{2}\right)^2 - \frac{\phi_{j+1/2}}{8}, \quad \alpha(y) = \frac{1}{2}\left(y - \frac{1}{2}\right)^2 - \frac{\phi_{j-1/2}}{8}. \tag{28}$$

As expected, one recovers for $\phi = 1$ the left-$\Lambda_2$ scheme, and, for $\phi = 0$, the left-$M_3$ scheme.

Let us now consider the case $-1/2 \leqslant \lambda = a\Delta t/h \leqslant 0$. Proceeding as before, we can write the center-$\Lambda_2$ scheme, limited by the center-$M_3$ scheme in the following incremental form:

$$u_j^{n+1} = u_j^n - \frac{1}{2}\left(\lambda + \frac{1}{2}\right)^2 \Delta u_{j-1/2}^n + \Delta u_{j+1/2}^n\left[\frac{1}{2}\left(\lambda - \frac{1}{2}\right)^2 - \frac{\psi_{j+1/2}}{8} + \frac{\psi_{j-1/2}}{8\tilde{r}_{j-1/2}}\right]. \tag{29}$$

where $\tilde{r}_{j-1/2} = \Delta u_{j+1/2}^n/\Delta u_{j-1/2}^n$ and $\psi_{j+1/2} = \psi(\tilde{r}_{j+1/2})$. The conditions on $\psi$ to enforce the TVD property of the finite-difference scheme are similar to those found above for $\phi$, namely:

$$\psi \geqslant 0, \quad \psi(r) = 0 \quad \text{if } r \leqslant 0, \quad \text{and} \quad |\psi(s) - \psi(r)/r| \leqslant 1.$$

The incremental formula (29) can then be turned into the following remeshing weights:

$$\alpha'(y) = \frac{1}{2}\left(y - \frac{3}{2}\right)^2 - \frac{\psi_{j-1/2}}{8}, \quad \beta'(y) = \frac{3}{4} - (y-1)^2 + \frac{\psi_{j+1/2} + \psi_{j-1/2}}{8}, \quad \gamma'(y) = \frac{1}{2}\left(y - \frac{1}{2}\right)^2 - \frac{\psi_{j+1/2}}{8}. \tag{30}$$

Note that the upwinding direction only appears in the choice of the slopes in the formulas giving $\phi$ (for positive velocities) and $\psi$ (for negative velocities).

## 5. Extension to higher dimension

Particle remeshing in dimension higher than one is in general performed by formulas that are tensor products of 1D formulas. However, the correction and limiters that we have just derived do not extend to tensor product formulas. It is therefore necessary to adopt the viewpoint of directional splitting to deal with multi-dimensional problems. Using 1D remeshing formulas saves computational time compared to the traditional tensor-product approach (for instance, for the 5 points $\Lambda_4$ remeshing, in 3D the cost will be of the order of $O(15N)$ instead of $O(125N)$ if $N$ is the number of particles) but makes the derivation of high order time-stepping schemes less straightforward.

In the context of particle methods, splitting can be envisioned in two different ways. The first option is to follow the classical Strang splitting method, which alternates advection equation in all directions. In this case, one has to successively push then remesh particles along each axis. If second order is desired, one has to alternate directions over half time-steps. In case a second order Runge–Kutta scheme is used to advect particles, in 2 dimensions one time-step push-and-remesh second order time stepping can be sketched as follows:

**Advection along 1st direction for $\Delta t/2$**
- Evaluation of velocities by mid-point rule: $\tilde{\mathbf{x}}_p = \mathbf{x}_p + \mathbf{a}_1(\mathbf{x}_p)\Delta t/4$, then $\tilde{\mathbf{a}}_1 = \mathbf{a}_1(\tilde{\mathbf{x}}_p)$
- Advection of particles along 1st axis: $\mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_1 \Delta t/2$
- 1D particle remeshing along 1st axis: $(\mathbf{x_i}, u_i) \leftarrow (\mathbf{x}_p, u_p)$

**Advection along 2nd direction for $\Delta t$**
- Evaluation of velocities by mid-point rule: $\tilde{\mathbf{x}}_p = \mathbf{x}_p + \mathbf{a}_2(x_p)\Delta t/2$, then $\tilde{\mathbf{a}}_2 = \mathbf{a}_2(\tilde{x}_p)$
- Advection of particles along 2nd axis: $\mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_2 \Delta t$
- 1D particle remeshing along 2nd axis

**Advection along 1st direction for $\Delta t/2$** $\cdots$

In the above description, the velocity $a_1, a_2$ are the components of the velocity field $\mathbf{a}$, and, by extension, we also denote by $\mathbf{a}_1$ the velocity field $(a_1, 0)$.

The second option is specific to particle methods. Assume again we choose a second order Runge–Kutta method for the particle advection. This means that at the beginning of a time-step one has available a velocity field which allows to advect particles with accuracy $O(\Delta t^2)$. Of course one obtains the same output by advecting particles successively along each axis with the corresponding velocity component. If remeshing takes place after advection in each direction, one can obtain the same accuracy in time provided some care is taken on the velocity values used after remeshing in the first direction. Instead of the local velocity on the grid points, one needs to track back the proper velocity. More precisely, the algorithm at time $t_n = n\Delta t$ is as follows:

**Computation of velocity first component by a RK2 scheme**

$$\text{Advection of particles in } 2D \text{ for } \Delta t/2 : \tilde{\mathbf{x}}_p = \mathbf{x}_p + \mathbf{a}(\mathbf{x}_p, t_n)\Delta t/2 \tag{31}$$

$$\text{Velocity evaluation by mid-point rule} : \tilde{\mathbf{a}}_1(\mathbf{x}_p) = \mathbf{a}_1(\tilde{\mathbf{x}}_p, t_n + \Delta t/2) \tag{32}$$

**Push and remesh along 1st direction for $\Delta t$**

$$\text{Advection of particles along 1st axis} : \mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_1(\mathbf{x}_p)\Delta t \tag{33}$$

$$\text{1D Particle remeshing along first axis} \tag{34}$$

**Computation of velocity second component by a RK2 scheme**

$$\text{Advection of particles in 2D for } \Delta t/2 : \tilde{\mathbf{x}}_p = \mathbf{x}_p + (-a_1(\mathbf{x}_p, t_n)\Delta t/2, a_2(\mathbf{x}_p, t_n)\Delta t/2) \tag{35}$$

$$\text{Velocity evaluation by mid-point rule} : \tilde{\mathbf{a}}_2(\mathbf{x}_p) = \mathbf{a}_2(\tilde{\mathbf{x}}_p, t_n + \Delta t/2) \tag{36}$$

**Push and remesh along 2nd direction for $\Delta t$**

$$\text{Advection of particles along 2nd axis} : \mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_2(\mathbf{x}_p)\Delta t \tag{37}$$

$$\text{1D Particle remeshing along second axis.} \tag{38}$$

The reason for choosing the velocity field $\tilde{\mathbf{a}}_2 = \mathbf{a}_2(\tilde{\mathbf{x}}_p)$, where $\tilde{\mathbf{x}}_p = \mathbf{x}_p + (-a_1\Delta t/2, a_2\Delta t/2)$ in (35), can be best understood in the case when a given particle moves exactly to the next grid point after advection along the first axis. In this case, remeshing does not change its weights and this particle should next move along the second axis with the RK2 velocity computed at the beginning of the time-step. It is not difficult to realize that $\tilde{\mathbf{a}}_2$ is a second order approximation of this velocity. The extension to 3D of this scheme is straightforward: one just has to add a push-and-remesh step in the third direction, with velocities evaluated on particles $\tilde{\mathbf{x}}_p = \mathbf{x}_p + (-a_1\Delta t/2, -a_2\Delta t/2, a_3\Delta t/2)$.

This scheme only requires two remeshing formulas – one in each direction, instead of 3 for the first option, so it is mode economical if velocity values are known analytically.

The extension to third order goes along the following lines:

**Push and remesh along 1st direction for $\Delta t$**
- Computation of velocity:

$$
\begin{aligned}
\mathbf{a}^{(1)}(\mathbf{x}_p) &= \mathbf{a}\left[\mathbf{x}_p + \left(\frac{2}{3}\Delta t a_1(\mathbf{x}_p), \frac{2}{3}\Delta t a_2(\mathbf{x}_p)\right)\right] \\
\tilde{a}_1(\mathbf{x}_p) &= -a_1(\mathbf{x}_p) + \frac{3}{4}a_1^{(1)}(\mathbf{x}_p) + a_1\left[\mathbf{x}_p + \frac{\Delta t}{4}(-a_1(\mathbf{x}_p) + a_1^{(1)}(\mathbf{x}_p), 0)\right] + \frac{1}{4}a_1\left[\mathbf{x}_p - \Delta t(0, a_2(\mathbf{x}_p) + a_2^{(1)}(\mathbf{x}_p))\right]
\end{aligned} \tag{39}
$$

- Advection of particles along 1st axis: $\mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_1 \Delta t$
- 1D Particle remeshing

**Push and remesh along 2nd axis for $\Delta t$**
- Computation of velocity:

$$a_1^{(1)}(\mathbf{x}_p) = a_1 \left[ \mathbf{x}_p + \left( -\frac{1}{6}\Delta t \; a_1(\mathbf{x}_p), \frac{1}{3}\Delta t \; a_2(\mathbf{x}_p) \right) \right]$$

$$a_2^{(1)}(\mathbf{x}_p) = a_2 \left[ \mathbf{x}_p + \left( -\frac{2}{3}\Delta t \; a_1(\mathbf{x}_p), \frac{1}{3}\Delta t \; a_2(\mathbf{x}_p) \right) \right] \tag{40}$$

$$\tilde{a}_2(\mathbf{x}_p) = -a_2(\mathbf{x}_p) + \frac{3}{4}a_2^{(1)}(\mathbf{x}_p) + a_2\left[ \mathbf{x}_p + \Delta t(a_1(\mathbf{x}_p) - a_1^{(1)}(\mathbf{x}_p), 0) \right] + \frac{1}{4}a_2\left[ \mathbf{x}_p - \Delta t(0, a_2(\mathbf{x}_p) + 2a_2^{(1)}(\mathbf{x}_p)) \right]$$

- Advection of particles along 2nd axis: $\mathbf{x}_p \leftarrow \mathbf{x}_p + \tilde{\mathbf{a}}_2 \Delta t$
- 1D Particle remeshing.

The numerical analysis of the above splitting methods are given in [18].

To conclude this section, let us give some remarks on the overall algorithm. For a 2D or 3D simulation, whatever time-splitting strategy is chosen, the algorithm proceeds by alternating advection-remeshing steps in successive directions. For an advection-remeshing step in a given direction, one proceeds line by line. On each line, particles with a strength beyond a certain threshold are created and sorted in blocks. Then one determines, on the basis of the time step and the local velocity of particles in each blocks, the types (*L* or *C*) and the indices of the blocks. Particles at the interface of blocks of different type and indices are tagged, together, for the corrected $\Lambda_4$ scheme, with the next particles on the left and on the right. Finally untagged particles are remeshed by regular remeshing kernels, and tagged particles with the corrected formulas. For the schemes with TVD limiters, slopes are determined when particles are created, and then incorporated in the $\Lambda_2$ remeshing formulas.

Concerning the corrected $\Lambda_2$ remeshing formulas, one may observe that one goes from case (c) to case (d) by changing, for the particle $I + 1$, the weights $\alpha$, $\beta$, $\gamma$ into $\alpha'$, $\beta'$, $\gamma'$, or equivalently using the distance of the particle under consideration to the nearest grid point instead of the nearest grid point on the left. The same observation ca be made for the particle $I$ in cases (c') and (d'). In the case of the corrected $\Lambda_4$ this observation is also valid to go from cases $(c_1)$ or $(d_1)$ to $(c_2)$ or $(d_2)$ for particle $I + 2$, and from cases $(c'_1)$ or $(d'_1)$ to $(c'_2)$ or $(d'_2)$ for particle $I - 1$. In practice this means that in the actual algorithm for the corrected $\Lambda_2$ or $\Lambda_4$ formulas, only two type of formulas have to be implemented. Overall, in a 3D code, the correction formulas concern only a portion of the particles and have therefore a marginal cost. The algorithm for sorting and tagging particles takes about the same amount of CPU time as the 3 line-by-line one-dimensional remeshing algorithms.

## 6. Numerical illustrations

In all the experiments below we have taken blocks of size $M = 1$ for the second order $\Lambda_2$ remeshing schemes, and $M = 3$ for the 4th order $\Lambda_4$ remeshing scheme.

### 6.1. Convergence study in 2D flows

To illustrate the correction formulas given in Section 3 and the time-stepping schemes in Section 4, we first consider the case of 2D flows. Our first example is a smooth axisymmetric blob $u$ advected by an incompressible rotating field $a$:

$$u(x, y) = \max(0, (1. - (x^2 + y^2))^6), \quad a(x, y) = \cos(3\pi\sqrt{x^2 + y^2})(y, -x) \tag{41}$$

The solution to this problem is stationary. In Fig. 5 we show the relative $L^2$ error for increasing resolution for several remeshing schemes and CFL numbers at $t = 0.8$. The remeshing schemes are the original and corrected $\Lambda_2$ schemes, the nominally 6th order $\Lambda_6$ scheme, and the corrected $\Lambda_4$ scheme. We also consider the case of the $M'_4$ remeshing formula (9). For the time advancement of particles we chose in all cases a classical RK3 scheme. Second order methods were used in combination with the second order time splitting method (31)–(37), and higher order methods were used with the third order time splitting method (39) and (40). As expected, the plain $\Lambda_2$ formula gives second order accuracy for CFL numbers smaller that 0.5. For larger CFL numbers (3 in this experiment) it is barely convergent. The higher order $\Lambda_6$ remeshing formula also fails to converge with the expected accuracy. On the contrary the corrected second order formula gives almost second order accuracy, and the corrected 4th order formula, in combination with third order splitting and particle time-stepping yields the desired 3rd order accuracy. One may also observe that in this test case the $M'_4$ formula also exhibits second order convergence. We next consider the case of the advection of a smooth ring in an expanding velocity field. The computational box is the square $[-2,+2]^2$. The initial condition is the $C^2$ axisymmetric function with value 0 for $r \leqslant 0.4$ or $r \geqslant 1$, value 1 for $0.6 \leqslant r \leqslant 0.8$ and polynomial of degree 5 for $0.4 \leqslant r \leqslant 0.6$ and $0.8 \leqslant r \leqslant 1$. The velocity field is

$$\mathbf{a}(x, y) = (x/r, y/r) \tag{42}$$

Fig. 6 shows a 3D view of the solution at time $t = 0.8$ for $h = 0.01$ and CFL number equal to 6, for the $M'_4$ formula, and the corrected $\Lambda_2$ and $\Lambda_4$ formulas. One can see that the $M'_4$ schemes generate oscillations along radii that correspond to variations of the local CFL number. Oscillations created by the original $\Lambda_2$ scheme (not shown) are even larger. The corrected $\Lambda_2$ and $\Lambda_4$ formulas avoid theses oscillations. One can also notice the effects of the first order corrections in the corrected $\Lambda_2$ formula.
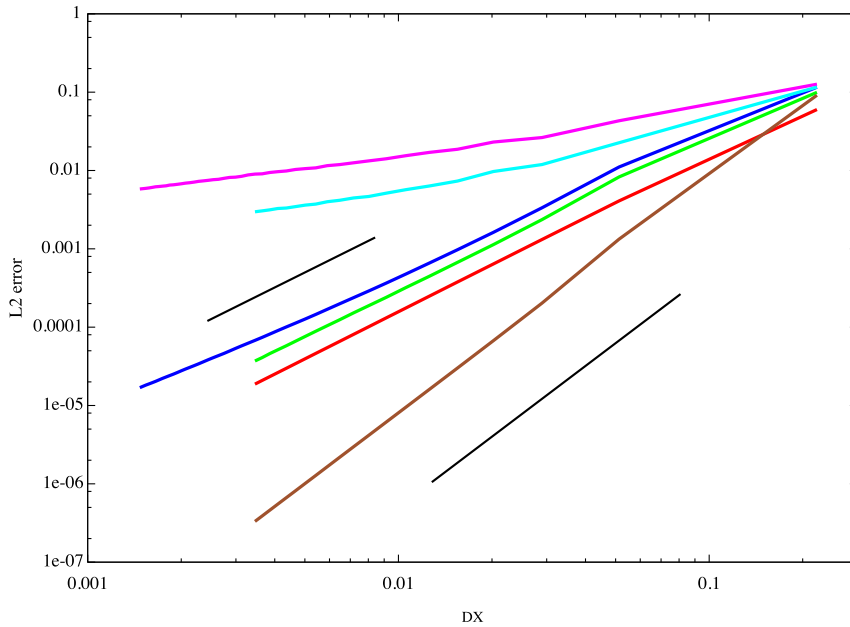
**Fig. 5.** Refinement study for the case of a smooth blob rotating in an incompressible flow. Remeshing by (from top to bottom): uncorrected $\Lambda_2$ (pink), uncorrected $\Lambda_6$ (magenta), $M_4'$ (blue), corrected $\Lambda_2$ (green), uncorrected $\Lambda_2$ (red) and corrected $\Lambda_4$ (brown). Except for the red curve, all cases used a CFL number of 3. For the red curve, the uncorrected $\Lambda_2$ remeshing was used with CFL number 0.4. All cases used a RK2 particle pusher and a second order time-splitting, except for the $\Lambda_4$ and $\Lambda_6$ cases which used a third Runge–Kutta and time-splitting. Black lines correspond to second and third order convergence rates. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The bottom-right picture in Fig. 6 is a convergence study in the $L^\infty$ norm for the same test, which illustrates the gain in accuracy of the corrected formulas.

## 6.2. Illustrations in incompressible 3D flows

Interface capturing is an important particular case where transport equations need to be solved with high accuracy. We first consider here the classical test case of a sphere advected in a shear flow proposed in [17]. The computations are done in a unit cubic box. The sphere is initially centered at the point of coordinates (0.5, 0.5, 0.5) and its radius is 0.15. In Fig. 7 we compare the particle method using the corrected $\Lambda_4$ scheme and a second order time splitting, with several similar recent experiments in the literature where lagrangian markers are used to complement grid-based methods: the particle level set method [12] and the volume of fluid method [25]. To compare with [12,25] we have first considered the following velocity field

$$\begin{cases} u(x,y,z,t) = 2f(t)\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\ v(x,y,z,t) = -f(t)\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\ w(x,y,z,t) = -f(t)\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{cases} \tag{43}$$

with $f(t) = \cos(\pi t/T)$, $T = 3$. Fig. 7 shows the interface at time $t = T/2$ when the flow distortion is maximum. In the top left picture, we show the results of [12]. In this method, 5th order finite-difference weno scheme is used to advance and reinitialize the level set equation. The method uses 100 grid points in each direction, supplemented by 4 particles per grid-size in each direction to improve the accuracy at the interface. The time-step value is not mentioned in the reference, but the CFL number, related to the finite-difference grid spacing is presumably of the order of 1. The top right picture shows the results of [25]. In this reference a Volume of Fluid method is used on a $64^3$ grid, and complemented with nine markers per cell, and the CFL number was 0.1. On the bottom left picture we use the particle method with the corrected $\Lambda_4$ remeshing formula on a grid using 100 points in each direction. The CFL number resulting from (13) is 8. The VOF and the particle method discretize a color function, with value 1 inside the sphere and 0 outside. The particle level set method of [12] captures a distance function to the surface of the sphere. The bottom-right picture corresponds to the same simulation on a grid using 160 points with a CFL number equal to 12. The three first results agree rather well. However the particle method uses significantly less particles than in [12] and the time step is larger. Our number of particles is comparable to that in [25] but our time step is significantly larger. The higher resolution case is shown to illustrate the resolution necessary to prevent the lack of connectivity of the surface. In all our experiments particles carrying a value smaller than $10^{-4}$ were discarded. In the higher resolution case, the number of particles started at 63,000 and reached 345,000. The total mass was conserved up to 0.04%. A complete run takes about 30 seconds for 40 iterations on a MacInstosh laptop running a 3.06 GhZ intel processor. Note that, if in interface capturing applications one is only interested in contours for levels above 0.5, one may choose a higher cut-off
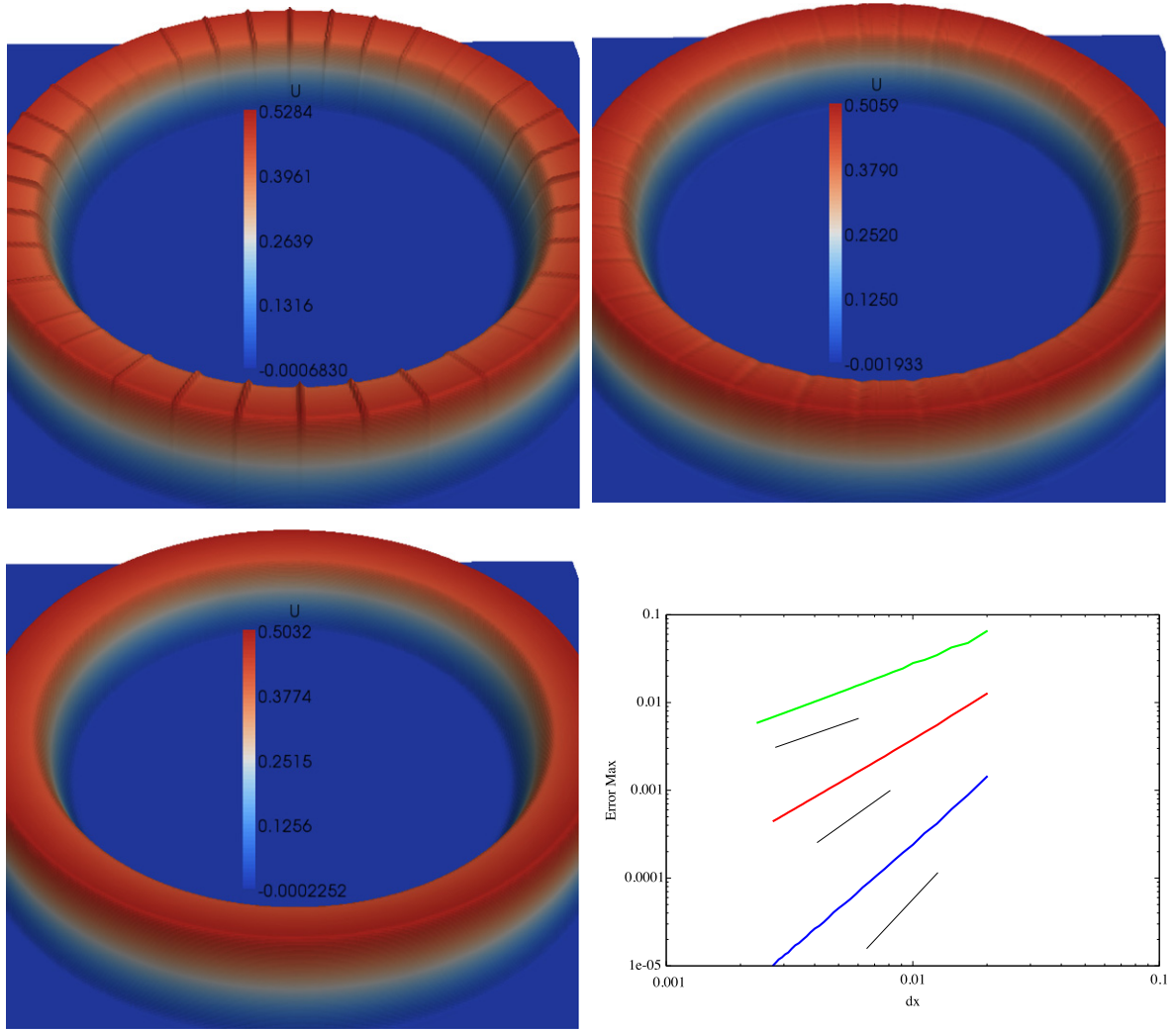
**Fig. 6.** Advection of a smooth axisymmetric function with a radial velocity fieldd (42). Top-left $M_4'$ remeshing scheme with second order time-splitting. Top-right corrected $\Lambda_2$ with second order time-splitting. Bottom left: corrected $\Lambda_4$ with second-order time-splitting. $N = 400$ and $CFL = 6$. Bottom-right convergence study in the maximum error norm for the $M_4'$ (green curve), $\Lambda_2$ (red curve), and $\Lambda_4$ (blue curve) for a CFL number equal to 3. Black lines correspond to first, second order and third order convergence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

level without changing the results. Taking a value of 0.025 like in [15] divides the final number of particle by a factor about 2, with of course a further reduction in the CPU cost.

For this particular experiment, very similar results are obtained with the same CFL condition by the classical $M_4'$ remeshing scheme that was used in [15]. A more quantitative comparison with other methods can be obtained by looking at the volume contained inside isosurfaces of different values. Ideally all these volumes should give the same constant value, equal to the volume of the sphere. In Fig. 8 we show the values obtained by the $M_4'$ and the corrected $\Lambda_4$ remeshing schemes, for the isosurfaces corresponding to the levels 0.5 and 0.75. The grid uses 160 points in each direction and the CFL number is equal to 12 in both cases. For this experiment we do not modulate the value of the velocity (in other words we take $f(t) = 1$ in (43)) and we run for longer times. This experiment shows the increased accuracy obtained by the $\Lambda_4$ scheme. In the previous experiment the solution at the time of maximum strain $T/2$ is equivalent to the one obtain here at time $t = T/\pi < 1$. At this time for the iso-surface 0.5 both remeshing methods gives similar volumes for the isosurface 0.5, with an error less than 1%. In passing we notice that this case is most challenging without modulating the velocity field on times larger than 1.

Our next experiment is of a more qualitative nature but shows the potential and limitation of the present methods. This experiment concerns the transport of a scalar in a turbulent plane jet. The jet is computed by the vortex in cell method using the $M_4'$ remeshing scheme [7], in a periodic box of size 1 using $128^3$ grid points. The scalar is advected by the velocity field obtained from the vortex method, filtered on a $32^3$ grid. The goal of such experiment, following the idea of [8], is to explore the possibility to use particle methods at a sub-grid resolution in a flow LES to resolve small scales of the scalar.
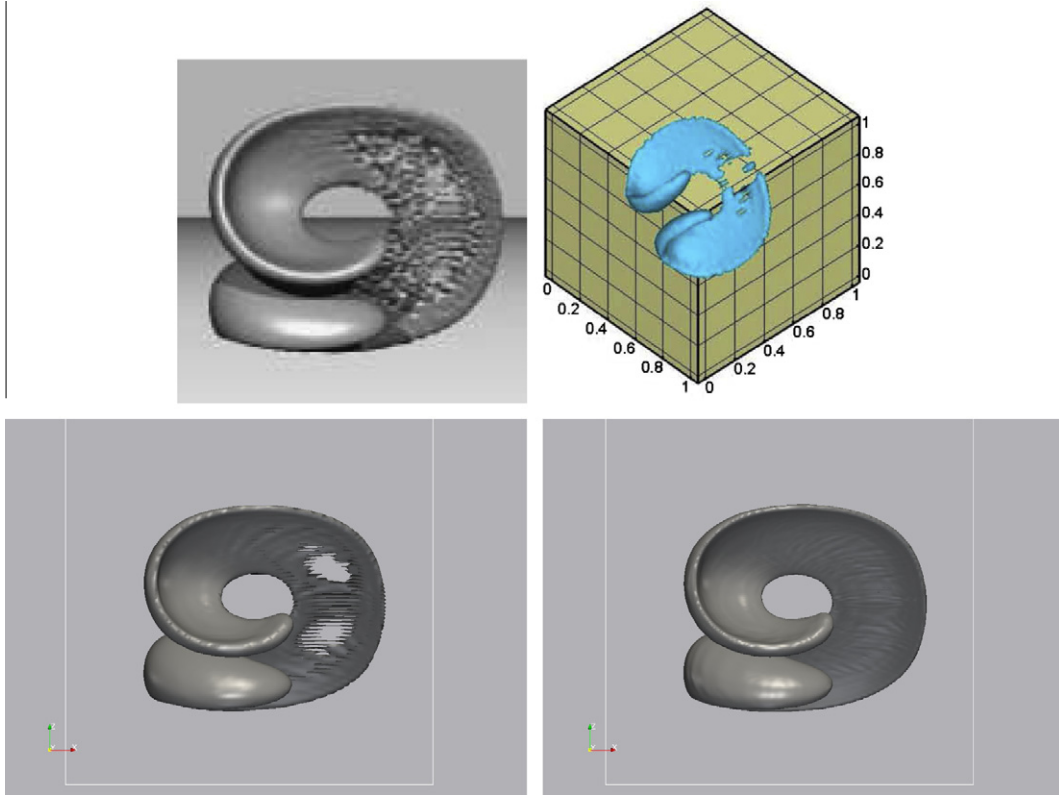
**Fig. 7.** Sphere advected in a shear flow (43). Top-left: [12] with N = 100 grid points complemented with 4 particles per grid-size in each direction around the interface. Top-right: [25], N = 64 and 9 particles per grid-cell, CFL = 0.1. Bottom: present 4th order method with N = 100 and CFL = 8 (left picture), N = 160 and CFL = 12 (right picture).



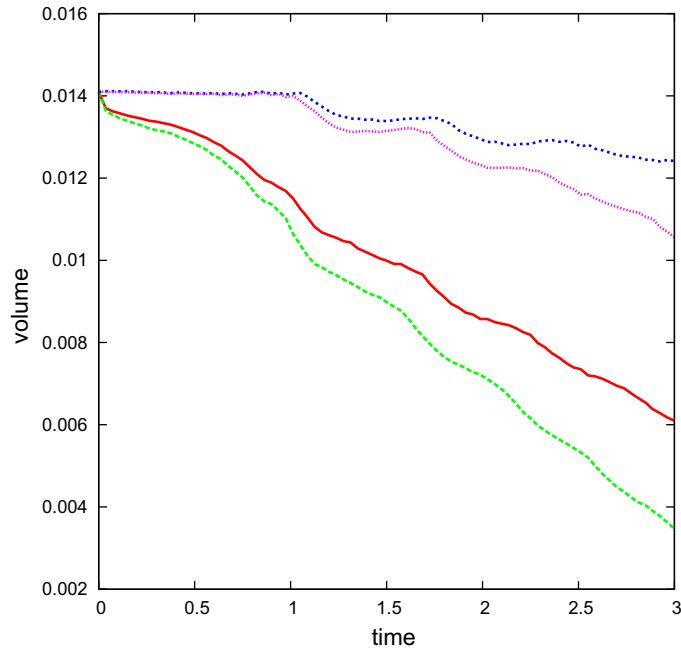**Fig. 8.** Sphere advected in a shear flow (43). Volumes contained inside the 0.5 (top curves) and 0.75 (bottom curves) contours for the $M'_4$ (magenta and green curves) and the fourth order remeshing scheme (blue and red curves). In both cases N = 160 and the CFL number is equal to 12. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

More precisely, the initial condition is as follows. In the unit box $[0,1]^3$, the initial vorticity is given by $\omega(x,y,z) = \mathbf{V} \times \mathbf{u}(x,y,z) + rand(x,y,z)$ where

$$u_x(x,y,z) = (1 + \tanh(y'))(1 + \lambda \sin 8\pi x)/2, \quad u_y = u_z = 0, \tag{44}$$

*rand* is a random perturbation of magnitude 0.05, $\lambda = 0.3$, $y' = (|y - 0.5| - w/2)\eta$, $\eta = 0.02$ and $w = 0.1$. The advected scalar is initialized with the same profile as $u_x$.

The scalar is discretized by particles remeshed on a $256^3$ uniform grid. The Reynolds number, based on the jet width $w$, is equal to $10^3$ and the Schmidt number, ratio of the scalar diffusivity and flow viscosity is equal to 10. In this experiment we use the corrected $\Lambda_4$ remeshing scheme, a classical second-order alternate direction Strang time-splitting for the scalar advection and a second order Runge Kutta time-stepping for the particle motion. The time-step is adapted at every time-step to the flow conditions to satisfy the upper bound of the condition (13) with $M = 3$.

The left picture of Fig. 9 shows the time evolution of the enstrophy, the time step for the scalar advection and the percentage of particles where corrections are implemented in the remeshing formula. The scalar time-step is expressed as a CFL number related to the maximal flow velocity and the scalar grid-size $h = 1/256$. One can observe that in a first stage, when the jet is laminar, the scalar time-step corresponds to CFL number of the order of 10 to 20. When the flow evolves to a turbulent stage, something which is correlated to a rapid enstrophy increase, the CFL number decreases to 3-4. Fig. 10 shows the scalar contours in a top and side views through the middle of the jet. The 3D scalar spectrum at time $t = 4.5$ is shown on the right picture of Fig. 9. It exhibits a $k^{-1}$ decay beyond the cut-off scale corresponding to the flow velocity. This test illustrates the capabilities of the method to allow very fine resolution of advected scalar without the need to reduce the time-step at a prohibitive value. However, when the advection field becomes turbulent the advantage of particle methods over more classical grid-based schemes is less clear. In that case, to take full advantage of the Lagrangian nature of particle methods, it is advisable to adapt locally the time-step, a possibility that was used in [1] but was not yet tested in the context of non-oscillatory formulas, independently of the local grid refinement.
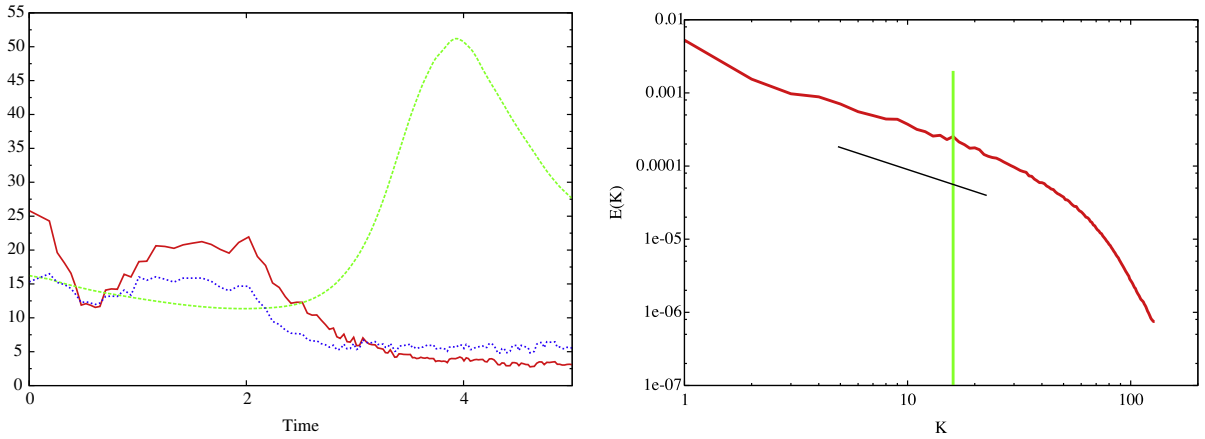


**Fig. 9.** Scalar transport in a plane jet using a filtered $32^3$ velocity field and a scalar discretization on $256^3$ points. Left picture: time evolution of the enstrophy (green curve), CFL number (red curve) and percentage of particles using correction formulas in $\Lambda_4$ remeshing (blue curve). Right picture: three-dimensional scalar spectrum at time $t = 4.5$ (red curve). Black line: $k^{-1}$ slope, green line: velocity cut-off wave number. The jet initial condition is given in (44). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
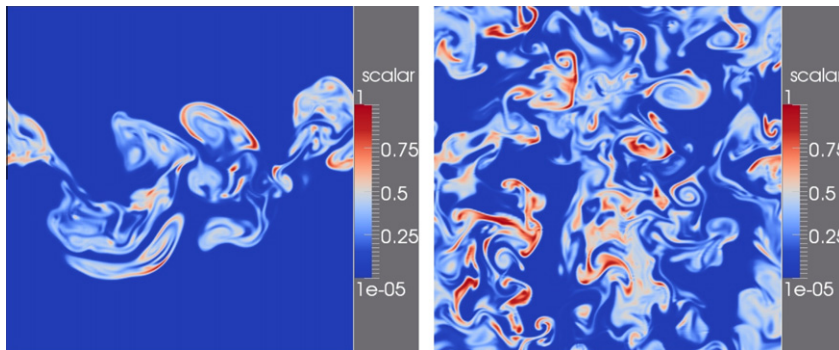


**Fig. 10.** Side and top view of scalar contours at $t = 4.5$ for the jet (44).

### 6.3. Results for remeshing formulas with limiters

We now give numerical illustrations of the limiters that we have derived in Section 4 for the $\Lambda_2$ remeshing formula. In one dimension, it of course does not make sense to consider the usual constant velocity case: in this case, as predicted by the condition (13), it is not necessary to remesh particles, and particle methods give the exact solution. We instead consider the case of the advection of a discontinuous function in a deformation field. The computational domain is the interval $[-1, +1]$. The scalar is at time $t = 0$ a double top hat function and the velocity field $a$ is a periodic sinusoidal wave:

$$u_0(x) = 1 \quad \text{if } x \in [-0.3, -0.1] \bigcup [0.1, 0.3], \quad 0 \text{ elsewhere,} \tag{45}$$

$$a(x) = 1 + \sin(\pi x)/2. \tag{46}$$

This velocity field results in a translation to the right together with alternating increase and decrease of the solution. The solution is time periodic with a period given by $T = 4/\sqrt{3}$. In the top picture of Fig. 11 we compare the solution obtained by the limited $\Lambda_2$ scheme with the solution obtained by a 5th order Weno scheme using a Lax–Friedrichs flux and a RK3 time-stepping. In both cases we take $h = 0.005$. The particle method uses a Van Leer limiter and the CFL number is equal
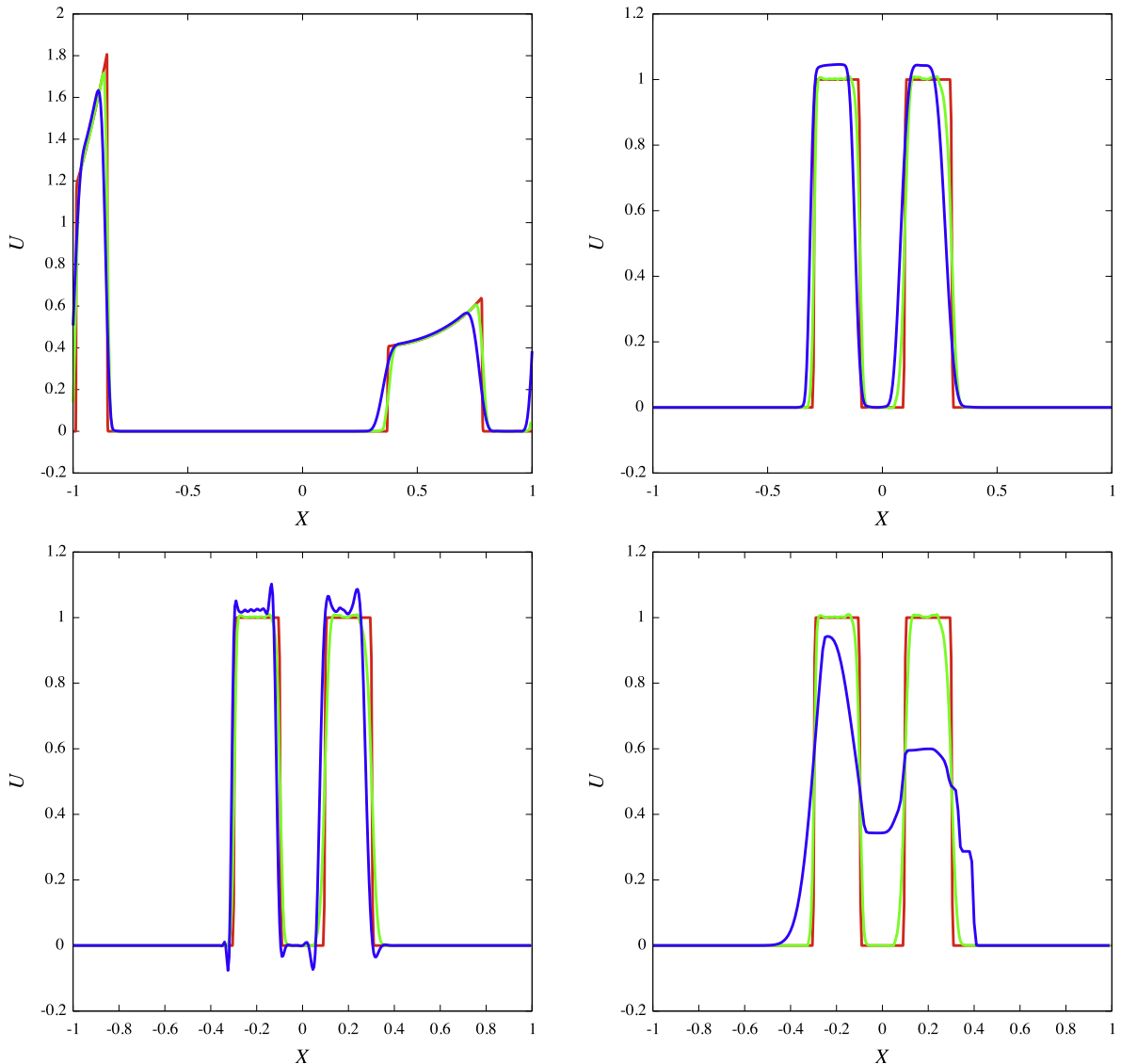


**Fig. 11.** Discontinuous profile (45) in a deformation field (46). Top pictures: Comparisons of particle methods with second order remeshing and TVD limiter at CFL12 (green curves) to the exact solution (red curve) and to a 5th order Weno scheme at CFL 2 (blue curve). Solution shown at $t = 3$ (top-left) and $t = 6.926 = 3T$ (top-right). Bottom picture: comparison at time $t = 3T$ with results of the $M_4'$ remeshing scheme at CFL 12 (blue curve, bottom-left) and to second order TVD remeshing at CFL 0.5 (blue curve, bottom-right). In all cases $h = 0.005$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
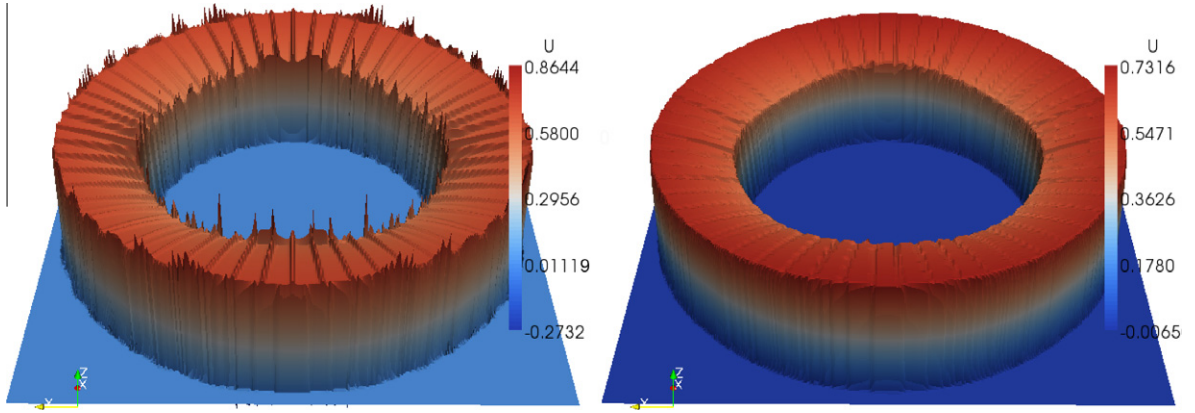
**Fig. 12.** Same as top pictures in Fig. 6 with top hat profile inside the ring.

to 12. For the Weno scheme the CFL number is equal to 2. The bottom-left picture of this figure shows a comparison of these 2 methods after 3 turns. One can see that the particle method resolves better the sharp variations of the solution, although it is only second order and uses a time-step 6 times larger. Finally the bottom-left picture is a comparison of our method with the result of a particle method using the same number of points, the same CFL number and the $M'_4$ remeshing formula. One can see that the $M'_4$ formula gives acceptable results for the location of the discontinuities, but produces overshoots and oscillations of the order of 10% of the maximum value of the solution. Finally the bottom-right picture of this figure shows the result of our method when a CFL number of 0.5 is used. In this case, no correction is implemented for the remeshing formula, but the limitation introduces a large amount of diffusion. This is not surprising, since, for this value of the CFL number, the particle method with $\Lambda_2$ remeshing reduces to a plain second order finite-difference scheme with first-order correction. This case is interesting in showing that using high CFL numbers is important, not only because it reduces the computational cost but also because, by taking advantage that particle methods give the exact solution when the advection field is constant, it reduces numerical diffusion.

Our last illustration is the ring in a radial velocity field already considered, but with an initial condition which is equal to 1 inside the ring, 0 outside. For this case, we compare our second order corrected and limited $\Lambda_2$ scheme with the $M'_4$ remeshing scheme. In both case we use a grid with $h = 0.01$ and a CFL number equal to 10. In Fig. 12 one can see that the $M'_4$ scheme produces oscillations near the discontinuities, in additions to those already seen in the smooth zone of the solution. The corrected and limited $\Lambda_2$ scheme avoids these oscillations.

## 7. Conclusion

In this paper we have analyzed remeshing formulas, when they are used at every time-step in particle methods, in a grid-based framework. We have derived, for second order and fourth order remeshing interpolation kernels, correction formulas to ensure the consistency of the resulting grid schemes when the CFL number varies from one particle to the next. In addition we have shown that the grid-based analysis enables the implementation of TVD limiters in remeshing formulas. This analysis yields rigorous bounds for the time step, or remesh frequency, to be used in remeshed particle methods. The accuracy of the resulting methods have been demonstrated in several examples. The results in particular show that the efficiency of the method, both in terms of computational cost and accuracy, is very much related to the possibility to use large time-steps only limited by the local flow strain.

Ongoing works include the implementation of these new remeshing schemes in AMR based particle methods and their applications in high resolution simulations of scalar transport in turbulent flows.

## Acknowledgment

## Appendix A

We prove here that the remeshing formulas defined in Table 2 are consistent in the cases (c′) and (d′). We first consider the case (c′). We set $\lambda_i = a(x_i)\Delta t/h$, $\alpha_I = \alpha(y_I)$ and similar notations for the other remeshing weights and indices. Let us first assume $\lambda_{I-1} \leqslant 0$.

At the grid point $I$ we obtain after remeshing the following value at time $t_{n+1}$:

$$u'_I = \gamma'_{I-1}u_{I-1} + \left(\beta'_I + \gamma'_I\right)u_I + \left(\alpha_{I+1} + \beta_{I+1}\right)u_{I+1} + \alpha_{I+2}u_{I+2}. \tag{47}$$

We next evaluate the weights by the formulas (10), (11) and observe that $y_{I-1} = 1 + \lambda_{I-1}, y_I = 1 + \lambda_I, y_{I+1} = 1 + \lambda_{I+1}, y_{I+2} = 1 + \lambda_{I+2}$. After some elementary calculations we obtain

$$u_I' = \frac{\lambda_{I-1}}{2}(\lambda_{I-1} + 1)u_{I-1} + \left(1 - \frac{\lambda_I}{2}(\lambda_I - 1)\right)u_I - \frac{\lambda_{I+1}}{2}(\lambda_{I+1} + 3)u_{I+1} + \frac{\lambda_{I+2}}{2}(\lambda_{I+2} + 1)u_{I+2}. \tag{48}$$

A Taylor expansion of $u$ gives

$$a_J u_J = a_I u_I + (J - I)h\frac{\partial}{\partial x}(au)(x_I, t_n) + O(h^2)$$

and thus

$$\lambda_J u_J = \lambda_I u_I + (J - I)\Delta t\frac{\partial}{\partial x}(au)(x_I, t_n) + O(\Delta t h).$$

Using this equality for $J$ from $I - 1$ to $I + 2$ in (48) gives

$$u_I' = u_I - \Delta t\frac{\partial}{\partial x}(au)(x_I, t_n) + \frac{1}{2}\left(\lambda_{I-1}^2 u_{I-1} - \lambda_I^2 u_I - \lambda_{I+1}^2 u_{I+1} + \lambda_{I+2}^2 u_{I+2}\right) + O(\Delta t h). \tag{49}$$

Finally, it is readily seen that

$$a_{I-1}^2 u_{I-1} - a_I^2 u_I - a_{I+1}^2 u_{I+1} + a_{I+2}^2 u_{I+2} = 2h^2\frac{\partial}{\partial x^2}(a^2 u)(x_i) + O(h^3).$$

Therefore:

$$u_I' = u_I - \Delta t\frac{\partial}{\partial x}(au)(x_I, t_n) + O(h^2 + \Delta t h),$$

which is consistent with one time-step for the advection Eq. (2).

In the case when $\lambda_{I-1} > 0$ the term $\gamma_{I-1}'$ in (47) has to be replaced by $\gamma_{I-1}$ but in this case $y_{I-1} = \lambda_{I-1}$ and the contribution to the grid point $I - 1$ is the same as in (48).

We now turn to the grid point $I + 1$. We obtain after remeshing the following value at time $t_{n+1}$:

$$u_{I+1}' = \gamma_{I+1}u_{I+1} + \beta_{I+2}u_{I+2} + \alpha_{I+3}u_{I+3} \tag{50}$$

From (10) we get

$$u_{I+1}' = \left(1 + \frac{\lambda_{I+1}}{2}(\lambda_{I+1} + 3)\right)u_{I+1} - \lambda_{I+2}(2 + \lambda_{I+2})u_{I+2} + \frac{\lambda_{I+3}}{2}(\lambda_{I+3} + 1)u_{I+3}. \tag{51}$$

By Taylor expansions of $au$ around the grid point $I + 1$ we get

$$u_{I+1}' = u_{I+1} - \Delta t\frac{\partial}{\partial x}(au)(x_{I+1}, t_n) + \frac{1}{2}\left(\lambda_{I+1}^2 u_{I+1} - 2\lambda_{I+2}^2 u_{I+2} + \lambda_{I+3}^2 u_{I+3}\right) + O(\Delta t h). \tag{52}$$

Finally, it is readily seen that

$$a_{I+1}^2 u_{I+1} - 2a_{I+2}^2 u_{I+2} + a_{I+3}^2 u_{I+3} = h^2\frac{\partial}{\partial x^2}(a^2 u)(x_{I+2}) + O(h^4)$$

and therefore

$$u_{I+1}' = u_{I+1} - \Delta t\frac{\partial}{\partial x}(au)(x_{I+1}, t_n) + O(h^2 + \Delta t h)$$

which is consistent with one time-step for the advection Eq. (2) at the grid point $I + 1$.

We now turn to the case (d'). Let us consider the grid point $I$. We first assume that $\lambda_{I-1} \geqslant 0$, and we get after remeshing the following value:

$$u_I' = \gamma_{I-1}u_{I-1} + (\beta_I + \gamma_I)u_I + (\alpha_{I+1} + \beta_{I+1})u_{I+1} + \alpha_{I+2}u_{I+2}. \tag{53}$$

We have $y_{I-1} = \lambda_{I-1}, y_I = \lambda_I, y_{I+1} = 1 + \lambda_{I+1}$ and $y_{I+2} = 1 + \lambda_{I+2}$, and we obtain after some calculation a formula identical to (48), and thus the same, consistent, finite difference approximation to the advection equation. In the case when $\lambda_{I-1} \geqslant 0$, on has to replace $\gamma_{I-1}$ by $\gamma_{I-1}'$. Since in this case $y_{I-1} = 1 + \lambda_{I-1}$ and $\gamma'(y) = \gamma(y - 1)$, we again find the same finite-difference formula.

Finally, on can notice that at the grid point $I + 1$ we obtain the same value in the case (d') as in the case (c').

## References

[1] M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, Multilevel adaptive particle methods for convection–diffusion equations, SIAM Multiscale Model. Simulat. 4 (2005) 328–357.
[2] M. Bergdorf, P. Koumoutsakos, A Lagrangian particle-wavelet method, SIAM Multiscale Model. Simulat. 5 (3) (2006) 980–995.
[3] M. Bergdorf, P. Koumoutsakos, A. Leonard, Direct numerical simulations of vortex rings at Re = 7,500, J. Fluid Mech. 581 (2007) 495–505.

[4] R. Cocle, G. Winckelmans, G. Daeninck, Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations, J. Comput. Phys. 227 (2008) 9091–9120.
[5] G.-H. Cottet, A. Magni, TVD remeshing formulas for particle methods, C.R. Math. 347 (23–24) (2009) 1367–1372.
[6] G.-H. Cottet, P. Poncet, Advances in direct numerical simulations of 3D wall-bounded flows by vortex-in-cell methods, J. Comput. Phys. 193 (2003) 136–158.
[7] G.-H. Cottet, B. Michaux, S. Ossia, G. Vanderlinden, A comparison of spectral and vortex methods in three-dimensional incompressible flows, J. Comput. Phys. 175 (2) (2002) 702–712.
[8] G.-H. Cottet, G. Balarac, M. Coquerelle, Subgrid particle resolution for the turbulent transport of a passive scalar, Adv. Turbul. XII, Springer Proc. Phys. 132 (2009) 779–782.
[9] G.-H. Cottet, P. Koumoutsakos, Vortex Methods: Theory and Practice, Cambridge University Press, 2000.
[10] G.-H. Cottet, E. Maitre, A level-set method for fluid-structure interactions with immersed surfaces, Math. Models Methods Appl. Sci. 16 (2006) 415–438.
[11] G.-H. Cottet, L. Weynans, Particle methods revisited: a class of high-order finite-difference schemes, C.R. Acad. Sci. Paris 343 (2006) 51–56.
[12] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (2002) 83–116.
[13] A. Ghoniem, D. Wee, Modified interpolation kernels for treating diffusion and remeshing in vortex methods, J. Comput. Phys. 213 (2006) 239–263.
[14] E. Godlewski, P.-A. Raviart, Numerical approximation of hyperbolic systems of conservation laws, Applied Mathamatical Science, vol. 118, Springer, 1996.
[15] S.E. Hieber, P. Koumoutsakos, A Lagrangian particle level set method, J. Comput. Phys. 210 (2005) 342–367.
[16] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.
[17] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627–665.
[18] A. Magni, PhD Thesis, Université de Grenoble, 2011. Available from: <http://tel.archives-ouvertes.fr/tel-00623128/fr/>.
[19] J.J. Monaghan, Extrapolating B-splines for interpolation, J. Comput. Phys. 60 (1985) 253–262.
[20] M.L. Ould-Salihi, G.-H. Cottet, M. Hamraoui, Blending finite-difference and vortex methods for incompressible flow calculations, SIAM J. Sci. Comput. 22 (5) (2000) 165–1674.
[21] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, M.S. Warren, Vortex Methods for Direct Numerical Simulation of Three-Dimensional Bluff Body Flows: Application to the Sphere at Re = 300, 500, and 1000, J. Comput. Phys. 178 (2002) 427–463.
[22] I.F. Sbalzarini, J.H. Walther, M. Bergdorf, S.E. Hieber, E.M. Kotsalis, P. Koumoutsakos, PPM – A highly efficient parallel particle-mesh library for the simulation of continuum systems, J. Comput. Phys. 215 (2006) 566–588.
[23] G. Strang, Trigonometric polynomials and difference methods of maximum accuracy, J. Math. Phys. 41 (1962) 147–520.
[24] W.M. van Rees, A. Leonard, D.I. Pullin, P. Koumoutsakos, A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers, J. Comput. Phys. 230 (8) (2011) 2794–2805.
[25] S. Vincent, G. Balmigère, J.-P. Caltagirone, E. Meillot, Eulerian Lagrangian multiscale methods for solving scalar equations application to incompressible two-phase flows, J. Comput. Phys. 229 (2010) 73–106.
[26] L. Weynans, Méthode particulaire multi-niveaux pour la dynamique des gaz, application au calcul d'écoulements multifluides, PhD thesis, Université de Grenoble, 2006, http://tel.archives-ouvertes.fr/tel-00121346/en/.