Chapter

A Monotonic Method of Split Particles

Yury Yanilkin, Vladimir Shmelev and Vadim Kolobyanin

Abstract

The problem of correct calculation of the motion of a multicomponent (multimaterial) medium is the most serious problem for Lagrangian–Eulerian and Eulerian techniques, especially in multicomponent cells in the vicinity of interfaces. There are two main approaches to solving the advection equation for a multicomponent medium. The first approach is based on the identification of interfaces and determining their position at each time step by the concentration field. In this case, the interface can be explicitly distinguished or reconstructed by the concentration field. The latter algorithm is the basis of widely used methods such as VOF. The second approach involves the use of the particle or marker method. In this case, the material fluxes of substances are determined by the particles with which certain masses of substances bind. Both approaches have their own advantages and drawbacks. The advantages of the particle method consist in the Lagrangian representation of particles and the possibility of" drawbacks. The main disadvantage of the particle method is the strong non-monotonicity of the solution caused by the discrete transfer of mass and mass-related quantities from cell to cell. This paper describes a particle method that is free of this drawback. Monotonization of the particle method is performed by spliting the particles so that the volume of matter flowing out of the cell corresponds to the volume calculated according to standard schemes of Lagrangian–Eulerian and Eulerian methods. In order not to generate an infinite chain of spliting, further split particles are re-united when certain conditions are met. The method is developed for modeling 2D and 3D gas-dynamic flows with accompanying processes, in which it is necessary to preserve the history of the process at Lagrangian points.

Keywords: Eulerian method, PIC method, numerical simulation, gas-dynamic

1. Introduction

Correct calculations of multi-material flows is the greatest challenge for ALE and Eulerian CFD codes, especially those using mixed cells at interfaces. There are two basic approaches to solving the advection equation for the multi-material case. In the first (grid-based) approach, interfaces are identified, and their position on the grid is tracked at each time step. The interface can be identified both explicitly, or it can be recovered based on the field of volume fractions. The latter algorithm serves as a basis for widely used methods, like the VOF method [1] (concentration method [2]). The second approach involves material particle methods first proposed by Harlow (the PIC method [3]). In this case, material fluxes from cells, including mixed ones, are controlled by particles, to which certain material masses are assigned.

Both approaches have their advantages and drawbacks. The advantages of the particle method consist in the Lagrangian representation of particles and the possibility of assigning material information to them. This minimizes the errors of solving the advection equation by the grid-based Eulerian methods. A number of modifications of the PIC method have been developed to improve its accuracy and extend the range of its applications [4–9]. An overview of these methods is provided in [10].

The central drawback of the particle method is the highly non-monotonic character of the solution caused by the discrete transfer of mass and mass-related quantities from cell to cell. The corresponding error can be reduced in the most straightforward manner by increasing the number of particles in cells, but such an increase limits the method's performance, especially in the 3D case. To minimize this drawback, a number of method modifications are employed. In [11, 12], for this purpose, the authors use particles having different masses. This approach, however, does not eliminate the need of involving a large number of particles. In a different approach, particles are used only in a limited part of the integration domain, for example, near interfaces [13]. As a result, only a small number of cells contain large quantities of particles. The rest part of the domain in this case is treated by the grid-based methods. Such a selective use of particles, however, does not eliminate the error of solving the advection equation by the grid-based methods and the need of remembering the history of a particular process in a large volume of the material.

This paper proposes a particle method that minimizes these drawbacks. Monotonization of the particle method is performed by particle splitting, so that the material volume flowing out of the cell corresponds to the volume calculated by schemes based on the grid approach. In order to prevent endless splitting, such split particles are further recombined under certain conditions. This approach allows us to do with a small number of particles in the cell, while delivering a monotonic solution.

2. Problem statement

The split-particle (SP) method has been implemented in a code called EGAK in the 2D approximation. In the source code, the major quantities for numerical solution of the multi-material gas dynamic equations include node-centered velocity vector components u_x and u_y and cell-centered thermodynamic quantities: density ρ_ξ, specific (per unit mass) internal energy e_ξ, and volume fractions $\beta_\xi = V_\xi/V$ of the constituent materials.

Particles can also be specified for some materials (in a particular case, these can be all materials). Each particle (with index p) has its coordinates in space $x_p(t)$, $y_p(t)$ and velocity vector components $u_{xp}(t)$, $u_{yp}(t)$ (these are used in interim calculations in the Lagrangian step). In addition, all particles represent thermodynamic states of the corresponding material (density, specific internal energy, volume): $\rho_{\xi p}$, $e_{\xi p}$, $V_{\xi p}$. Note that densities and volumes of particles can also give us their masses. Also note that in the method proposed particle velocities are obtained by interpolation between nodal velocities rather than "remembered" like in the classical PIC method.

Approximation of the corresponding equations is performed in two steps using a decomposition procedure. The **first** (Lagrangian) step involves calculations of the gas dynamic equations without convective members, i.e. gas dynamic equations in Lagrangian variables. In the **second** (Eulerian) step, a new grid is constructed (it generally coincides with the grid of the previous timestep), and the quantities are remapped onto it. As inputs, this step takes the outputs of the Lagrangian step. In turn, this step is divided into two sub-steps, i.e., approximation of the advection equation is done with decomposition in directions.

The difference equations below are presented in as much detail as needed to understand the algorithm of particle introduction; please refer to [14, 15] for a more detailed description of EGAK's basic difference scheme. In what follows, if no confusion is possible, no subscripts or subscript *n* are used to denote the outputs of the previous timestep, and subscripts n+1/2 and n + 1 denote the outputs of the Lagrangian and the Eulerian step, respectively.

3. Lagrangian step

3.1 Approximation of the cell-centered quantities

The Lagrangian gas dynamic equations are approximated using EGAK's standard scheme. As outputs, the Lagrange step delivers updated node-centered velocities, as well as densities, energies and volume fractions of each constituent material. This also applies to cells containing particles.

3.2 Definition of particle-specific quantities

In addition to the material-specific quantities, some particle-specific quantities are also defined for particles in the cells containing particles.

3.3 Updating of particle coordinates

Updated particle coordinates are found in two steps:

Step 1. At the Lagrangian step, particles are assumed to move together with the cell and inside the cell, without crossing its boundaries. The relative change in the particle position in the cell is associated with the difference in divergences (compression ratios) of different materials as a result of employing one closing model or another for the mixed-cell gas dynamic equations. In this study, we use only one assumption that the materials have equal divergences. This means that the sub-cell motion of particles does not change their position relative to the grid nodes.

Particle coordinates, $\tilde{x}_p^{n+1/2}$, $\tilde{y}_p^{n+1/2}$ are updated by bilinear interpolation between coordinates of cell nodes $x^{n+1/2}$, $y^{n+1/2}$, just as at time t^n.

Step 2. It is easy to show that the calculations of particle velocities by bilinear interpolation violate the law of conservation of momentum in the particle-containing cell. To ensure its conservation, the calculated particle velocities are corrected as follows:

1. Components of cell momentum are calculated:

$$P_{cx} = \frac{1}{4}\left(u_{x0}^{n+1/2} + u_{x1}^{n+1/2} + u_{x2}^{n+1/2} + u_{x3}^{n+1/2}\right) \cdot M, \quad (1)$$
$$P_{cy} = \frac{1}{4}\left(u_{y0}^{n+1/2} + u_{y1}^{n+1/2} + u_{y2}^{n+1/2} + u_{y3}^{n+1/2}\right) \cdot M.$$

Here, M is the cell mass and u_{xi}, u_{yi} (i = 0, 1, 2, 3) are the velocity vector components at four cell nodes.

2. Components of the total momentum of particles belonging to the cell are calculated:

$$P_{px} = \sum_p \tilde{u}_{xp}^{n+1/2} \cdot m_{p,}\ P_{py} = \sum_p \tilde{u}_{yp}^{n+1/2} \cdot m_{p.} \tag{2}$$

Here, the particle velocities are calculated using the particles' coordinates determined by bilinear interpolation and previous particle coordinates:

$$\tilde{u}_{xp}^{n+\frac{1}{2}} = \frac{\tilde{x}_p^{n+1/2} - x_p^n}{\tau}, \tilde{u}_{yp}^{n+\frac{1}{2}} = \frac{\tilde{y}_p^{n+1/2} - y_p^n}{\tau}, \tag{3}$$

were τ = t^{n+1}- t^n.

3. Coefficients $\lambda_x = P_{cx}/P_{px}$, $\lambda_y = P_{cy}/P_{py}$ are calculated.

The particles' velocities and coordinates are updated using the resulting weights:

$$u_{xp}^{n+1/2} = \lambda_x \cdot \tilde{u}_{xp}^{n+1/2}, u_{yp}^{n+1/2} = \lambda_y \cdot \tilde{u}_{yp}^{n+1/2}; \tag{4}$$

$$x_p^{n+1/2} = x_p^n + u_{xp}^{n+1/2} \cdot \tau, y_p^{n+1/2} = y_p^n + u_{yp}^{n+1/2} \cdot \tau.$$

3.4 Determination of particle velocity, density and energy

Changes in the relative density and energy of particles of a given constituent material are assumed to be equal to the corresponding relative changes in these quantities calculated for the respective material on average. This gives the following formulas:

$$\rho_{\xi p}^{n+1/2} = \rho_{\xi p}^n + \left(\rho_\xi^{n+1/2} - \rho_\xi^n\right)\rho_{\xi p}^n/\rho_\xi^n, \tag{5}$$

$$e_{\xi p}^{n+1/2} = e_{\xi p}^n + \left(e_\xi^{n+1/2} - e_\xi^n\right), \tag{6}$$

$$V_{\xi p}^{n+1/2} = V_{\xi p}^n\left(V_\xi^{n+1/2}/V_\xi^n\right). \tag{7}$$

It is easy to show that, when using (5)–(7), the particles' total masses will remain unchanged, and the particles' total internal energies will be equal to the energy calculated for the given material as a whole, i.e. the following relationships hold:

$$\rho_\xi^{n+1/2} V_\xi = \sum_p \rho_{\xi p}^{n+1/2} V_{\xi p}, e_\xi^{n+1/2} m_\xi = \sum_p e_{\xi p}^{n+1/2} m_{\xi p}. \tag{8}$$

4. Eulerian step

Major difficulties in implementing the particle method are associated with the Eulerian step, and namely, with calculations of mass and internal energy fluxes from cell to cell. In the PIC method, when a particle migrates to a neighbor cell, its mass and energy "migrate" with it. Because of the discrete (and, accordingly, non-monotonic) character of the transfer of mass and all the quantities defined per unit mass, this method delivers highly non-monotonic quantity profiles. Section 4 provides a detailed description of the monotonization algorithm for the PIC method.

In calculations, it is not always efficient to represent all constituent materials by particles because this requires extra calculations and computer memory. Therefore, it is reasonable to use particles for the constituent materials, for which the errors due to the solution of the advection equation are most essential, for example, for thin layers or materials, which require remembering the history of their Lagrangian particle.

As part of the proposed SP method, the following algorithms have been developed:

1. Interaction of materials described by particles and materials calculated by the code's standard scheme;
2. Support of existence, creation and removal of particles only in the vicinity of the interface;
3. Particles merging;
4. Remapping of particles density and energy to the cell as a whole.

These algorithms are listed in the order of their execution in the Eulerian step after the monotonization algorithm.

5. Monotonization algorithm for the particle method

5.1 One-dimensional case

Let us discuss the concept of the algorithm as applied to a one-dimensional flow for a single particle migrating from cell to cell (**Figures 1** and **2**). The figures show two cells containing particles represented by dots and imaginary boundaries of volumes represented by dashed segments. Note that calculations by this technique require only numerical values of the volumes, not their layout.

The flow is directed from left to right as indicated by velocity vector (**Figure 1**). Volume ΔV flowing out of the left cell (in what follows we call it the *volume flux*, darker color) is then equal to the product of the cell's lateral side length L and quantity S = u·τ:

$$\Delta V = L \cdot u \cdot \tau. \tag{9}$$

The non-monotonic behavior of the classical PIC method stems from the discrepancy between the real volume flux (and, accordingly, the mass flux) calculated by (9) and the volume of the particle crossing the cell side. In one case (**Figure 1a**), the volume moving from the left cell is smaller than the particle volume, and in the

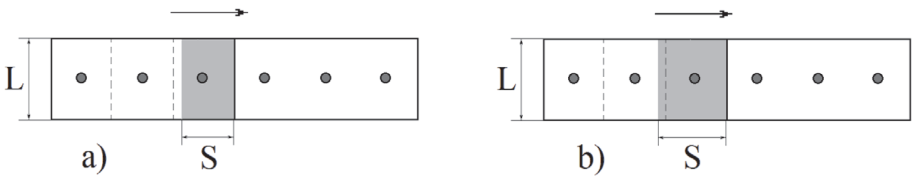

Figure 1.
Illustration of the reason for the non-monotonic behavior: a) volume flux is smaller than the particle volume; b) volume flux is larger than the particle volume.

other (**Figure 1b**), it is larger than the particle volume. In the 2D case and in the case of several migrating particles, the situation remains fundamentally the same.

Let us introduce the following notation:

$$\delta V = \Delta V\text{-}V_p, \tag{10}$$

where ΔV is the volume flux calculated by (4), V_P is the volume of particle p migrating from one cell into another.

Below we explain the monotonization algorithm for both of these cases.

1. *Volume flux ΔV is smaller than the migrating particle volume* ($\delta V < 0$). This case is illustrated in **Figure 2a**. Left is the state at $t^{n+1/2}$; right, at t^{n+1}. In this case, the particle migrating from the donor to the acceptor cell splits into two parts, a mother and a daughter particle. The mother particle migrates into the acceptor cell and now has new coordinates corresponding to its velocity and a new volume equal to the volume flux leaving the donor cell ΔV. The daughter particle, whose volume is equal to the difference between the initial particle volume and volume flux ΔV, is placed in the donor cell and acquires coordinates on the cell side. The link between the mother and the daughter particle is indicated by a broken line.

2. *Volume flux ΔV is larger than the migrating particle volume* ($\delta V > 0$). This case is illustrated in **Figure 2b**. In this case, the missing volume of the migrating particle must be made up by forced transfer of some particles or particle fragments from the donor cell to the acceptor. To be split is the particle lying next to the side of these cells and not yet transferred to the acceptor cell. It produces a daughter particle of volume δV, which migrates into the acceptor cell with donor-acceptor side coordinates.

If more than one cell migrates from cell to cell, formula (10) will take the form of

$$\delta V = \Delta V - \sum_p V_p, \tag{11}$$

where ΔV is the volume flux calculated by (9); V_P is the volume of the particle with index p migrating from cell to cell; summing is performed for all transferred particles.

Here, let us describe the differences from the algorithm described above.

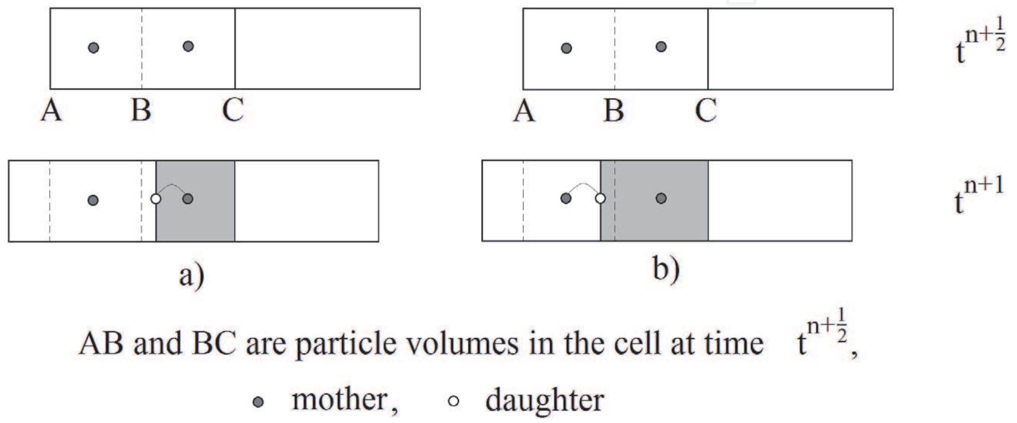
A
B
C
A
B
C
$t^{n+\frac{1}{2}}$
t^{n+1}
a)
b)
AB and BC are particle volumes in the cell at time $t^{n+\frac{1}{2}}$,
● mother, ○ daughter

Figure 2.
Illustration of the monotonization algorithm in the 1D case: a) δV < 0; b) δV > 0.

1. *The volume flux* ΔV *is smaller than the migrating particles' total volume* (δV < 0). In this case, to be split are all the migrating particles. The mother particles migrating into the acceptor cell have volumes

$$V_p^M = V_p\left(\Delta V/\sum_p V_p\right). \tag{12}$$

The daughter particles stay in the donor cell and have the following volume each:

$$V_p^D = V_p - V_p^M. \tag{13}$$

2. *Volume flux ΔV is larger than the migrating particles' total volume* (δV > 0). The particle staying next to the interface on the donor side is split to fill the remaining volume δV. Note that if $V_p < \delta V$, the mother particle's volume entirely goes to the daughter particles, and to be split is the next particle from the donor cell. If the donor cell is mixed, and the acceptor cell is pure and filled with the material present in the donor cell, the particles of this material will be split first.

5.2 Two-dimensional extension

Of particular interest in this case is the particle transition to the neighbor cell located diagonally from the donor cell (**Figure 3**). This case is special, because EGAK solves the advection equation using decomposition in directions, whereas no provision is made for diagonal cell-to-cell fluxes.

Consider the particular case depicted in **Figure 3**. Suppose only one node A moves to the new position B in the Eulerian step. The dashed lines in the figure show the locations of the cell sides, for which the grid node is a common vertex, at tn+1/2. C and D denote the points of intersection of straight lines AG and BE, and AF and BH, respectively.

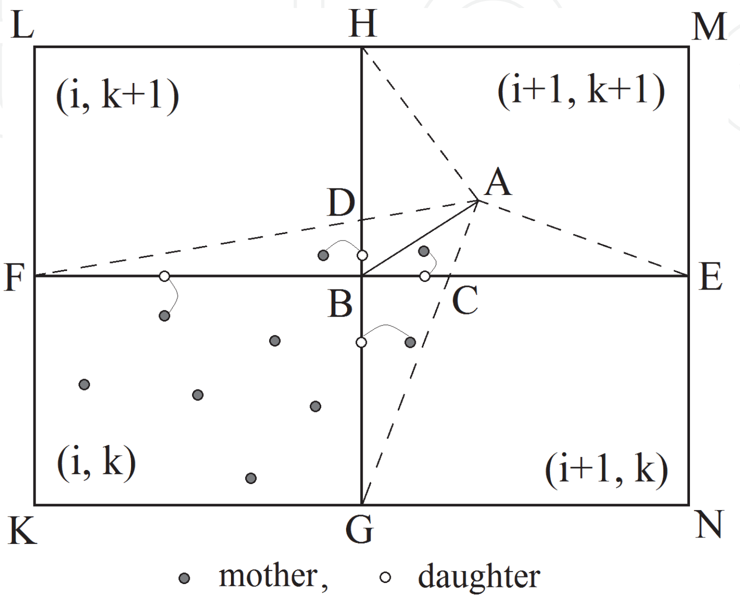

Figure 3.
Illustration of volume flux calculations in 2D case.

In EGAK, the cell-to-cell volume fluxes are defined as follows. The volume flux corresponding to triangle ABG (for short, *volume in triangle ABG*) is transferred from cell (i, k) to cell (i + 1, k), that is:

$$\begin{aligned} V_{i,k}^{n+1} &= V_{i,k}^{n+1/2} - V_{ABG}, \\ V_{i+1,k}^{n+1} &= V_{i+1,k}^{n+1/2} + V_{ABG}. \end{aligned} \tag{14}$$

Accordingly, the following relationships apply to all the cells under consideration including all the fluxes:

$$\begin{aligned} V_{i,k}^{n+1} &= V_{i,k}^{n+1/2} - V_{ABG} - V_{ABF} = V_{BFKG}, \\ V_{i+1,k}^{n+1} &= V_{i+1,k}^{n+1/2} + V_{ABG} - V_{ABE} = V_{BGNE}, \\ V_{i,k+1}^{n+1} &= V_{i,k+1}^{n+1/2} - V_{ABH} + V_{ABF} = V_{BFLH}, \\ V_{i+1,k+1}^{n+1} &= V_{i+1,k+1}^{n+1/2} + V_{ABE} + V_{ABH} = V_{BHME}. \end{aligned} \tag{15}$$

Note that in accordance with (15) the volume of triangle ABC is included in the volume of cell (i + 1,k) twice – as part of triangles ABG and ABE – but in one case it is positive, and in the other, negative. Thus, in fact it is not included in the updated volume of this cell; but it will be included in the volume of cell (i + 1,k + 1). The same applies to the volume of triangle ABD, which will be included in the volume of cell (i + 1,k + 1) and not included in the volume of cell (i,k + 1).

In accordance with the above, when considering particle contributions, flux calculations between cell (i,k) and its non-diagonal neighbors assume that the particle lying in triangle ABC migrates into cell (i + 1,k), and the particle lying in triangle ABD, into cell (i,k + 1). Then, in calculations of the flux between cells (i + 1, k), (i + 1,k + 1) and (i,k + 1), (i + 1,k + 1), these particles migrate into cell (i + 1, k + 1). Therefore, when considering this process in terms of flux monotonicity, corresponding daughter particles are introduced as shown in **Figure 4**. Note that the mother particle's position during the flux calculations is nevertheless defined in the true acceptor cell (i + 1,k + 1).

The particle splitting is based on the following principles:

- Both particles produced from the particle being split share its thermodynamic state (to comply with the conservation laws);
- The index assigned to the daughter particle is the same as the index of its mother particle, which also indicates that the particle is a daughter;
- The mother particle "knows" nothing about its daughter particles;

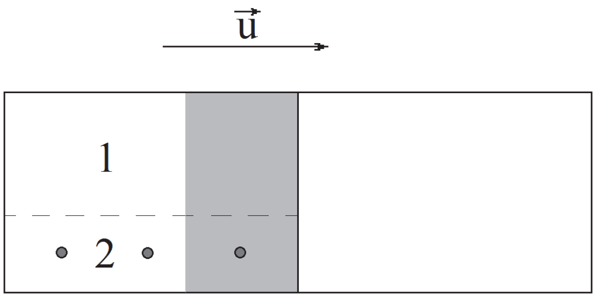

Figure 4.
Volume flux from a mixed cell.

- One mother particle may have several daughter particles;
- One daughter particle may have only one mother particle;
- The daughter particles may also split, but all the subsequent daughter particles will "remember" the index of their initial mother particle;
- The daughter particles are placed at the common donor/acceptor side to ensure their quickest possible combination with their respective mother particles.

6. Algorithm of interaction between particles and particle-free materials

The algorithm for volume flux calculations has a modification to deal with mixed cells containing materials with and without particles.

Consider the case of a cell filled with heterogeneous materials, one described only by grid quantities, and the other, by particles (1 and 2, respectively, in **Figure 4**). Suppose we need to divide the flux moving from left to right (shown with a darker color) between the materials. The volume flux leaving the cell is first filled with the volume of migrating particles. If the migrating particles' total volume exceeds this volume flux, then the above particle splitting algorithm is engaged (see Section 4, case $\delta V < 0$).

Otherwise, the missing part of the outflowing volume flux is filled with the particle-free material. If there are several particle-free materials, the volume is distributed among them based on the VOF algorithm [1]. If the particle-free materials are still not enough, the remaining volume flux is filled with particles using the splitting algorithm for the case of $\delta V > 0$ from Section 4.

7. Near-interface algorithms

As part of the proposed method, we have developed an algorithm involving the particles located only near the interface. The region near the interface includes mixed cells and one layer of adjacent pure cells of each material on each side.

Figure 5 shows possible particle layouts relative to the interface. The dark and light cells (**Figure 5a**) are pure, and the intermediate-color cell (**Figure 5b**) is

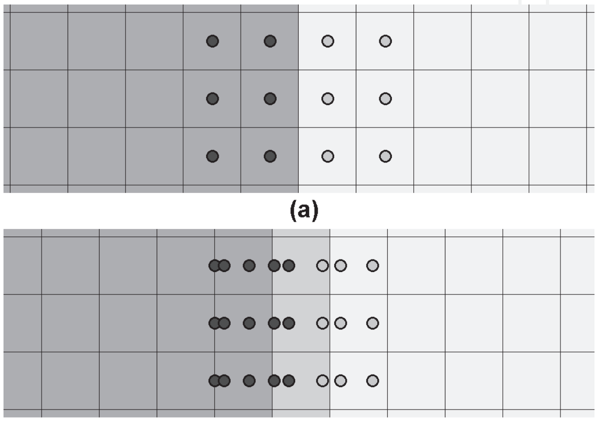

Figure 5.
Particle layout near the interface: a) t = 0, b) t > 0.

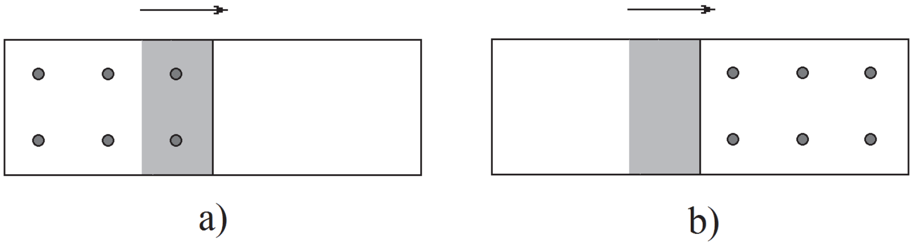

Figure 6.
Conflicting cases of two representations of the same material: a) volume flux from a cell with particles to a particle-free cell; b) volume flux from a particle-free cell to a cell with particles.

mixed. The particles in the cells are marked with a contrasting color. **Figure** 5, a illustrates the case when both materials are represented by particles only near the interface at t = 0, and **Figure 5b** shows the particle layout during the computation.

Thus, if the material is represented by particles only near the interface, then the same material can be represented both by particles and without particles depending on the interface location. To simplify the algorithm, the same material described by particles and without particles cannot occupy the same cell. This condition for each cell is formally given by

$$V_{\xi} = \sum_{\xi p} V_{\xi p}, \text{or} \sum_{\xi p} V_{\xi p} = 0. \tag{16}$$

Let us consider two reasons leading to the case when two different representations of the same material are present in the same cell. **Figure 6a, b** shows cells filled with the same material. In each case, however, in one of the cells the material is represented by particles (black dots), and in the other, without particles. The darker color shows the volume flux relative to the cells' total volume; the arrow indicates its direction. Below we describe the unwanted cases and the ways to avoid them.

Figure 6a shows a volume flux from a cell with particles to a particle-free cell. In this case, the particle splitting algorithm described in Section 4 stays unchanged in the donor cell, but the particles that were supposed to migrate into the acceptor cell are removed with an update of the thermodynamic state, and the particles staying in the donor cell become ordinary (they are no daughter cells any more if they were). **Figure 6b** shows a volume flux from a particle-free cell to a cell with particles. In this case, the acceptor cell receives a particle, the volume of which is equal to the volume flux and the state of which is the same as the donor-cell material parameters. The added particle then immediately combines with one of the particles in the acceptor cell. The combination rules are given below.

To preserve the layout, where particles are present only near the interface, the particles lying beyond this region are removed from the cells and new particles are created in the cells appearing in the region near the interface.

8. Particle combination algorithm

To balance the particle splitting algorithm (Section 4), we have developed a particle combination procedure. The latter serves to prevent uncontrolled multiplication of particles as a result of their splitting.

Two particles of the same material within the same cell must be combined if one of the following criteria is met:

- One of the particles is a daughter of the other one;

- Two daughters have the same mother;
- The particles have close (to within a constant) coordinates;
- The particle number exceeds the maximum number specified for the cell;
- One of the particles has a relatively small volume.

Particle combination rules:

- If a daughter is combined with its mother, the resulting particle inherits the mother's coordinates;
- If two daughters of the same mother are combined (p_1 and p_2), the coordinates of the resulting particle (p) are chosen in accordance with their mass ratio:

$$\begin{aligned}\tilde{x}_p &= x_{p1} + (x_{p2} - x_{p1})(m_{p2}/(m_{p1} + m_{p2})), \\ \tilde{y}_p &= y_{p1} + \left(y_{p2} - y_{p1}\right)(m_{p2}/(m_{p1} + m_{p2}));\end{aligned} \tag{17}$$

- The parameters of the resulting particle are calculated subject to the laws of conservation of their mass, specific internal energy and volume:

$$\begin{aligned}\tilde{e}_p &= \frac{e_{p1}\rho_{p1}V_{p1} + e_{p2}\rho_{p2}V_{p2}}{\rho_{p1}V_{p1} + \rho_{p2}V_{p2}}, \\ \tilde{\rho}_p &= \frac{\rho_{p1}V_{p1} + \rho_{p2}V_{p2}}{V_{p1} + V_{p2}}, \\ \tilde{V}_p &= V_{p1} + V_{p2}.\end{aligned} \tag{18}$$

9. Particle-to-cell density and energy remapping algorithm

For each cell containing particles, the quantities are remapped as follows:

$$\begin{aligned}\rho_\xi^{n+1} &= \sum_p \rho_{\xi p} V_{\xi p} / \sum_p V_{\xi p}, \\ e_\xi^{n+1} &= \sum_p e_{\xi p} \rho_{\xi p} V_{\xi p} / \sum_p \rho_{\xi p} V_{\xi p},\end{aligned} \tag{19}$$

where summing is performed for the particles of material ξ in the cell.

10. Method testing

10.1 Test problem 1. A moving cruciform density discontinuity

Domain $0 < x < 12$, $0 < y < 12$ is divided into two subdomains (0 and 1). In subdomain 0: $\rho_0 = 1$, $e_0 = 0$, $u_x = 1$, $u_y = 1$, no particles are specified; in subdomain 1: $\rho_0 = 10$, $e_0 = 0$, $u_x = 1$, $u_y = 1$, each cell contains one particle. P = 0 all over the domain, so the problem involves virtually no gas dynamics, only convective flow. The calculations were performed on a fixed grid of 60x60 cells.

Results calculated by the SP method and EGAK (in what follows, SP is the particle method with flux correction described above, PIC is the correction-free particle method similar to the PIC method, and EGAK is used to denote the particle-free code) are shown in the form of density distributions at t = 7.08 (**Figure** 7). The figure shows that the result produced by the SP method is exact.

10.2 Test problem 2. A one-dimensional steady shock wave

We consider the following 1D problem statement. Domain $0 < x < 50$, $0 < y < 4$ is occupied by an ideal gas with $\rho = 1$, $P = 0$, $u = 0$, $\gamma = 3$. A plane shock wave propagates in the material from left to right. Its parameters behind the shock front are $\rho = 2$, $e = 2$, $P = 8$, $u = 2$. The calculations were performed on a fixed grid of 100x4 cells. In the PIC and SP calculations, there were four particles in each cell.

Figure 8 shows the plots of density as a function of coordinate at t = 10 calculated by the SP, PIC and EGAK methods. One can see that the SP method gives nearly the same result as EGAK and a better result compared to PIC. The exact shock position is x = 40.

10.3 Test problem 3. A point explosion

Domain $0 < x < 20$, $0 < y < 20$ contains two materials: a circle of radius 0.1 with its center at the origin is occupied by an ideal gas with $\rho = 1$, $e = 1$, $P = 0$, $\gamma = 1.4$; the remaining part is occupied by an ideal gas with $\rho = 1$, $e = 0$, $P = 0$, $\gamma = 1.4$.

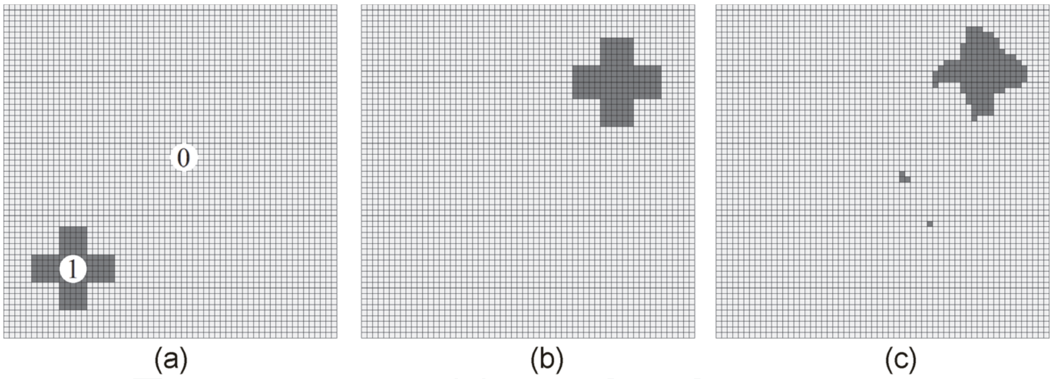

Figure 7.
Test problem 1. Density distributions: a) t = 0; b) t = 7.08, method SP; c) t = 7.08, method EGAK.

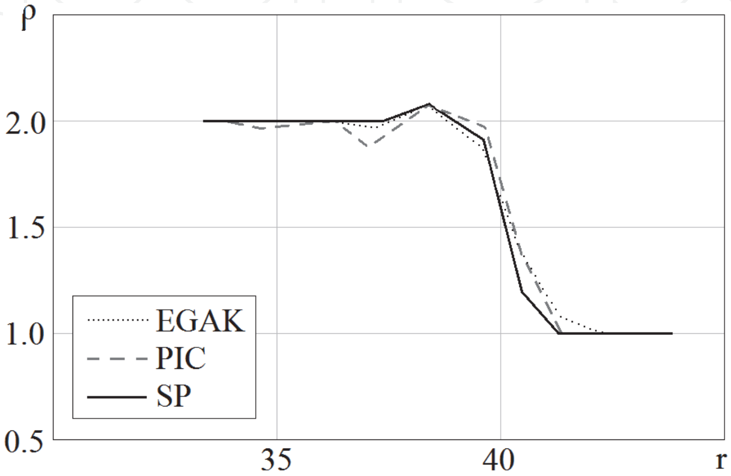

Figure 8.
Test problem 2. Density as a function of shock position, t = 10.

Calculations were performed by the SP, PIC and EGAK methods. In the SP and PIC calculations, each cell initially contained four particles. In the SP problem calculation, the number of particles in the region with gas increased because of its strong expansion. The calculations were done on a fixed regular grid of 100x100 cells.

The SP calculation results are presented in **Figure 9** as a density distribution at t = 100. **Figure 10** shows plots of density as a function of radius for all the cells in the domain occupied by the shock wave at a given instant. They demonstrate how efficient the methods are in preserving the flow's spherical symmetry. **Figure 11** shows plots of density as a function of radius for section x = y.

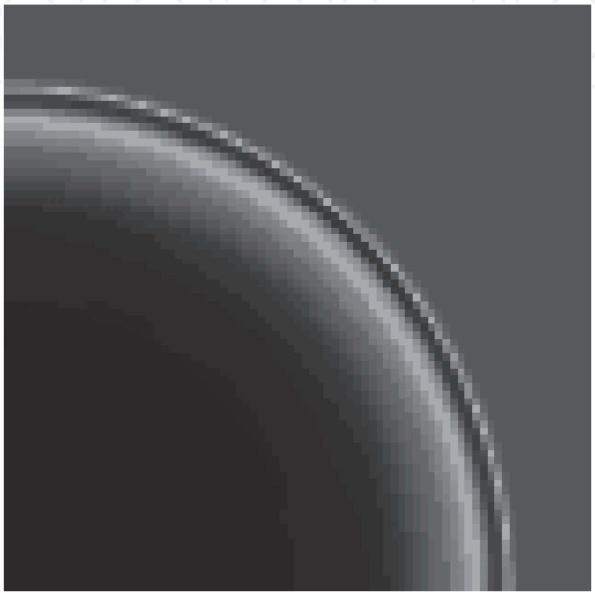

Figure 9.
Test problem 3. Density distribution at t = 100.

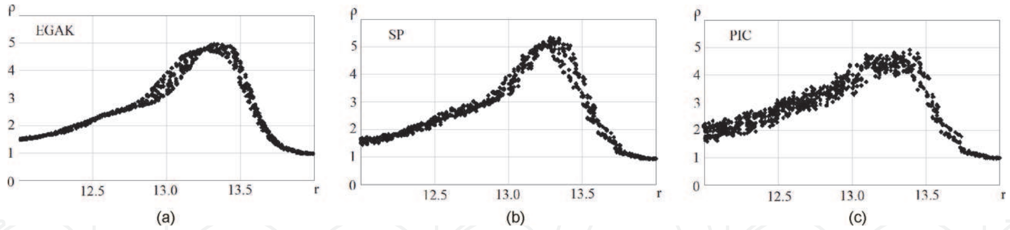

Figure 10.
Test problem 3. Density across the cells as a function of radius, t = 100: a) EGAK; b) SP; c) PIC.

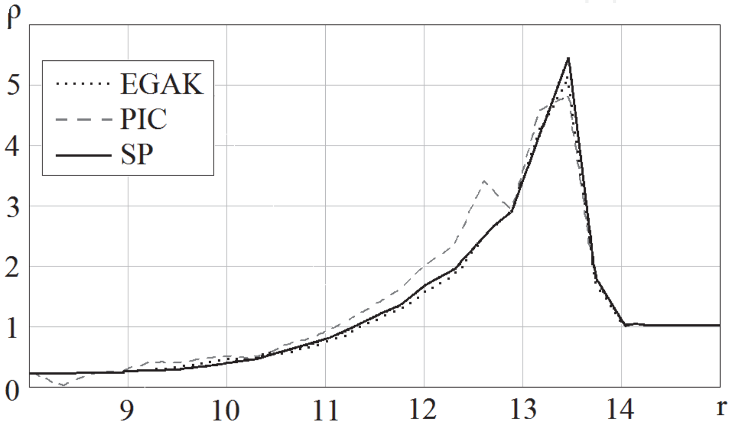

Figure 11.
Test problem 3. Density as a function of radius in section x = y, t = 100.

These figures demonstrate that the SP result is again nearly as good at the EGAK result and noticeably more accurate than the PIC result.

10.4 Test problem 4. A spherically converging shell

The initial problem geometry is borrowed from paper [16] and shown in **Figure 12**. Domain 1: $R_1 = 0.8$, $\rho_0 = 0.01$, $e_0 = 0$, $u_0 = 0$, ideal gas with $\gamma = 5/3$. Domain 2: $R_2 = 1$, $\rho_0 = 10$, $e_0 = 0$, $U_0^R = -1$, equation of state of Mie-Grueneisen type with constants $\rho_0 = 10$, $c_0 = 4$, $n = 5$, $\gamma = 2$. Boundary condition at R2: pressure $P = 0$. Domain 3: vacuum with $P = 0$. Units of measurement: ρ - [g/cm^3], t - [10 μs], L - [cm]. The calculations were done on a fixed regular grid of 110x110 cells. The SP calculations involved particles in both gas and shell domains (one particle per cell, with a limitation of no more than four particles). Their results are compared with the EGAK results. An interesting feature of this problem is that the number of particles in the computation constantly decreases, because both materials are compressed.

The main target result in this problem is maximum gas compression. Note that the reference density obtained in convergence calculations by the 1D method [17] is ≈25. The maximum average gas density and respective time for SP and EGAK are 16.03 at $t = 0.368$ and 16.49 at $t = 0.369$, respectively. As an illustration, **Figure 13a** shows a fragment of the domain with particles at $t = 0.368$. The figure also shows density values across the cells of the compressed-gas domain from the calculations by SP (**Figure 13b**) and EGAK (**Figure 13c**).

The maximum gas compression ratios and their respective time obtained by EGAK and SP are close, but the maximum compression ratios are much lower than the reference solution, which is explained by a small number of cells in these calculations (the solutions converge to the reference solution with mesh size). A smaller compression ratio in the SP calculation compared to EGAK is attributed to the presence of gas spots "split-off" from the main domain in the SP calculation. As for the flow symmetry preservation, SP is nearly as good as EGAK.

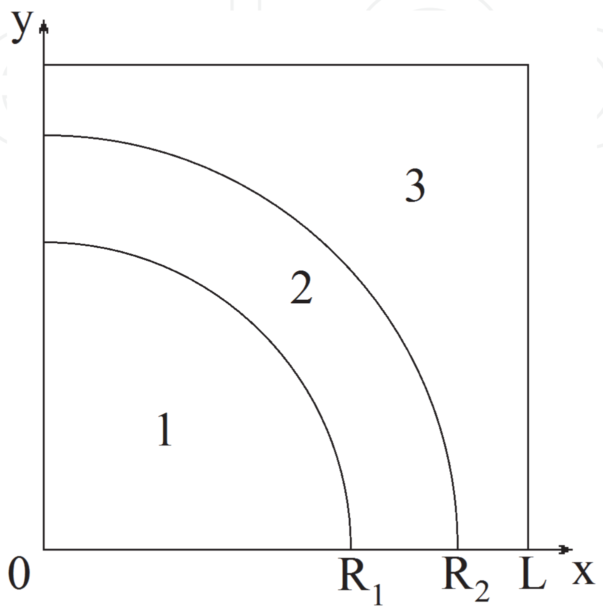

Figure 12.
Geometry of test problem 4.

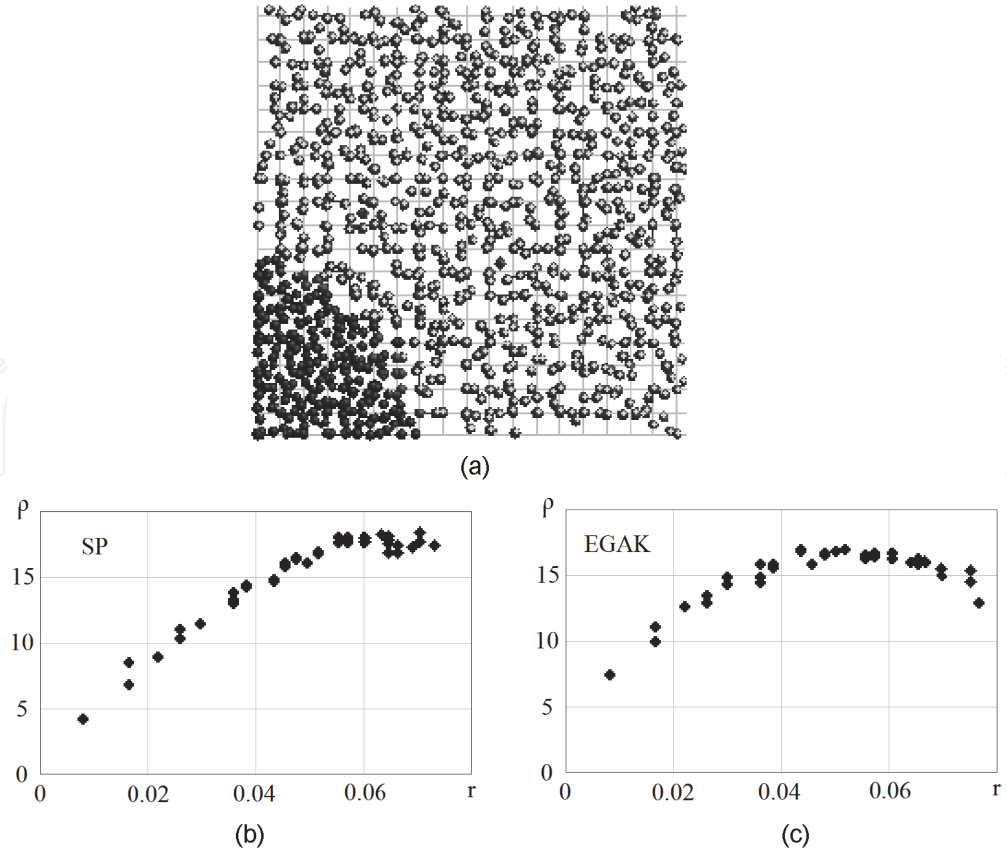

Figure 13.
Test problem 4. a) Particle positions in the central domain; b) Density value across the gas cells calculated by SP, and c) Density value across the gas cells calculated by EGAK.

11. Conclusion

The paper describes a monotonic split-particle method. The method has been developed to simulate multi-material gas dynamic flows using a combination of grid methods implemented in EGAK and the SP method for some layers. The calculations demonstrated that the SP method is close to EGAK in the accuracy of shock-capturing simulations and is much more accurate as applied to convective flow simulations, like the PIC method. At the same time, the SP method is free of the major drawback of the PIC method, namely the severe nonmonotonicity of its solution due to the discrete mass transfer. In addition, this method uses a relatively small number of particles.

Further prospects of the SP method are related to its application to the problems that require "remembering" the process history at Lagrangian points, like detonation and combustion of explosives, elastoplastic behavior and fracture of materials, etc. In particular, implemented to date have been the kinetic model of explosive HE transformation by Morozov and Karpenko [18] and the model of materials fracture by Kanel et al. [19]. This method has also been implemented in the 3D extension of the EGAK code.

Acknowledgements

We would like to thank Tatiana Zezyulina for the translation of this paper.

Author details

Yury Yanilkin*, Vladimir Shmelev and Vadim Kolobyanin
Russian Federal Nuclear Centre, VNIIEF, Sarov, Russia

*Address all correspondence to: n.yanilkina@mail.ru

References

[1] Hirt CW. Nicols BD. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. Journal of Computational Physics. 1981; 39: 201–225.

[2] Bakhrakh SM, Glagoleva YuP, Samigulin MS, Frolov VD, Yanenko NN, Yanilkin YuV. Gas dynamic flow simulations by the concentration method. DAS USSR. 1981; **257 (3)**: 566–569. (In Russian).

[3] Harlow FH. The particle-in-cell computing methods for fluid dynamics. Meth. Comput. Phys. 1964; 3: 319–343.

[4] Tskhakaya D, Matyash K, Schneider R, Taccogna F. The Particle-In-Cell Method. Contributions to Plasma Physics. 2007; 47(8–9): 563–594.

[5] Shalaby M, Broderick AE, Chang P, Pfrommer C, Lamberts A, Puchwein E. SHARP: A Spatially Higher-order, Relativistic Particle-in-Cell Code. The Astrophysical Journal. 2017; 841(1): 52.

[6] Jiang C, Schroeder C, Selle A, et al. The affine particle-incell method. ACM Trans. Graph. 2015; 34:4.

[7] Bogomolov SV, Zvenkov DS. Explicit particle method, nonsmoothing gas-dynamic discontinuities. Matematicheskoe modelirovanie. 2007; 19:3: 74–86. (In Russian).

[8] Jiang C, Schroeder C, Teran J. An angular momentum conserving affine-particle-incell method. Journal of Computational Physics. 2017; 338: 137–164.

[9] Fu C, Guo Q, Gast T, et al. A polynomial particle-in-cell method. ACM Trans. Graph. 2017; **36**: 222:1–222:12.

[10] Grigoryev YuN, Vshivkov VA, Fedoruk MP. Numerical "Particle-in-Cell" Methods. Theory and Applications. Utrecht Boston. 2002.

[11] Lapenta G, Brackbill JU. Dynamic and selective control of the number of particles in kinetic plasma simulations. Journal of Computional Physics. 1994; 115: 213–217.

[12] Welch DR, Genoni TC, Clark RE, Rose DV. Adaptive particle managment in a Particle-in-Cell Code. Journal of Computional Physics. 2007; 227: 143–155.

[13] Sapojnikov GA. A combine method of fluid fluxes and particle-in-cell for calculations of the gasdynamical flows. In "Voprocy razrabotki I ekspluatacii paketov prikladnyx program". Novosibirsk: ITPM SO AN SSSR. 1981; 89–97. (In Russian).

[14] Yanilkin YuV, Belyaev SP, Bondarenko YuA, et al. EGAK and TREK: Eulerian codes for multidimensional multi-material flow simulations. RFNC-VNIIEF Transactions. Research Publication, Sarov: RFNC-VNIIEF. 2008; **12**: 54–65. (In Russian).

[15] Yanilkin YuV, Goncharov EA, Kolobyanin VYu, Sadchikov VV, Kamm JR, Shashkov MJ, Rider WJ. Multi-material pressure relaxation methods for Lagrangian hydrodynamics. Computers & Fluids. 2013; 83: 137–143.

[16] Yanilkin YuV, Toporova OO. Two-dimensional scalar artificial viscosity of the EGAK code in spherical systems. VANT, Series MMFP. 2010; 3: 46–54. (In Russian).

[17] Bondarenko YuA. The order of approximation, the order of numerical convergence and cost efficiency of Eulerian multi-dimensional gas dynamics computations illustrated by "blast waves" problem simulations. VANT, Series MMFP. 2004; 4, 51–61. (In Russian).

[18] Morozov VG, Karpenko II, Kuratov SE, Sokolov SS, Shamraev BN, Dmitrieva LV. Theoretical Substantiation of the Phenomenological Model of HE Sensitivity to Shock Waves Basing on TATB. Chemical Physics. 1995; **14 (2–3)**: 32–37. (In Russian).

[19] Kanel GI, Pazorenov SV, Utkin AV, Fortov VE. Studies of mechanical response of materials to their shock-wave loading. Izvestiya RAN. MTT. 1999; 5:173–188. (In Russian).