A discontinuous particle method for the inviscid Burgers’ equation

S V Bogomolov¹, M A Filippova¹ and A E Kuvshinnikov²

¹Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Leninskiye Gory, 1-52, Moscow, 119991, Russia
²Keldysh Institute of Applied Mathematics RAS, Miusskaya sq., 4, Moscow, 125047, Russia

E-mail: kuvsh90@yandex.ru

Abstract. The discontinuous particle method for simple problems associated with gas dynamics is under consideration. The origin of the method is based on the micro-model describing the movement of particles with prescribed velocities. We show that with the micro-model the inviscid Burgers’ equation is solved in a weak sense. Numerical experiments have confirmed a low viscosity of the method: the solution is smeared by only one particle.

1. Introduction

Discrete methods for simulations can be divided into the three classes: finite difference methods, finite element methods, and particle methods [1, 2]. Usually, the particle method is considered to be a representation of unknown function as a set of δ-functions. And then δ-functions are approximated by classical functions with which calculations are performed. The particle method was created to solve the gas dynamics problems [3], and it was successful for the plasma physics problems [1].

The particle methods, in turn, can also be divided into the three subclasses [1]: particle-mesh (PM), particle-particle (PP) and hybrid particle-particle–particle-mesh (PPPM or P³M). The PM are cheaper, the PP are more costly but more accurate.

One of the first PM methods was the particle-in-cell (PIC) method [3, 4]. In the PIC, particles have only mass and position. The remaining values are calculated in the grid cells. The disadvantage of this method is a high dissipation. To overcome this problem, a fluid-implicit-particle method (FLIP) was created [5, 6], where the particle velocity was calculated without interpolation. However, this method was unstable in some problems. Later, the affine particle-in-cell method (APIC) was developed [7, 8], eliminating these drawbacks. Also, the method of the polynomial particles-in-cells (PolyPIC) was created [9], which is an improvement of the previous method.

Most often, to solve the gas dynamics problems with the PP methods, the smoothed particle hydrodynamics (SPH) method [10] is used. This method has undergone numerous modifications aimed at increasing the accuracy and productivity [11]. It is also worth noting the particle finite element method (PFEM) [12] and the finite volume particle method (FVPM) [13, 14]. These methods are widely used. The particle methods were also studied in the Soviet Union and Russia [15–20].

We propose a novel discontinuous particle method as another version of the PP method with a high accuracy which manifests by its low viscosity: the discontinuous solution is spread over only one cell. Various versions of our particle method have been developed: with splitting, with a choice of the time
step, for one-dimensional problems of gas dynamics, for two-dimensional problems of an incompressible fluid [21–23]. The origin of the method is based on the micro-model describing the movement of particles with prescribed velocities. This viewpoint makes it possible to construct “through” algorithms for the micro – meso – macro modeling [24].

2. The discontinuous particle method

2.1. Moving from a micro model to a macro model

Let us introduce a micro model and consider its transition to a macro model.

The Cauchy problem for a system of ODEs describes the motion of N material points with given velocities $v_i(x, t)$:

$$\frac{dx_i(t)}{dt} = v_i(x_i(t), t),$$

$$x_i(0) = x_i^0, \quad i = 1, \ldots, N,$$

where $x_i(t)$ is the trajectory of the material point with the initial position $x_i^0$.

We determine the distribution density $u(x, t)$:

$$u(x, t) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i(t)), \quad (2)$$

or, more precisely,

$$\forall \varphi(x) \in C_0^2 : \int \varphi(x) u(x, t) dx = \frac{1}{N} \sum_{i=1}^{N} \varphi(x_i(t)).$$

In [25], it is shown that the distribution density $u(x, t)$ is a solution of the Cauchy problem for the generalized transport equation:

$$\frac{\partial u(x, t)}{\partial t} + \frac{\partial [v(x, t)u(x, t)]}{\partial x} = 0,$$

$$u(x, 0) = u_0(x), \quad (3)$$

where $u_0(x)$ is the initial density. That is, if the coordinates of the points change according to the system of equations (1), then the density $u(x, t)$ is a generalized solution of the Cauchy problem for the transport equation (3). This means that while solving (1) we solve equation (3).

2.2. The particle method algorithm

To get a classical solution $\bar{u}$, we have to approximate a set of $\delta$-functions by a set of classical ones $\Pi_i$:

$$\bar{u} = \sum_{i=1}^{N} \Pi_i(x_i).$$

Thus, the function $u(x, t)$ is replaced by a set of rectangle particles $\Pi(x, t)$ whose centers (the coordinates of particles) satisfy the equations:

$$\frac{dx_i}{dt} = v_i,$$

$$x_i(0) = x_i^0, \quad i = 1, \ldots, M.$$
The main feature of our method is a proper approximation of each delta-function from this set taking into account their neighbors. We present this algorithm a bit later while describing the corrector stage of the method.

Now we extend the definition (2) by:

$$u(x,t) = \sum_{i=1}^{M} c_i(t) \delta(x - x_i(t)),$$

where $c_i(t)$ are the weights, $x_i(t)$ are the coordinates of macro particles. The results obtained for ordinary (with the same weights $1/N$) particles will also be true for macro particles. For a numerical solution, the function $u(x,t)$ will be approximated by a set of $M$ rectangular particles with the width $\Delta x_i$. Then:

$$c_i(t) = u(x_i(t), t) \Delta x_i.$$

We describe an algorithm for applying the particle method to one-dimensional problems. To implement the particles, introduce the following notation: $x_i^k$ is the coordinate of the center of the $i$-th particle at the $k$-th time moment, $v_i^k$ is the velocity of a particle, $h_i^k$ is the height (density) of a particle, $w_i^k$ is the width of a particle, $S_i = h_i^k w_i^k$ is the area of a particle. We assume that the area (mass) of the particles is constant which means that we consider the conservative transport: the particles do not exchange their masses.

We define the particle parameters at the initial time moment. To do this, we divide the initial domain into a finite number of segments, each taken as the width of the particle $w_i^0$. For the coordinates of the center of a particle we take the coordinates of the center of the segments, and for the height of the particles we take values of the given function $u^0(x)$ at the centers of the particles $h_i^0 = u^0(x_i^0)$.

The algorithm is based on the predictor-corrector principle. First, we solve the ODE system using an explicit Euler scheme:

$$x_i^{k+1} = x_i^k + \tau v_i^k, \ i = 1, ..., N.$$  

After moving the particles, intersections or gaps may appear between them. At the corrector stage, it is necessary to change the particle widths so that the intersections and gaps are minimized. We keep the particles symmetric about the center. Figure 1 shows the possible cases of the particles interactions.

- A particle with a higher density “bumps” into a particle with a lower density. We remove an intersection by narrowing the particle with a lower density. Since the area of the particle remains unchanged, when the width of the particle decreases, its height increases.

- A particle with a higher density “flies away” from a particle with a lower density. We remove the resulting gap, expanding the particle with a higher density. With such an expansion, the particle height decreases.

After changing the width and the height, the particle can again form an intersection or a gap, but with another particle. We ignore such interactions because we are aimed at an explicit algorithm.

It is worth noting that the number of particles is constant. But particles move and, for comparison with the analytical solution, particles are initially set in a larger interval than the one shown in the figure.
Figure 1. Possible versions of particles interactions.

We write down the correction condition for the i-th particle as follows:

\[ (h_i^k - h_{i+1}^k) \left( x_{i}^{k+1} - x_{i-1}^{k+1} - \frac{w_i^k + w_{i+1}^k}{2} \right) > 0. \]

We change the width of the i-th particle so that it adjoins the neighboring one. The height is calculated from the condition of a constant particle area.

\[ w_i^{k+1} = 2 \left( x_i^{k+1} - x_{i-1}^{k+1} - \frac{w_i^k}{2} \right), \]
\[ \nu_i^{k+1} = h_i^{k+1} = \frac{S_i}{w_i^{k+1}}. \]

3. The 1D inviscid Burgers’ equation

Particle coordinates satisfy the following equations:

\[ \frac{dx_i}{dt} = \frac{u_i}{2}, \quad i = 1, \ldots, N, \]
\[ x_i(0) = x_i^0. \]

Moving to the Π-shaped functions, we obtain:

\[ \frac{d\bar{x}_i}{dt} = \frac{\bar{u}_i}{2}, \quad i = 1, \ldots, N, \]
\[ \bar{x}_i(0) = \bar{x}_i^0. \] 

Let us consider the discontinuous initial condition:

\[ u_0(x) = \begin{cases} a, & x \leq x_0, \\ b, & x > x_0. \end{cases} \]
If $a > b$, the analytical solution looks like [26]:

$$u(x,t) = \begin{cases} 
    a, & x \leq x_0 + Dt, \\
    b, & x > x_0 + Dt,
\end{cases}$$

(6)

where $D = (a + b)/2$ is a discontinuity movement speed.

We use $a = 1.0$, $b = 0.5$, $x_0 = 0.8$. Figure 2 presents a comparison of the numerical solution of the Cauchy problem (4) with the initial condition (5) with a step $\tau = 0.05$ obtained by the discontinuous particle method and the analytical solution (6) at the time point (a) $T = 0.0$ and (b) $T = 2.5$.

**Figure 2.** Comparison of the numerical solution (rectangles) and the analytical solution (red line) of the Cauchy problem for a 1D inviscid Burgers’ equation.

The analysis of Figure 2 shows that there are no oscillations in the numerical solution. We can also see that the solution is smeared by only one particle. This shows a very low viscosity of the method.

4. **The 2D inviscid Burgers’ equation**

Let us consider the following microscopic Cauchy problem for a 2D case:
\[ \frac{dx_i^u(t)}{dt} = u(x_i^u(t), y_i^u(t), t), \quad i = 1, ..., N_u; \]
\[ \frac{dy_i^u(t)}{dt} = v(x_i^u(t), y_i^u(t), t), \quad i = 1, ..., N_u; \]
\[ \frac{dx_i^v(t)}{dt} = u(x_i^v(t), y_i^v(t), t), \quad i = 1, ..., N_v; \]
\[ \frac{dy_i^v(t)}{dt} = v(x_i^v(t), y_i^v(t), t), \quad i = 1, ..., N_v; \]
\[ x_i^u(0) = x_i^{u_0}, \quad y_i^u(0) = y_i^{u_0}, \quad i = 1, ..., N_u; \]
\[ x_i^v(0) = x_i^{v_0}, \quad y_i^v(0) = y_i^{v_0}, \quad i = 1, ..., N_v; \]

which solves the inviscid Burgers’ equation in a weak sense
\[ \frac{\partial u(x, y, t)}{\partial t} + \frac{\partial u(x, y, t)}{\partial x} u(x, y, t) + \frac{\partial v(x, y, t)}{\partial y} u(x, y, t) = 0, \]
\[ \frac{\partial v(x, y, t)}{\partial t} + \frac{\partial u(x, y, t)}{\partial x} v(x, y, t) + \frac{\partial v(x, y, t)}{\partial y} v(x, y, t) = 0, \]
\[ u(x, y, 0) = u^0(x, y); v(x, y, 0) = v^0(x, y). \]

We have introduced two sets of particles with the coordinates \((x_i^u(t), y_i^u(t)), \ i = 1, \ldots, N_u\) and \((x_i^v(t), y_i^v(t)), \ i = 1, \ldots, N_v\) for the functions \(u(x, y, t)\) and \(v(x, y, t)\), respectively.

The coordinates of the particles \(x_i^u, y_i^u\) and \(x_i^v, y_i^v\) will match each other if the same number of particles is selected for representing \(u\) and \(v\): \(N_u = N_v\). Thus, we will solve the system:
\[ \frac{dx_i(t)}{dt} = u(x_i(t), y_i(t), t), \quad i = 1, ..., N; \]
\[ \frac{dy_i(t)}{dt} = v(x_i(t), y_i(t), t), \quad i = 1, ..., N; \]
\[ x_i(0) = x_i^0, \quad y_i(0) = y_i^0, \quad i = 1, ..., N; \] \hspace{1cm} (7)

In the two-dimensional case, for selecting an interacting neighbor one needs to use the aiming parameter, which is equal to the cosine of the angle between the velocity vector of the particle and the vector connecting the centers of the particles.

For example, \(x_a, y_a\) are the coordinates of the first interacting particle, \(x_b, y_b\) are the coordinates of the second particle, \(v_x, v_y\) are the coordinates of the first particle’s velocity vector. If we use the definition of an inner product, then:
\[ \cos(\theta) = \frac{(x_a - x_b)v_x + (y_a - y_b)v_y}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \sqrt{v_x^2 + v_y^2}}. \]

The particle will interact with its neighbor with the largest aiming parameter.

We will solve the problem for which \(u_0(x, y) = v_0(x, y)\). Let us consider a discontinuous initial condition (“step” function):
\[ u_0(x, y) = \begin{cases} a, & x \leq x_0, y \leq y_0, \\ b, & \text{otherwise}. \end{cases} \] (8)

If \( a > b \), the analytical solution looks like:

\[ u(x, y, t) = \begin{cases} a, & x \leq x_0 + (a + b)t, y \leq y_0 + (a + b)t, \\ b, & \text{otherwise}. \end{cases} \] (9)

We will use \( a = 6, b = 4, x_0 = 3, y_0 = 3 \). Figure 3 presents a comparison of the numerical solution of the Cauchy problem (7) with the initial condition (8) with a time step \( \tau = 0.01 \) obtained by the discontinuous particle method and the analytical solution (9) at the time point (a) \( T = 0.0 \) and (b) \( T = 1.0 \).

![Figure 3](image)

**Figure 3.** Comparison of the numerical solution (colored rectangles) and the analytical solution (red line) of the Cauchy problem for a 2D inviscid Burgers’ equation.

Figure 3 shows that the discontinuous particle method smears the solution by only one particle. Deviation from the exact solution arises only in the upper right angular particle.

5. Conclusion

An algorithm for the discontinuous particle method for the 2D inviscid Burgers’ equation with a new method for selecting interacting particles by the value of the aiming parameter is presented. It is based on another view of the problem like a microscopic one. The solution fits the evolution of the two-dimensional “step”. Our method has a very low viscosity which means that its accuracy at discontinuous solutions is rather high. This was our objective from the beginning of the study. Now
we are able to consider the gas dynamics simulation at the unified computational basis of the particle methods including the stochastic ones.

Acknowledgments
This work was funded by Russian Foundation for Basic Research, project number 20-01-00358.

References
[1] Hockney R W and Eastwood J W 1988 Computer simulation using particles. (New York: Taylor & Francis)
[2] Bogomolov S V and Kuznetsov K V 1998 Matematicheskie modelirovanie 10 93–100 [in Russian]
[3] Harlow F H 1955 A machine calculation method for hydrodynamic problems Tech. Rep. LAMS1956 (Los Alamos Scientific Laboratory)
[4] Harlow F H 1964 Methods in Computational Physics vol 3 ed B Alder, S Fernbach and M Rotenberg (New York: Academic Press) 319–43
[5] Brackbill J and Ruppel H 1986 J. Comput. Phys. 65 314–43
[6] Brackbill J, Kothie D and Ruppel H 1988 Comput. Phys. Commun. 48 25–38
[7] Jiang C, Schroeder C, Selle A, Teran J and Stomakhin A 2015 ACM Trans. Graph. 34 51:1–51:10
[8] Jiang C, Schroeder C and Teran J 2017 J. Comput. Phys. 338 137–64
[9] Fu C, Guo Q, Gast T, Jiang C and Teran J 2017 ACM Trans. Graph. 36 222:1–222:12
[10] Gingold R A and Monaghan J J 1977 Mon. Not. R. Astron. Soc. 181 375–89
[11] Liu G R and Liu M B 2003 Smoothed Particle Hydrodynamics: A Meshfree Particle Method (World Scientific Publishing)
[12] Oñate E, Idelsohn S R, Del Pin F and Aubry R 2004 Int. J. Comput. Methods 1 267–307
[13] Hietel D, Steiner K and Struckmeier J 2000 Math. Models Methods Appl. Sci. 10 1363–82
[14] Teleaga D and Struckmeier J 2008 Int. J. Numer. Methods Fluids 58 945–67
[15] D’yachenko V F 1965 USSR Comput. Math. & Math. Phys. 5 133–43
[16] Yanenko N N, Anuchina N N, Petrenko V E and Shokin Yu I 1973 AIAA Select. Repr. Ser. 15 121–32
[17] Andrianov A N and Efimkin K N 2016 Keldysh Institute Preprints 71 1–16
[18] Belotserkovskii O M and Davydov Yu M 1982 Method of large particles in gas dynamics (Moscow: Nauka) [in Russian]
[19] Sigov Yu S 2001 Computing experiment: the bridge between the past and future of plasma physics (Moscow:FIZMATLIT) [in Russian]
[20] Grigoryev Yu N, Vshivkov V A and Fedoruk M P 2002 Numerical "particle-in-cell" methods: theory and applications. (Utrecht, Boston: VSP)
[21] Bogomolov S V, Zakharov E V and Zerkal S V 2002 Matematicheskie modelirovanie 14 103–16 [in Russian]
[22] Bogomolov S V and Zvenkov D S 2007 Matematicheskie modelirovanie 19 74–86 [in Russian]
[23] Bayev A Zh and Bogomolov S V 2018 Math. Models Comput. Simul. 10 186–97
[24] Bogomolov S V, Esikova N B and Kuvshinnikov A E 2016 Math. Models Comput. Simul. 8 533–47
[25] Bogomolov S V and Kuvshinnikov A E 2019 Math. Models Comput. Simul. 11 768–77
[26] Logan J D 1998 Applied partial differential equations (Springer)