Scientific Visualization, 2023, volume 15, number 4, pages 112 - 123, DOI: 10.26583/sv.15.4.09

Visualization of Calculations of the Discontinuous Particle Method in Problems with Viscosity

S.V. Bogomolov[1,A], A.E. Kuvshinnikov [2,B]

[A] Moscow State University, Faculty of Computational Mathematics and Cybernetics
[B] Keldysh Institute of Applied Mathematics RAS

[1] ORCID: 0000-0002-5414-6019, bogomo@cs.msu.su
[2] ORCID: 0000-0003-1667-6307, kuvsh90@yandex.ru

Abstract

In previous studies, it was shown that the discontinuous particle method performs well in computational hydrodynamics problems with strong gradients, exemplified by the formation of an oblique stress jump. This article explores the application of the discontinuous particle method to problems involving viscosity. The investigation includes a one-dimensional Burgers' equation with an initial condition in the form of a smoothed wave and a two-dimensional Blasius problem. Numerical experiments showed agreement between the obtained solution and the analytical one. However, in the two-dimensional case, the algorithm's performance significantly decreases due to the need to determine particle neighbors. It is concluded that the discontinuous particle method can handle viscosity problems in one dimension, but modifications to the existing algorithm are required for higher-dimensional cases. The study of applying the discontinuous particle method to viscous problems was conducted as part of a comprehensive research effort comparing the relative accuracy of numerical methods on benchmark solutions.

Keywords: Discontinuous particle method, Burgers' equation, computational gas dynamics, Blasius boundary layer.

1. Introduction

In recent years, particle methods have become a valuable tool for numerically solving partial differential equations and have been successfully applied to a wide range of problems in astrophysics, plasma physics, solid-state physics, medical physics, and hydrodynamics [1–4]. In these methods, the solution is sought as a linear combination of δ-functions, where the positions and coefficients represent the locations and weights of particles, respectively. The solution is then obtained by tracking the temporal evolution of particle positions and weights according to the system of ordinary differential equations derived from considering the weak formulation of the problem. To recover pointwise values of the computed solution at some time $t > 0$, regularization of the particle solution is necessary. Therefore, the efficiency of the particle method depends on the quality of regularization procedures, allowing the reconstruction of an approximate solution based on the particle distribution. Typically, regularization of the particle solution involves convolution with a so-called smoothing function, which serves as a smooth approximation of the δ-function and, after appropriate scaling, accounts for the tightness of particle discretization.

Particle methods offer numerous advantages over finite-difference methods. The numerical viscosity introduced by the discretization of convective terms in most finite-difference methods can significantly degrade the accuracy of the computational method, especially when a coarse grid is used. Lagrangian-type methods, on the other hand, can alleviate many issues associated with numerical viscosity since particles provide a non-dissipative approximation of convection. Additionally, in scientific applications like kinetic theory, finite-difference methods may be impractical for realistic scenarios due to problem dimensionality [5]. In contrast, particle schemes concentrate particles in the relevant phase space region, optimizing computer memory usage. Being meshless, particle methods are highly flexible and advantageous for problems with complex geometry and/or moving boundaries.

However, it's essential to consider that the self-adaptation of particle positions to the local flow map comes at the expense of particle distribution regularity. Distances between particles can vary over time, and particles may cluster near discontinuities while excessively spreading apart near non-smooth fronts. This can lead not only to poor resolution of the computed solution but also to extremely low method efficiency. The latter is related to the fact that the time step of the ODE solver used to evolve the particle system in time generally depends on the distance between particles. Therefore, the success of various particle methods relies not only on the accuracy of reconstruction procedures, allowing the recovery of pointwise values of the numerical solution from its particle distribution but also on precise and efficient redistribution algorithms providing adequate resolution in different regions of the computational domain.

The study of applying the discontinuous particle method to viscous problems was conducted as part of a comprehensive research effort comparing the relative accuracy of numerical methods on benchmark solutions. This research was previously carried out for inviscid gas dynamics problems with benchmark solutions presented in [6–8]. Currently, research is ongoing to comparatively assess the accuracy of numerical methods for viscous problems using benchmark solutions. Particularly interesting is the application of the discontinuous particle method, given its fundamentally different nature compared to common numerical methods.

2. Basic equations of the discontinuous particle method

Let there be N material points, located at the starting moment in coordinates x_i^0 and moving with the speed $v_i(x,t)$ $(i = 1, \dots, N)$. Such verbal formulation corresponds to the initial value problem (Cauchy problem).

$$\begin{cases} \dfrac{dx_i(t)}{dt} = v(x_i(t), t), \\ x_i(0) = x_i^0, \ i = 1, \dots, N. \end{cases} \tag{1}$$

In articles [9, 10], the transition from equation (1) to the transport equation in a differential form is shown:

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} + \dfrac{\partial v(x,t)u(x,t)}{\partial x} = 0, \\ u(x,0) = u_0(x). \end{cases} \tag{2}$$

As to say, if the coordinates of points change according to the system of equations (1), then the density $u(x,t)$ is a generalized solution to the Cauchy problem for the transport equation (2).

Let's describe a modification of the discontinuous method with a different density correction approach. The particles chosen for correction will be called interacting, and the correction process will be called interaction. Introduce a uniform time grid with a step. Consider the system as a set of N macroparticles. To describe the particles, we introduce the following notations: x_i^k — the coordinate of the center of the i-th particle at the k-th moment in time, v_i^k — the velocity of the particle, H_i^k — the height (density) of the particle.

Also, each particle has a time-invariant mass, indicating the conservativeness of the method. The new algorithm is based on the conservation of mass between particles. The mass between particle coordinates is equal to half the sum of the masses of the particles and, in the absence of diffusion, should also remain constant. Let S_i denote the mass between the (i-1)-th particle and the i particle. We calculate the mass S_i as the area of trapezoids (Figure 1):

$$S_i = \frac{H_i + H_{i-1}}{2}(x_i - x_{i-1}).$$

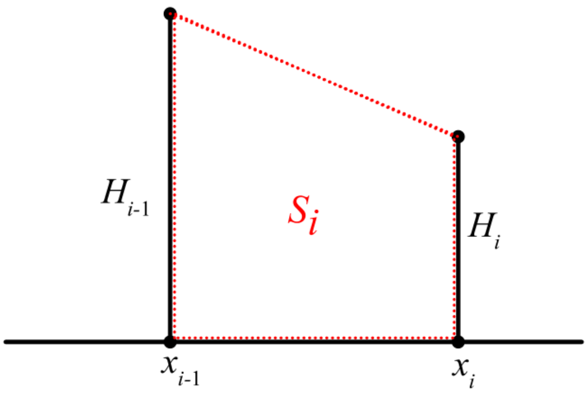

Figure 1 – Particles forming a trapezoid

With the absence of mass diffusion, the mass between particles remains constant over time. We'll record the values of S_i^0 at the initial time moment $t = 0$. Let's proceed with the initialization procedure of particle parameters at the initial time. Let's suppose we have an initial density $u_0(x)$. The coordinates of particles x_i^0 can be uniformly distributed over the computational domain, where $i = 1, \dots, N$.

$$H_i^0 = u_0(x_i^0),\;\; i = 1, \dots, N;$$

$$S_i^0 = \frac{1}{2}(H_{i-1}^0 + H_i^0)(x_i^0 - x_{i-1}^0),\;\; i = 2, \dots, N.$$

Coordinates of particles when solving the Hopf equation must be correct for the system of equations [11]:

$$\begin{cases} \dfrac{dx_i(t)}{dt} = \dfrac{1}{2}H_i, \;\; i = 1, \dots, N; \\ x_i(0) = x_i^0. \end{cases}$$

Let's not forget that the particle method algorithm is constructed as a predictor-corrector. First, we solve the system of ordinary differential equations using the explicit Euler method:

$$x_i^{k+1} = x_i^k + \frac{1}{2}\tau H_i^k.$$

After the particle shift, the distance between them changes, leading to changes in the trapezoid areas. Therefore, at the correction stage, it is necessary to adjust the heights of the particles so that the mass between the particles remains constant. Let's consider possible cases of particle interactions (Figure 2).

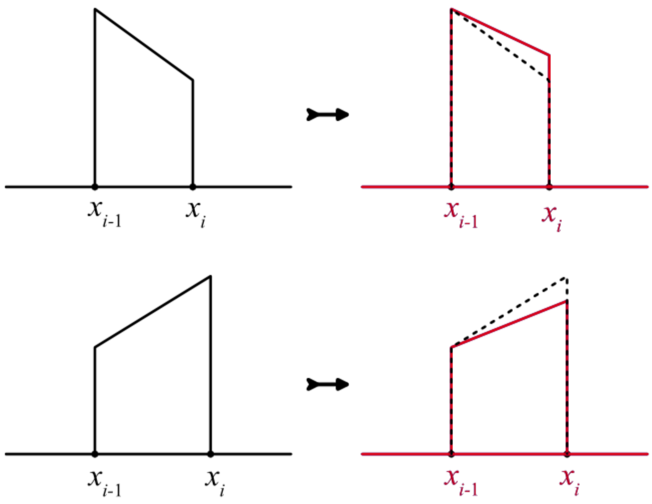

Figure 2 – Examples of particle interactions

A particle with a higher density collides with a particle with a lower density, resulting in a reduction of the trapezoid area between the particles. In this case, to maintain the trapezoid area, we will increase the height of the particle with lower density.

A particle with higher density moves away from a particle with lower density, leading to an increase in the trapezoid area between the particles. In this case, to maintain the trapezoid area, we will decrease the height of the particle with higher density.

By applying these rules for particle rearrangement and selection criteria, the interaction result with one of the neighbors may change the trapezoid area with another neighbor, for which correction has already been made, indicating the algorithmic error. Interactions between particles arising in this way are not considered.

The corrector adjusts the height of the i-th ($i = 2, \ldots, N$)) particle in such a way that the trapezoid area between the particles remains constant:

$$\tfrac{1}{2}\left(H_i^{k+1} + H_{i-1}^k\right)\left(x_i^{k+1} - x_{i-1}^{k+1}\right) = S_i^0. \tag{3}$$

Consequently, the height of the i particle on the new moment of time $(k+1)$ is calculated this way:

$$H_i^{k+1} = \frac{2S_i^0}{x_i^{k+1} - x_{i-1}^{k+1}} - H_{i-1}^k.$$

3. Diffusion for the one-dimensional particle method

Let’s analyze Burgers’ equation:

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} = \mu \frac{\partial^2 u(x,t)}{\partial x^2},$$

where μ — diffusion coefficient. This equation describes quasi-linear advection with diffusion. In the initial stage, we solve the advection equation without considering diffusion:

$$x_i^{k+1} = x_i^k + \tfrac{1}{2}\tau H_i^k, \ \ i = 1, \ldots, N.$$

$$\widetilde{H}_i^k = \frac{2S_i^0}{x_i^{k+1} - x_{i-1}^{k+1}} - H_{i-1}^k.$$

where S_i^k — mass, located between the (i-1)-th and i particles on the k temporal layer, $\widetilde{H}_i^k$ is the intermediate value of the particle height. The particle coordinates form a non-uniform grid. Let's introduce the following notations:

$$h_i = x_i^{k+1} - x_i^k,$$

$$\tilde{h}_i = \frac{h_{i+1} + h_i}{2},$$

Let's write out the finite difference approximation of the second derivative on a non-uniform grid and obtain the value of H_i^{k+1} on the new temporal layer:

$$H_i^{k+1} = \widetilde{H}_i^k + \tau \frac{\mu}{\tilde{h}_i}\left(\frac{H_{i+1}^k - H_i^k}{h_{i+1}} - \frac{H_i^k - H_{i-1}^k}{h_i}\right). \tag{4}$$

Also, due to diffusion, the mass between particles will change, so it's necessary to find the new trapezoid areas. Let's consider the i trapezoid between the (i-1)-th and i-th particles. We'll calculate the flux density at the trapezoid boundaries according to Fick's law:

$$j = -\mu \frac{\partial u}{\partial x}.$$

The flux density at the right boundary of the i-th trapezoid is determined by the expression:

$$j_2 = -\mu \frac{H_{i+1}^k - H_{i-1}^k}{x_{i+1}^k - x_{i-1}^k}.$$

In the same way, for the flux density at the left boundary of the i trapezoid, what we have is:

$$j_1 = -\mu \frac{H_i^k - H_{i-2}^k}{x_i^k - x_{i-2}^k}.$$

The mass that has flowed from the τ of the $(i+1)$ trapezoid to the i is equal to $-\tau j_2$ over the time step. The mass that has flowed from the $(i-1)$ to the i is equal to τj_1. The new mass of the trapezoid is now:

$$S_i^{k+1} = S_i^k - \tau(j_2 - j_1) = S_i^k + \tau\mu\left(\frac{H_{i+1}^k - H_{i-1}^k}{x_{i+1}^k - x_{i-1}^k} - \frac{H_i^k - H_{i-2}^k}{x_i^k - x_{i-2}^k}\right). \quad (5)$$

Among Burgers' equation solutions, there is a smoothed wave:

$$u(x,t) = \frac{a + b\exp\big(\lambda(x - x_0 - Dt)\big)}{1 + \exp\big(\lambda(x - x_0 - Dt)\big)},$$

where $D = \frac{1}{2}(a+b)$, $\lambda = \frac{1}{2}(a-b)/\mu$. Let it be so that $a = 1$, $b = 0.2$, $x_0 = 5$, $\mu = 0.15$. Let's take $u_0 = u(x,0)$ as the initial condition The comparison between the numerical and analytical solutions is depicted in Figure 3. The animation starts at time $T = 0$ and ends at $T = 10.0$.

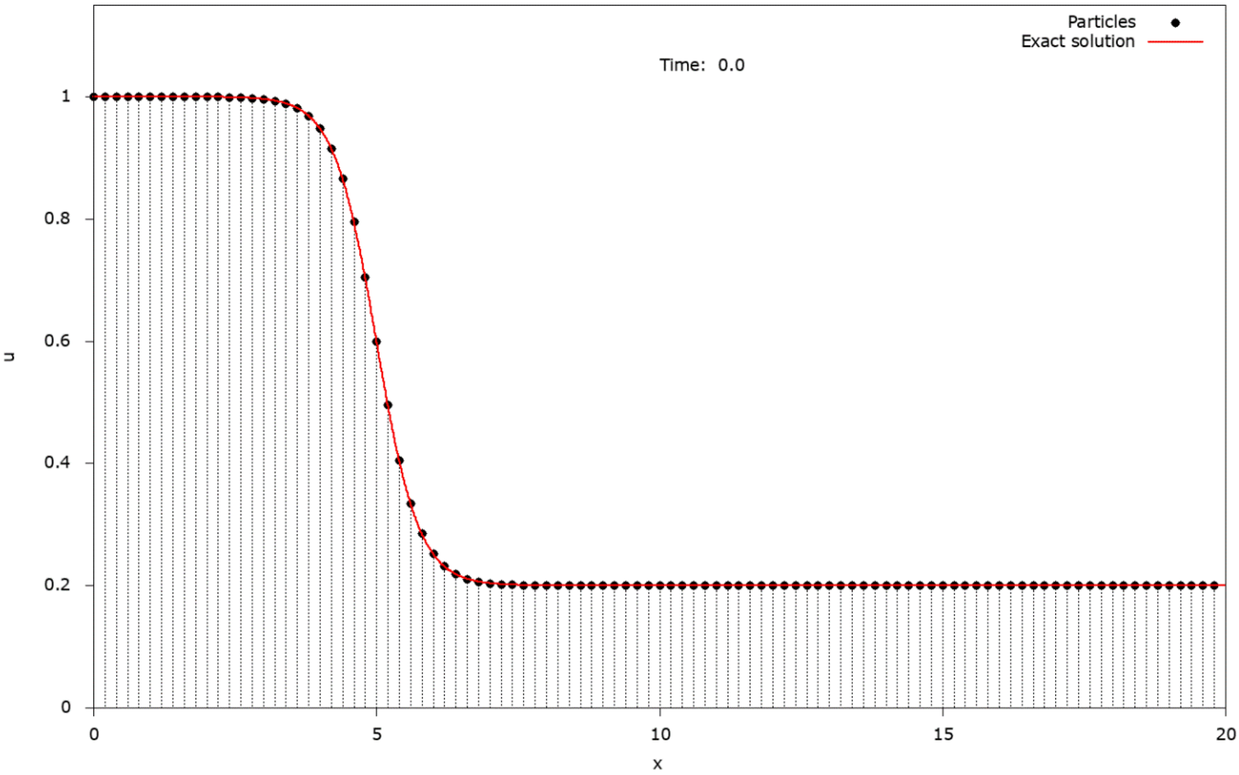

Figure 3 – Applying particle method to Burgers' equation

The black dots are the centers of the particles, the red line is the exact solution. For better visual perception of the particle density, vertical lines are drawn from the particle centers. It can be seen that the particle method allows solving such problems with a given accuracy. It is also worth noting the densification of the particles, due to which the time step is reduced. In problems with strong gradient densification helps to better calculate regions with strong gradient, however, in the Burgers' equation such a high density of particles reduces the space step too much. Thus, the authors see the necessity of using the "birth-death" of particles scheme previously applied in [11].

4. Increasing the particle method accuracy with the problem for two-dimensional linear transport equation as an example

Let us consider the following microscopic Cauchy problem for a 2D case:

$$\begin{cases} \dfrac{dx_i^u(t)}{dt} = u(x_i^u(t), y_i^u(t), t), & i = 1, \dots, N_u; \\ \dfrac{dy_i^u(t)}{dt} = v(x_i^u(t), y_i^u(t), t), & i = 1, \dots, N_u; \\ \dfrac{dx_i^v(t)}{dt} = u(x_i^v(t), y_i^v(t), t), & i = 1, \dots, N_v; \\ \dfrac{dx_i^v(t)}{dt} = v(x_i^v(t), y_i^v(t), t), & i = 1, \dots, N_v; \\ x_i^u(0) = x_i^{u0},\ y_i^u(0) = y_i^{u0}, i = 1, \dots, N_u; \\ x_i^v(0) = x_i^{v0},\ y_i^v(0) = y_i^{v0}, i = 1, \dots, N_v. \end{cases}$$

which solves the inviscid Burgers' equation in a weak sense.

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} + \dfrac{\partial u(x,y,t)u(x,y,t)}{\partial x} + \dfrac{\partial v(x,y,t)u(x,y,t)}{\partial y} = 0 \\ \dfrac{\partial v(x,y,t)}{\partial t} + \dfrac{\partial u(x,y,t)v(x,y,t)}{\partial x} + \dfrac{\partial v(x,y,t)v(x,y,t)}{\partial y} = 0 \\ u(x,y,0) = u^0(x,t), \qquad v(x,y,0) = v^0(x,t). \end{cases}$$

We have introduced two sets of particles with the coordinates $(x_i^u(t), y_i^u(t))$, $i = 1, \dots, N_u$ and $(x_i^v(t), y_i^v(t))$, $i = 1, \dots, N_v$ 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 of u and v: $N_u = N_v$. Thus, we will solve the system:

$$\begin{cases} \dfrac{dx_i(t)}{dt} = u(x_i(t), y_i(t), t), & i = 1, \dots, N; \\ \dfrac{dy_i(t)}{dt} = v(x_i(t), y_i(t), t), & i = 1, \dots, N; \\ x_i(0) = x_i^0,\ y_i(0) = y_i^0, & i = 1, \dots, N. \end{cases}$$

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:

$$\cos(\theta) = \frac{(x_a - x_b)v_x + (y_a - y_b)v_y}{\sqrt{(x_a - x_b)^2 + (x_a - x_b)^2}\sqrt{v_x^2 + v_y^2}}$$

Let's analyze the test problem for the two-dimensional linear advection equation [12]:

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

$$\psi(x,y,0) = \psi_0(x,y).$$

where $u(x,y) = -(y - y_0)/25$, $v(x,y) = -(x - x_0)/25$. With such a velocity problem, the initial profile ψ_0 should rotate around the point (x_0, y_0) without changing its shape. Let's compare two numerical solutions obtained by the particle method we described, in the predictor stage, we use the first-order Euler approximation and the modified second-order Euler method.

Let's use the modified second-order Euler method. First, take a time step of $\frac{1}{2}\tau$.

$$x_i^{k+1/2} = x_i^k + \tfrac{1}{2}\tau u\left(x_i^k, y_i^k\right),$$
$$y_i^{k+1/2} = y_i^k + \tfrac{1}{2}\tau v\left(x_i^k, y_i^k\right).$$

In the second step, we obtain the coordinates of the particle center x_i^{k+1}, y_i^{k+1} at the new time step using the values $x_i^{k+1/2}$, $y_i^{k+1/2}$:

$$x_i^{k+1} = x_i^k + \tau u\left(x_i^{k+1/2}, y_i^{k+1/2}\right),$$
$$y_i^{k+1} = y_i^k + \tau v\left(x_i^{k+1/2}, y_i^{k+1/2}\right).$$

Let's take the initial function $\psi_0(x,y)$ in the form of a right circular cone with a radius of 5 and a height of 1, whose center is located at the point with coordinates $(25,10)$:

$$\psi_0(x,y) =$$
$$= \begin{cases} -\dfrac{(x-25)^2}{25} - \dfrac{(y-10)^2}{25} + 1, & (x-25)^2 + (y-10)^2 \le 25; \\ 0, & otherwise. \end{cases}$$

Figure 4 shows the initial condition. Figures 5 and 6 show solutions by two methods after three circles. On the left is a three-dimensional image, and on the right is a top-down view, more suitable for visual analytics. It is evident that the solution obtained with the modified method better preserves the shape of the initial profile.

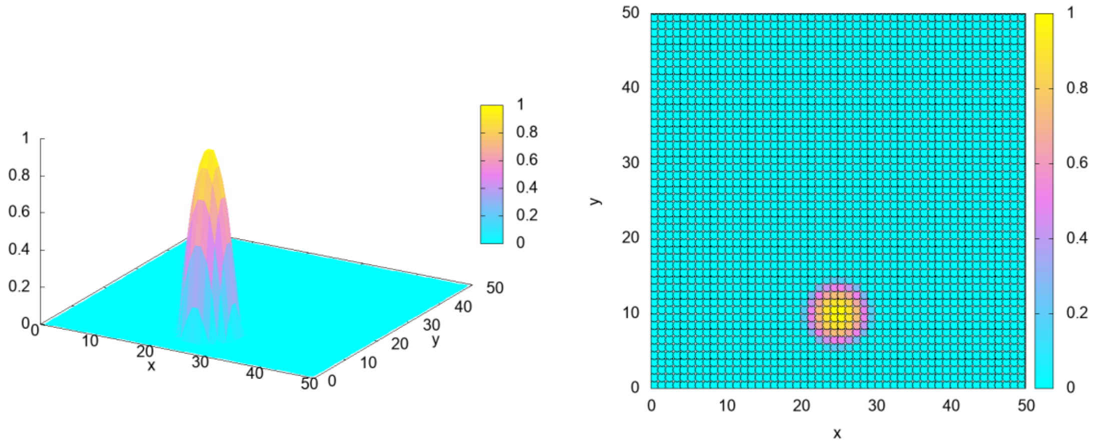

Figure 4 – Initial conditions

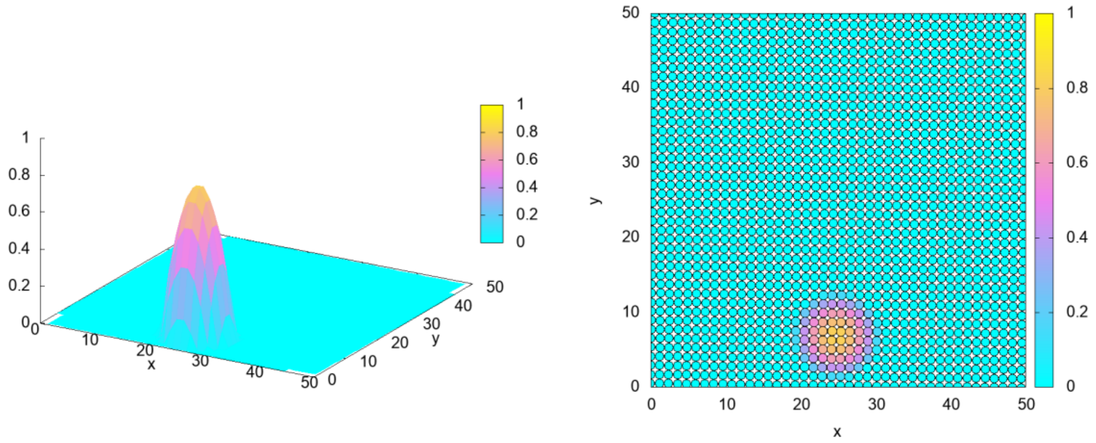

Figure 5 – Numerical solution with particle method using Euler method

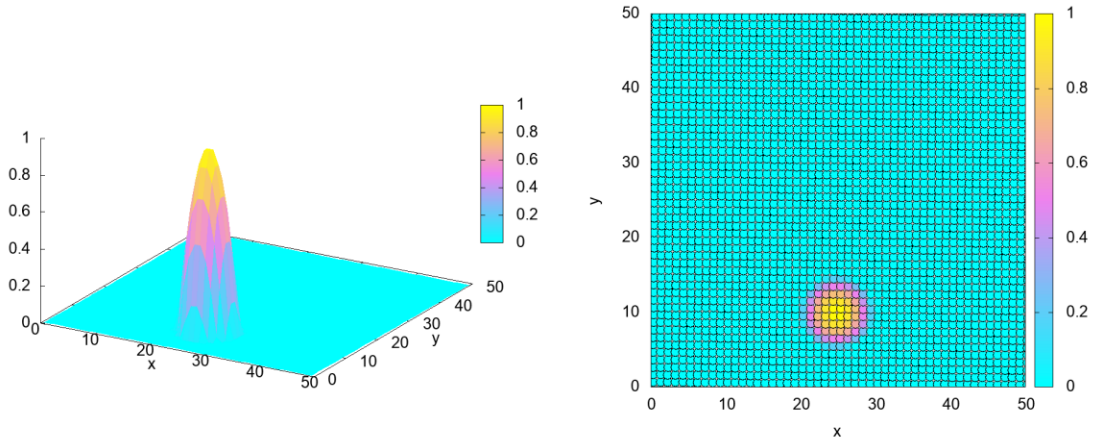

Figure 6 – Numerical solution with particle method using modified Euler method

5. Particle method for viscous gas

The equations of gas dynamics express the general laws of mass, momentum, and energy. Following [13, 14], let's write the system of equations for the two-dimensional case in Eulerian variables:

$$\begin{cases} \dfrac{\partial \rho}{\partial t}+\dfrac{\partial u\rho}{\partial x}+\dfrac{\partial v\rho}{\partial y}=0 \\ \dfrac{\partial \rho u}{\partial t}+\dfrac{\partial (u\rho u)}{\partial x}+\dfrac{\partial (v\rho u)}{\partial y}=-\dfrac{\partial p}{\partial x}+2\mu\dfrac{\partial^2 u}{\partial x^2}+\mu\dfrac{\partial^2 v}{\partial x\partial y}+\mu\dfrac{\partial^2 u}{\partial y^2} \\ \dfrac{\partial \rho v}{\partial t}+\dfrac{\partial (u\rho v)}{\partial x}+\dfrac{\partial (v\rho v)}{\partial y}=-\dfrac{\partial p}{\partial y}+\mu\dfrac{\partial^2 v}{\partial x^2}+\mu\dfrac{\partial^2 u}{\partial x\partial y}+2\mu\dfrac{\partial^2 v}{\partial y^2} \\ \dfrac{\partial E}{\partial t}+\dfrac{\partial uE}{\partial x}+\dfrac{\partial vE}{\partial y}=-\dfrac{\partial pu}{\partial x}-\dfrac{\partial pv}{\partial y}+\dfrac{\partial (u\tau_{xx})}{\partial x}+\dfrac{\partial \left(u\tau_{yx}\right)}{\partial y}+\dfrac{\partial \left(v\tau_{xy}\right)}{\partial x}++\dfrac{\partial \left(v\tau_{yy}\right)}{\partial y} \\ p=(\gamma-1)\left(E-\dfrac{\rho}{2}(u^2+v^2)\right) \end{cases}$$

Ideal gas, $\gamma = 1.4$. $\rho, u, v,\ p, E, \tau$ — density, x and y components of velocity, pressure, total energy, and stress tensor.

The algorithm for solving the two-dimensional problem is similar to the one-dimensional case. The computational domain is divided into a finite number of regions serving as the bases of particles. Particle heights are determined from the initial condition – functions$\rho(x, y, 0)$, $\rho u(x, y, 0)$, $\rho v(x, y, 0)$, $E(x, y, 0)$ computed at the particle base centers (points $x_i(0), y_i(0)$). The radius of particle bases of all types is assumed to be the same.

First of all, just like in the one-dimensional case, we solve systems of ordinary differential equations for the coordinates of the four types of particles using the Euler method. In the two-dimensional case, a partner for interaction is chosen by minimizing the "aiming" parameter—the angle between the relative velocity vector and the vector connecting the centers of the particles (algorithmically, maximizing the cosine of this angle). Once we've selected the j-th particle for interaction, we move on to a one-dimensional problem. Then, using a corrector, we adjust the height of the i particle similarly to (3) to keep the area of the trapezoid between the particles constant.

$$\tfrac{1}{2}\left(H_i^{k+1}+H_j^k\right)\sqrt{\left(x_i^{k+1}-x_j^{k+1}\right)^2+\left(y_i^{k+1}-y_j^{k+1}\right)^2}=S_{ij}^0.$$

From here, we can find the preliminary height (without considering pressure forces yet):

$$H_i^{k+1}=\frac{2S_{ij}^0}{\sqrt{\left(x_i^{k+1}-x_j^{k+1}\right)^2+\left(y_i^{k+1}-y_j^{k+1}\right)^2}}-H_j^k.$$

The next step in the algorithm involves accounting for pressure forces. The pressure difference on the left and right sides of a particle leads to changes in its momentum and energy, increasing the volume of the respective particles. Similarly to [8], we arrive at the following computational formulas:

$$V_{\rho u_i}\left(t_{j+1}\right) = V_{\rho u_i}\left(t_j\right) + \tau\left(p_{1i}^{-}\left(t_j\right) - p_{1i}^{+}\left(t_j\right)\right)$$

$$V_{\rho v_i}\left(t_{j+1}\right) = V_{\rho v_i}\left(t_j\right) + \tau\left(p_{2i}^{-}\left(t_j\right) - p_{2i}^{+}\left(t_j\right)\right)$$

$$V_{E_i}\left(t_{j+1}\right) = V_{E_i}\left(t_j\right) + \tau\left(p_{1i}^{-}\left(t_j\right)u_{1i}^{-}\left(t_j\right) - p_{1i}^{+}\left(t_j\right)u_{1i}^{+}\left(t_j\right)\right) +$$
$$+\tau\left(p_{2i}^{-}\left(t_j\right)v_{2i}^{-}\left(t_j\right) - p_{2i}^{+}\left(t_j\right)v_{2i}^{+}\left(t_j\right)\right)$$

The computed values of density, momentum, and energy from the previous step allow for determining the pressure at the particle's center. To achieve this, the equation of state is applied.

To determine the pressure and velocity values at the particle boundary, a pressure calculation scheme based on particle "interaction" is employed to determine the pressure and velocity values at the particle boundary. If an interaction occurred at the particle boundary during the time step (according to the criteria described above for the one-dimensional configuration), then the pressure and velocity at that boundary are set equal to the pressure and velocity of the particle that caused the rearrangement. If no interaction occurs, the pressure at the boundary is set equal to the pressure at the particle's center. Consequently, the volumes of particles ρu, ρv, and E are additionally increased.

Next, we need to consider the diffusive part. It is needed to take a particle i and select all neighboring particles j_i within a certain interaction radius R. For each pair (i, j_i), simplify the problem to make it one-dimensional and use formulas (4) and (5). It's worth stating that the current algorithm requires finding neighbors at each time step, significantly impacting the method's performance and eliminating one of the advantages compared to traditional numerical methods where grid nodes are stationary, and neighbors are always known.

As a test case to develop the capabilities of the particle method, the problem of a compressible gas flow around a plate is considered. The boundary conditions on the plate are standard no-slip conditions (u=v=0). A significant difference from the Blasius problem is that a singularity forms at the plate's nose. For the boundary condition, special particles with zero velocity are used; their height does not change.

For the laminar boundary layer, the layer thickness δ_{incom} for the incompressible case is determined by the relation [15]:

$$\delta_{\text{incom}} = 5.0\frac{x}{\sqrt{\text{Re}}},$$

With this, it is possible to calculate the thickness of the boundary layer for compressible gas δ_{com}:

$$\delta_{\text{com}} = \delta_{\text{incom}}\left(1 + 0.72r\frac{\gamma - 1}{2}M_w^2\right)^{0.85}.$$

Figure 7 shows the balanced velocity field. The Mach number $M = 300$, free stream temperature $T_\infty = 300$, free stream density $\rho_\infty = 1.17641$, thickness for compressible gas $\delta_{\text{com}} = 0.00121572$.

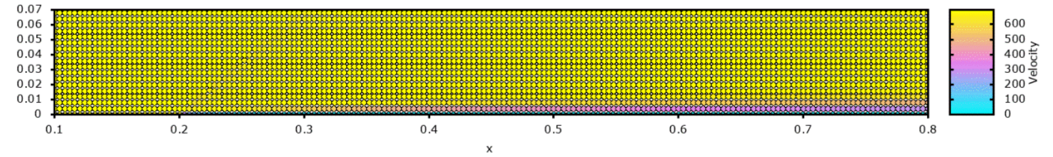

Figure 7 – Velocity distribution in the Blasius problem

As a result of the establishment, a qualitative flow pattern has been obtained. However, there are visible white spots in the figure, indicating the absence of particles at the spout. This result can be explained by the fact that at the nose of the plate the particles experience sharp braking and fly up, thereby forming a narrow strip without particles. This can also be eliminated by the mechanism of "birth-death" of particles.

6. Conclusion

With the use of the discontinuous particle method without shape consideration two diffusion problems were solved: the one-dimensional problem for Burgers' equation and Blasius' two-dimensional problem. It was also discovered that the use of a high-precision circuit at the corrector stage increases the final accuracy of the particle method. However, in problems with viscosity, regions arise with an increased number of particles, which is why the time step has to be reduced, and regions in which there are no particles, that is, the numerical result is not defined. It is proposed to solve such problems by the mechanism of "birth-death" of particles, which will be done in subsequent works. The developed approach ensures the participation of the discontinuous particle method in the general comparative analysis of the accuracy of numerical methods on reference solutions for viscous problems.

7. Acknowledgments

Calculations were performed on the hybrid supercomputer K-100 installed in the Supercomputer Centre of Collective Usage of KIAM RAS.

8. References

1. Harlow F.H. The Particle-in-Cell Computing Method for Fluid Dynamics // Methods in Computational Physics. Vol. 3, ed. B Alder, S Fernbach, M Rotenberg. New York: Academic Press. 1964. pp. 319–343.
2. Hockney R.W., Eastwood J.W. Computer simulation using particles, New York: Taylor&Francis, 1988. doi:10.1201/9780367806934
3. Choquin J.P., Huberson S. Particle simulation of viscous flow // Comput. & Fluids. 1989 Vol. 17 no.2. pp. 397–410. doi:10.1016/0045-7930(89)90049-2
4. Hermeline F. A deterministic particle method for transport diffusion equations: Application to the Fokker-Planck equation // J. Comput. Phys. 1989. Vol. 82. no.1. pp. 122–146. doi:10.1016/0021-9991(89)90038-7
5. Dimarco G., Pareschi L. Numerical methods for kinetic equations // Acta Numerica. 2014. Vol. 23. pp. 369–520. doi:10.1017/S0962492914000063
6. Alekseev A.K., Bondarev A.E., Kuvshinnikov A.E. Comparative analysis of the accuracy of OpenFOAM solvers for the oblique shock wave problem // Mathematica Montisnigri. 2019. Vol. XLV. pp. 95–105. doi:10.20948/mathmontis-2019-45-8
7. Alekseev A.K., Bondarev A.E., Kuvshinnikov A.E. On uncertainty quantification via the ensemble of independent numerical solutions // Journal of Computational Science 2020. Vol. 42. 101114. doi:10.1016/j.jocs.2020.101114
8. Bondarev A.E, Kuvshinnikov A.E. Analysis and Visualization of the Computational Experiments Results on the Comparative Assessment of OpenFOAM Solvers Accuracy for a Rarefaction Wave Problem // Scientific Visualization. 2021. Vol. 13. no. 3. pp. 34–46. doi:10.26583/sv.13.3.04
9. Bogomolov S.V., Kuvshinnikov A.E. Discontinuous particles method on gas dynamic examples // Math. Models Comput. Simul. 2019 Vol.11 no.5. pp. 768–777. doi:10.1134/S2070048219050053
10. Bogomolov S.V, Bondarev A.E., Kuvshinnikov A.E. Comparative Verification of Numerical Methods Involving the Discontinuous Shapeless Particle Method// Scientific Visualization. 2022. Vol. 14. no.4. pp. 97–109. doi:10.26583/sv.14.4.09

11. Bogomolov S.V., Zvenkov D.S. Explicit particle method, non-smoothing gas-dynamic discontinuities. Matem. Mod. 2007. Vol. 19. no.3. pp. 74–86. (in Russian)

12. Crowley W.P. Numerical advection experiments // Monthly Weather Rev. 1968. Vol. 96. no.1. pp. 1–11. doi:10.1175/1520-0493(1968)096<0001:NAE>2.0.CO;2

13. Rozdestvenskii B.L., Janenko N.N. Systems of Quasilinear Equations and Their Applications to Gas Dynamics, American Mathematical Society, 1983. doi:10.1090/mmono/055

14. Landau L.D., Lifshitz E.M. Fluid Mechanics, Pergamon Press, Oxford, 1987. doi:10.1016/C2013-0-03799-1

15. Schlichting H., Gersten K. Boundary layer theory, 9th ed. Berlin Heidelberg, Springer-Verlag, 2016. doi: 10.1007/978-3-662-52919-5