Supercomputer model of dynamical dusty gas with intense momentum transfer between phases based on OpenFPM library

Vitaliy Grigoryev\textsuperscript{1,2}, Olga Stoyanovskaya\textsuperscript{1} and Nikolay Snytnikov\textsuperscript{3}

\textsuperscript{1}Lavrentiev Institute of Hydrodynamics SB RAS, Ac Lavrentieva ave., 15, Novosibirsk, 630090, Russia
\textsuperscript{2}Crimean astrophysical observatory RAS, p. Nauchny, Bakhchisaray, Crimea, 298409, Russia
\textsuperscript{3}Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Ac Lavrentieva, 6, Novosibirsk, 630090, Russia

E-mail: vitaliygrigoryev@yandex.ru

Abstract. The paper considers the solution of model gas-dynamic problems (propagation of plane sound wave, one-dimensional shock tube problem, three-dimensional problem of a point explosion in a continuous medium) in the case of a gas-dust medium. The interaction of dust and gas was taken into account using the IDIC method within the SPH method used to solve gas-dynamic equations. An important feature of the work is the use of the open computational package OpenFPM, which makes it easy to carry out parallel computations. The main advantage of this package is the ready-made (implemented by the authors of the package) and intuitive, automatically parallelizable vector data structures, the use of which is identical both in the case of calculations on a personal computer and in the case of using supercomputer resources. The paper analyzes the efficiency of parallelization of numerical solutions of the considered problems.

1. Introduction

Two-phase dispersed media occur in technical applications and natural phenomena. We will say that a medium is a medium with intense interaction between phases if the characteristic time of mutual exchange of momentum or energy between the carrier and dispersed phases (relaxation time) is much less than the characteristic time of the process under consideration (dynamic time). Such tasks are multiscale in time or stiff. Examples of media with intense interphase interaction are mixtures of gas with a fine catalyst in chemical reactors of natural gas processing, flows in devices for thermal spraying of matter on the surface, gas-dust circumstellar disks from which planets are formed, and others. Numerical simulation of the dynamics of media with interphase interaction by explicit methods requires the use of a time step that is less than the shortest relaxation time. For media with intense interphase interaction, this requirement turns out to be much more demanding than Courant’s condition for explicit methods. To reduce computational costs, specially designed asymptotic-preserving or AP-methods are usually used for stiff problems (see e.g. [1]). AP-methods allow using a time step that does not depend on the minimum relaxation time. Moreover, the number of arithmetic operations for each step of the AP-method practically coincides with the computational costs per step of the explicit method.
For Euler methods (for example, finite differences and finite volumes methods), AP-approaches to integrating equations with stiff relaxation terms are well known [1]. However, for Lagrangian methods, in which the carrier and dispersed phases are represented by separate sets of particles, the development and verification of such approaches has not yet been completed. The use of methods that preserve asymptotics is especially important for problems in which the dispersed phase cannot be characterized by one size (monodisperse approximation), but is polydisperse, since the amount of calculations increases in proportion to the number of dust fractions. Moreover, three-dimensional simulation of the dynamics of gas and polydisperse solid phase requires the use of high-performance computing even when using AP-methods.

The presented work is aimed at developing a supercomputer model of the dynamics of a polydisperse gas suspension with intense interphase interaction for modeling gas chemistry processes. The mathematical model is based on the approach of interpenetrating continua and is a system of equations for a continuous medium written for gas and each fraction of dispersed particles. Here we report the results of a particular case of such a medium in which dust particles are monodisperse and have a negligible volume. The numerical model is based on the new AP-method SPH-IDIC, which uses the Lagrangian approach to calculating the pressure gradient (Smoothed Particle Hydrodynamics, SPH) and the Euler mesh to calculate the interphase interaction (Implicit Drag in Cell, IDIC). The effectiveness of the numerical SPH-IDIC method for the integration of stiff equations of motion was studied earlier in the works [2] (monodisperse model) and [3] (polydisperse model) on one-dimensional problems. The first results of dynamical three-dimensional simulations of dusty gas by the SPH-IDIC method using CUDA technology for graphics accelerators are briefly described in [4]. In this article, we present preliminary results on the parallel implementation of SPH-IDIC method for supercomputers with distributed memory and three-dimensional simulation of dusty gas.

To develop such implementation we used free library OpenFPM [5], which allows automatic parallelization of particle-particle and particle-mesh methods. Our implementation of SPH-IDIC model of dynamical dusty gas was verified using analytical solutions and alternative numerical scheme results.

In Section 2, we describe the governing equations of the gas-dust mixture (Subsection 2.1) and numerical methods for solving these equations (Subsection 2.2), and also instruments of OpenFPM library used for parallelization (Subsection 2.3). In Section 3, we demonstrate the verification and simulation results: subsection 3.1 is reserved for the problem of sound wave propagation, subsection 3.2 — for the shock-tube problem and subsection 3.3 — for the Sedov blast wave problem. In Section 4, we summarize the results of our research.

2. Model of a gas-dust medium and its implementation

2.1. Governing equations

In this work, the following equations describing the dynamics of gas-dust flows will be solved. Let $\varepsilon$ be the ratio of the mass fractions of dust and gas in a unit volume. Then for a gaseous medium with a density $\rho_g$ and a dust component with a density $\rho_d = \varepsilon \rho_g$ one can write down the standard equations of continuity:

$$\frac{\partial \rho_g}{\partial t} + (\vec{v} \cdot \nabla \rho_g) = -\rho_g \nabla \vec{v},$$

$$\frac{\partial \rho_d}{\partial t} + (\vec{u} \cdot \nabla \rho_d) = -\rho_d \nabla \vec{u}.\tag{2}$$

Here $\vec{v}$ and $\vec{u}$ — are the velocities of the gas and dust media, respectively. Let us underline that one should distinguish between the density of the dust component $\rho_d$ in the equations (also called concentration or number density), and the density of the substance of which the dust grains directly consist (material density).
The equations of motion, taking into account the relaxation terms responsible for the mutual exchange of momentum due to drag, can be written as follows:

\[
\rho_g \left( \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right) = -\nabla p + \frac{\rho_d (\vec{u} - \vec{v})}{t_{\text{stop}}},
\]

\[
\rho_d \left( \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right) = -\frac{\rho_d (\vec{u} - \vec{v})}{t_{\text{stop}}},
\]

where \( p \) is the gas pressure. An important parameter here is \( t_{\text{stop}} \) — the relaxation time of dust particles relative to the gas, which can be extremally small for fine particles.

We write the energy conservation law in the form

\[
\rho_g \left( \frac{\partial \epsilon}{\partial t} + \vec{v} \nabla \epsilon \right) = -p \nabla \vec{v} + \frac{\rho_d (\vec{u} - \vec{v})^2}{t_{\text{stop}}},
\]

Here the specific internal energy of the gas \( \epsilon \) appeared, which is also included in the equation of state:

\[ p = (\gamma - 1) \rho g \epsilon, \]

where \( \gamma = 1.4 \) — adiabatic parameter.

2.2. Numerical methods. SPH-IDIC and Lax–Wendroff schemes

The Smoothed-Particle Hydrodynamics method was proposed by the authors [6, 7] and is actively used in hydro- and gas-dynamic simulations, primarily when describing complex flow geometry.

For calculations, the continuity equations (1) - (2) were replaced by interpolation over SPH particles of mass \( m \). The equations of motion and energy (3) - (5) were approximated within the SPH ideology by the kernel \( M_4 \) (cubic spline) at the smoothing length \( h \):

\[
W(\vec{r}, h) = \alpha \begin{cases} 
1 - \frac{3}{2} q^2 + \frac{3}{4} q^3, & 0 \leq q \leq 1, \\
\frac{1}{4} (2 - q)^3, & 1 < q \leq 2, \\
0, & 2 < q,
\end{cases} \quad q = \frac{|\vec{r}|}{h}.
\]

Here \( \vec{r} = (x, y, z) \) is the radius vector connecting the considered particle and its neighbor, \( \alpha \) is the normalization parameter equal to \( 2/(3h) \) in the case of one-dimensional calculations and \( 1/(\pi h^3) \) in the case of three-dimensional calculations. We used the approximation that guarantees global conservation of mass and momentum with machine precision [8]. A detailed description of all calculation formulas is available in [2, 3, 4].

Consideration of drag terms in the equations of motion within the SPH can be significantly different: the classical approach of Monaghan and Kocharian [9], the use of a two-humped kernel [10], but we have implemented the IDIC method [2, 3]. The essence of the method consists in dividing the entire computational domain into cells of arbitrary shape with the linear size close to the SPH smoothing radius. The drag terms in the equations of motion are calculated based on the cell-averaged values of macroparameters. Then, based on the found drag terms, the new velocities of gas and dust are implicitly found at the next time step, taking into account the smoothing kernel and its gradient. For problems with intense momentum transference the IDIC method has advantages due to its AP-nature.

To solve the same equations (1) - (5) in the one-dimensional case, Lax–Wendroff scheme as an alternative approach was used. To implement this scheme all the equations were converted into divergent form:
\[ \frac{\partial A}{\partial t} + \frac{\partial}{\partial x} F(A) = B. \] (7)

Here \( A = (\rho_g, \rho_d, \rho_g \vec{u}, \rho_d \vec{u}, \epsilon + v^2/2) \) — a vector consisting of the required macroparameters and their combinations, \( F(A) \) — a vector obtained by reducing the equations to divergent form, \( B \) — a vector containing drag terms and associated cooling.

The scheme looks like this. Let \( \Delta t \) — time step, \( \Delta x \) — grid step, \( n \) — current time integration step, \( j \) — number of the current considered cell, then

\[ \tilde{A}_{j+1/2} = \frac{A^n_j + A^{n+1}_j}{2} - \frac{\Delta t}{2\Delta x} \left[ F(A^n_{j+1}) - F(A^n_j) \right] \] (8)

\[ A^{n+1}_j = A^n_j - \frac{\Delta t}{\Delta x} \left[ F(A^n_{j+1/2}) - F(\tilde{A}^{n}_{j-1/2}) \right] \] (9)

Using the predictor-corrector scheme allows obtaining second order accuracy in time and space.

In addition, to reduce the influence of nonphysical oscillations on the solution, it was decided to introduce artificial viscosity by modifying \( F \):

\[ F'(A) = F(A) - D \Delta x^2 \nabla A, \] (10)

where \( D = \text{const} \) — viscosity parameter.

2.3. OpenFPM as instrument for parallelization and basic data structures

To solve particular problems of dusty gas dynamics we implemented numerical methods described above, using data structures of OpenFPM library [5]. The library is written in C++ and designed for easy and fast parallel realization of particle, mesh, and particle-mesh numerical methods. Once the numerical method is implemented on OpenFPM data structures, no extra development efforts are required to run it on processors (CPUs) within a single personal computer and supercomputers using MPI technology, and on graphic cards (GPUs). Moreover, there is a possibility of fine manual tuning when working with distributed memory for maximum efficiency.

The authors of the library showed in [5] that the code implemented using OpenFPM works mostly better (computational speed, scalability) than similar packages (LAMMPS, DualSPHysics, AMReX). Moreover, these tests covered various ideologies. So, for example, for comparison with LAMMPS, the classical problem of molecular dynamics was solved using the Lennard-Jones potential by the distribution of particles on the computational grid, and for comparison with DualSPHysics — the hydrodynamic problem simulating the collision of water with an obstacle, solved by the SPH method.

The most primitive OpenFPM data structure is vector. The interface of use is similar to the standard vector of the C++ language, but with automatic parallelization, implemented within the package. A more serious structure, used primarily for describing particles, is the distributed vector dist, into which coordinates are already "included" (dimensions from 1 to 3 are allowed), and additional parameters (for example, velocity or mass) can be added using the aggregate template. It is convenient to describe the search for neighboring particles using lists (Cell-list or Verlet-list) to further account for their interaction. Grid calculations can be done using a data structure that describes cells distributed in the Cartesian space grid_dist.

The authors of the package provide built-in procedures for generating the distribution of particles in a given geometric primitive. For example, the DrawBox command uniformly fills the specified volume in the form of a parallelepiped (1D/2D/3D) with the specified number of particles. And the DrawSkin command will help you select particles that are near the boundary, which is useful for describing boundary conditions.
3. Code verification and parallelization efficiency

To verify the implemented SPH-IDIC algorithm based on OpenFPM library, two one-dimensional problems were solved: DustyWave and DustyShock. The first is the problem of the propagation of a sound wave, and the second is the shock-tube or Riemann problem in a gas-dust medium. For dusty gas with intense interphase interaction, reference solutions for both problems are known, which are constructed on the basis of solutions of similar problems for a pure gas with a modified sound speed (see e.g. [10, 11] for details)

\[ c_s' = \frac{c_s}{\sqrt{1 + \varepsilon}} \]  

where \( c_s \) is the sound speed in pure gas. However, for moderate momentum exchange, the reference solution of DustyShock is unavailable. For such regime verification, we solved this problem by an fundamentally different method (grid), implemented on the basis of the same OpenFPM package.

The efficiency of the parallel implementation of the SPH-IDIC method on OpenFPM structures for supercomputers with distributed memory was investigated using the three-dimensional Sedov blast wave problem in dusty gas.

3.1. Dustywave

The one-dimensional problem of the propagation of a sound wave in a medium is formulated as follows: at the initial moment of time, the tube is a homogeneous isothermal medium with a small sinusoidal density and velocity perturbation. It is necessary to consider the dynamics of this perturbation, assuming that the boundary conditions are periodic.

The input parameters of the problem were: the density of the gas \( \rho_g = 1 \), the density of the dust \( \rho_d = \varepsilon \rho_g \), \( \varepsilon = 2.5 = \text{const} \), perturbation amplitude 0.01, time of the end of calculation \( t = 0.4 \), the relaxation time \( t_{\text{stop}} = 10^{-3} \).

![Figure 1. Propagation of a sound wave in a dusty gas with a short relaxation time. The black dotted line shows the initial position of the disturbance in the gas density, the red solid line — the position of the wave at the time \( t = 0.4 \) for a mixture with dust to gas mass ratio equal to unity, red squares — the analytical solution of this case, the blue solid line — the position of the wave at the moment of time \( t = 0.4 \) for a mixture with dust to gas mass ratio equal to eight, in blue squares — the analytical solution of this case. 64 cells were used, evenly spaced in the computational domain.](image)
step $\Delta t = 10^{-3}$. The results of numerical simulations are shown in figure 1. One can see perfect match of numerical solution with the reference.

The solution of this problem is probably the simplest demonstrative example of the OpenFPM performance on user-defined problem, which is not included in the demo tests supplied with the library.

3.2. Dustyshock

The one-dimensional problem of the propagation of a shock wave in a medium (the Sod's problem) is formulated as follows: there is a partition in the tube, to the left of it — a stationary high-density hot gas, to the right — a stationary cold gas of low density. At the initial moment of time, the partition instantly disappears, thereby starting the evolution of the contact discontinuity (there is a discontinuity in the medium in terms of density, internal energy, and, as a consequence, pressure). The dynamics of the macroparameters of the medium is investigated.

The initial state was set as follows:

$$
(\rho_g, v, u, \varepsilon, \epsilon, p) =
\begin{cases}
(1.0, 0, 0, 0.3, 2.5, 1) & 0 < x < 0.5; \\
(0.125, 0, 0.3, 2, 0.1) & 0.5 < x < 1.0.
\end{cases}
$$

(12)

The dust density is $\rho_d = \varepsilon \rho_g$, the relaxation time of the dust component is $t_{\text{stop}} = 0.1$. The control time of the calculation $t = 0.2$, the final distribution of the densities and velocities of the fractions in the pipe are shown in Fig. 2.

**Figure 2.** Propagation of a shock wave in a gas-dust medium with a moderate relaxation time. The density values of the gas and the dust (left panel), and the velocity values of the gas and the dust (right panel) are shown at the time $t = 0.2$. Red and blue solid lines correspond to the Lax-Wendroff results, green and cyan solid lines — to SPH-IDIC results.

Two numerical codes were developed based on the OpenFPM package to solve this problem. In the first one, the grid method of calculation according to the Lax–Wendorff scheme was implemented taking into account the artificial viscosity (parameter $D = 8$). To construct the graph in figure 2, a grid with the number of cells 512, time step $\Delta t = 10^{-3}$ was used. In the second code, the SPH-IDIC method was implemented, the smoothing length $h = 5 \times 10^{-3}$, the number of SPH particles of each fraction 2048, the time step $\Delta t = 10^{-3}$. The results of numerical calculations by both methods show almost perfect agreement with each other. Both implementations of the code gave an identical result both in the single-processor calculations and in the case of parallel calculations on a personal computer.
An important result of solving this problem is the establishment of the fact that it is possible to parallelize the algorithm for calculating the interphase interaction IDIC, as well as its correct implementation.

3.3. Sedov blast wave

Figure 3. The results of simulation of Sedov blast wave problem at the time $t = 0.4$ for an explosion energy equal to 1, in a gas with a density $1$ and a specific internal energy $2 \times 10^{-9}$, dust to gas mass ratio $\varepsilon = 0.1$, velocity relaxation time $t_{\text{stop}} = 10^3$ (all values are dimensionless). The distribution of the interpolated parameters in the calculation area is shown: green — gas velocity, red-blue — gas density, red-yellow — specific internal energy.

The three-dimensional problem of the propagation of a spherical shock wave resulting from a point explosion in a homogeneous medium (Sedov’s problem) is formulated as follows. A homogeneous stationary gas of known density fills the space uniformly. At the initial moment of time, a powerful explosion occurs at some point, characterized by a large amount of energy. The dynamic evolution of a spherically symmetric shock wave in this medium is investigated. An example of the calculation result is shown in figure 3.

In numerical experiments to study the efficiency of parallelization, the number of particles (from 1 million to 27 million of each phase), the number of MPI processes, and the parameter $h$, the smoothing radius of the SPH method, were varied. From the point of view of the parallel algorithm, the last parameter determines the cell size for finding the nearest neighbors of a given particle: the larger the cell size, the more neighboring particles must be processed for a given particle (i.e., the execution time depends on the number of neighboring particles quadratically). For test experiments, we used a server computer based on a 48-core AMD EPYC 7002 Series processor with a clock frequency of 2.3-3.3 GHz (96 virtual cores based on Hyperthreading technology) and 192 GB of RAM.

Figure 4 shows the wall-clock time for one time step measured on Sedov blast wave in dusty gas. For a relatively small number of particles ($1 + 1$ million) and a smoothing radius $h = 1.25 \times 10^{-2}$, the parallel implementation shows acceptable acceleration up to 32 MPI processes. With a further increase in number of processors, there is practically no increase
Figure 4. Wall-clock time of the one step computing of Sedov blast wave problem on a supercomputer. The legend indicates the number of particles of every phase and the smoothing radius. See text for details.

in performance. This is due to the growing volume of communication between processes in comparison with the relatively small time spent on calculations (about 1 second). For a larger number of particles ($8 + 8$ million) with the same smoothing radius $h = 1.25 \times 10^{-2}$, the amount of computation increases, and the acceleration is almost linearly dependent on the number of MPI processes. The loss of performance when the number of MPI processes is 49 or more is due to the use of Hyperthreading technology (so-called virtual cores), which does not provide equivalent performance (a slight acceleration of the order of 5-10% is observed only when all 96 virtual cores are used). Similar results are observed for calculations with $1 + 1$ million particles and smoothing radius $h = 2.5 \times 10^{-2}$ and for calculations with $27 + 27$ million particles and smoothing radius $h = 0.625 \times 10^{-2}$. Thus, good scaling performance has been achieved in the strong sense (that is, increasing the number of processors without increasing the amount of computation) and in the weak sense (an increase in the number of processors with an increase in the amount of computation).

The total computation time with 1000 time steps for $8 + 8$ million particles and 48 MPI processes is about 2 hours.

4. Discussion and conclusions
The article presents preliminary results of the development of a numerical model of the dynamical dusty gas based on the SPH-IDIC particle method for supercomputers with distributed memory. The model is built on the data structures of the OpenFPM library and uses the algorithms for parallelization and dynamic load balancing of processors implemented in this library. The implementation of the SPH-IDIC method using the OpenFPM library was verified on problems with reference solutions and by comparison with the results found by alternative numerical
schemes.

We found that the advantages of using the OpenFPM library structures for parallel implementations of the particle method are

- fully automated process of code parallelization using MPI, OpenMP or CUDA technology for user-defined algorithm from scratch,
- high performance of the implemented standard procedures for finding neighbors, and calculating the trajectories of model particles,
- ability to implement boundary conditions of different types for Lagrangian and Eulerian models using built-in library procedures.

To measure the efficiency of parallelization of SPH-IDIC on distributed memory supercomputers, we run the code with up to 54 million particles on up to 100 processors. Our measurements indicate that with sufficient load on each processor, the achieved acceleration almost linearly depends on the number of MPI processors. The results of test experiments give reason to assume that the developed parallel implementation will show acceptable acceleration and good performance for numerical experiments with about 100 million SPH particles when using a larger number of MPI processes (200 MPI processes or more) running on a multiprocessor supercomputer with distributed memory.

Acknowledgments
The research was founded by the Russian Science Foundation grant number 19-71-10026.

References
[1] Jin S 2012 Rivista di Matematica della Universita di Parma, 3 177–216.
[2] Stoyanovskaya O P, Glushko T A, Snytnikov N V and Snytnikov V N 2018 Astron. & Comp., 25 25-37
[3] Stoyanovskaya O, Davydov M, Arendarenko M, Isaenko E, Markelova T and Snytnikov V 2021 J. Comp. Phys., 430 110035
[4] Davydov M N, Stoyanovskaya O P and Glushko T A 2020 J. Phys.: Conf. Ser., 1675 012071
[5] Incardona P, Leo A, Zaluzhnyi Ya, Ramaswamy R and Sbalzarini I F 2019 Comp. Phys. Comm., 241 155-77
[6] Gingold R A and Monaghan J J 1977 MNRAS, 181 375–89
[7] Lucy L B 1977 ApJ, 82, 1013-24
[8] Monaghan J J 1992 An. Rev. A&, 30 543–74
[9] Monaghan J J and Kocharyan A 1995 Comp. Phys. Comm., 87 225–35
[10] Laibe G and Price D J 2012 MNRAS, 420 2345–64
[11] Markelova T, Arendarenko M, Isaenko E and Stoyanovskaya O 2021 J. Appl. Mech. Tech. Phy., 62 4