NUMERICAL SIMULATION OF SHOCK WAVE PROPAGATION IN 2-D CHANNELS WITH OBSTACLES FILLED WITH CHEMICALLY REACTING GAS SUSPENSIONS

by

Yulia KRATOVA*, Alexander KASHKOVSKY, and Anton SHERSHNEV

Khristianovich Institute of Theoretical and Applied Mechanics SB RAS,
Novosibirsk, Russia

Original scientific paper
https://doi.org/10.2298/TSCI19S2623K

Modification of the serial Fortran code for solving unsteady 2-D Euler equations for the mixture of compressible gas and polydisperse particles was carried out using OpenMP technology. Modified code was verified and parallel speed-up was measured. Analysis showed that the data on parallel efficiency is in a good agreement with the Amdahls law, which gives the estimate for serial code fraction about 30%. Parallel code was used for the numerical simulation of two test-cases, namely shock wave propagation in 2-D channel with obstacles filled with reactive Al-O_2 gas particle mixture and heterogeneous detonation propagation in polydisperse suspensions. For the first test-case the data on particles distribution in the flow was obtained, the existense of particle free zones inside the vortices was demonstrated and the attenuation of a shock wave was studied. In the second test, numerical simulation of detonation shock wave propagation in plain 2-D channel for the three polydisperse mixtures was carried out and data on detonation regimes was also obtained.

Key word: *numerical simulation, gas particle mixture, shock, detonation*

Introduction

Investigations of shock wave and heterogeneous detonation propagation in gas-particle mixtures are important for providing the explosion safety in various industrial processes [1, 2] and for development of a detonation engine [3, 4]. This research area is often characterized by lack of fundamental knowledge about the physical processes responsible for initiation, propagation, and decay of explosive and detonation waves, as well as difficulties in experimental modeling of these processes. All this restricts the possibility of solving applied problems and developing adequate technological regulations for modern technologies. An important issue is the distribution of particle sizes because such media usually consists of several fractions of particles. Problems of practical interest require 3-D computations with hundreds of millions of cells, which can be performed only with an employment of modern parallel supercomputers. The present paper is aimed at the modernization and parallelization of the existing code for numerical simulation of the heterogeneous detonation in gas-particles suspensions. At the first stage the parallelization was carried out using the OpenMP technolo-

* Corresponding author, e-mail: yulia@itam.nsc.ru

gy, the obtained parallel speed-up and efficiency were measured on modern computational architectures. The results of numerical simulation, performed using the updated code, are given along with discussion on advantages and drawbacks of used approach.

Governing equations

The computations are based on a mathematical model of detonation of aluminum particles in oxygen [5], which was verified against experimental data for the detonation velocity as a function of the fraction of particles [6]. The experimental data [7] were used to calibrate Al particle burning rate model. The model is based on the concept of interpenetrating continua. The governing equations that describe the detonation flow in monodisperse gas suspensions of aluminum particles for a 2-D unsteady case are presented in [8]. In the present work, we use a mathematical model of a polydisperse heterogeneous medium with several fractions of particles [9]: one fraction consists of reacting particles, and the other fractions consist of inert particles.

The governing equations can be written:

$$\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}_x}{\partial t}+\frac{\partial \mathbf{F}_y}{\partial t}=\mathbf{S}$$

$$\mathbf{Q}=\begin{pmatrix}\tilde{\mathbf{Q}}\\ \{\hat{Q}^i\}\end{pmatrix} F=\begin{pmatrix}\tilde{\mathbf{F}}_\alpha\\ \{\hat{\mathbf{F}}^i_\alpha\}\end{pmatrix},\ \mathbf{S}=\begin{pmatrix}\tilde{\mathbf{S}}_\alpha\\ \{\hat{\mathbf{S}}^i_\alpha\}\end{pmatrix},\ \tilde{Q}=\begin{pmatrix}\rho\\ \rho\mathbf{u}\\ E\end{pmatrix}$$

$$\tilde{\mathbf{F}}_\alpha=\begin{pmatrix}\rho u_\alpha\\ \rho\mathbf{u}u_\alpha+p\mathrm{e}_\alpha\\ (E+p)u_\alpha\end{pmatrix},\ \tilde{\mathbf{S}}=\sum_{i=1}^{N_{Dp}}\hat{\mathbf{S}}^i$$

$$\hat{\mathbf{Q}}^i=\begin{pmatrix}\rho^i\\ \rho^i\mathbf{u}^i\\ E^i\end{pmatrix},\ \hat{\mathbf{F}}^i_\alpha=\begin{pmatrix}\rho^i u^i_\alpha\\ \rho^i\mathbf{u}^i u^i_\alpha\\ E^i u^i_\alpha\end{pmatrix}$$

$$\hat{\mathbf{S}}^i=\begin{pmatrix}-J^i\\ \mathbf{f}^i-J^i\mathbf{u}^i\\ q^i+\mathbf{f}^i\mathbf{u}^i\end{pmatrix},\ \ \alpha\in\{x,y\},\ i=1,\ ...,\ N_{DP}$$

where tilde symbol (~) above pseudo-vectors denotes carrier phase variables, while hat symbol (∧) denotes disperse phase variables. In this notation, ρ is the carrier phase (gas) density, $\mathbf{u}\equiv(u,\,v)\equiv(u_x,\,v_x)$ – the gas velocity vector, E – the total energy of the gas phase, p – the gas pressure, and e_α, $\alpha\in\{x,\,y\}$ are the unity vectors along the corresponding a direction. The variables for the i^{th} disperse phase are indicated with i superscript ($i=1,\ ...\ ,N_{DP}$, N_{DP} is a total number of disperse phases), ρ^i, $\mathbf{u}^i$, and E^i are density, velocity, and energy of the respective disperse phase. The **S** source terms describe the gas-particles interaction, including the drag force, $\boldsymbol{f}^i$, heat transfer term, q^i, and mass transfer term, J^i.

The drag forces $\boldsymbol{f}^i$ acting on the i^{th} species of particles and the heat exchange source terms are calculated:

$$\mathbf{f}^i=\frac{3m^i\rho^i_p}{4d_i}C^i_D\left|\mathbf{u}-\mathbf{u}^i\right|\left(\mathbf{u}-\mathbf{u}^i\right),\ q^i=\frac{6m^i\lambda}{d_i^2}\mathrm{Nu}\left(T-T_i\right),\ i=1,\ ...,\ N_{DP}$$

The reduced mechanisms of chemical reactions are given by the expressions of the form:

$$J^i = \frac{\rho}{\tau_\xi^i}\max\left(0,\xi^i - \xi_k^i\right)\exp\left(\frac{E_\alpha}{\mathrm{R}T_i}\right) \text{ for } T_i \geq T_{\mathrm{ign}}^i$$

$$J^i \equiv 0 \quad \text{for} \quad T_i < T_{\mathrm{ign}}^i$$

The drag coefficient c_D^i is calculated taking into account the Reynolds and Mach numbers based on the diameter of the particles and the velocity of the relative motion of phases. The employed formula was validated against the experimental data on the trajectories of particles interacting with the shock wave [10]

$$c_D^i(\mathrm{Re}_i, M_i) = \left[1+\exp\left(-\frac{0.43}{M_i^{4.67}}\right)\right]\left(0.38+\frac{24}{\mathrm{Re}_i}+\frac{4}{\sqrt{\mathrm{Re}_i}}\right)$$

$$\mathrm{Nu}_i = 2 + 0.6\mathrm{Re}_i^{1/2}\mathrm{Pr}_i^{1/3}$$

$$\mathrm{Re} = \frac{\rho_p^i d_i \left|u-u^i\right|}{\mu},\ M_i = \frac{\left|u-u^i\right|}{a}$$

where T is the temperature of the gas phase, T^i – the temperature of i^{th} particulate phase, and q^i – its heat release. Here ρ_p^i – the density of i^{th} phase particles material (also called true density of the i^{th} phase). The dimensionless mass fraction are defined as $\xi_i = \rho_i/\bar{\rho}$, $\bar{\rho} = \sum_i \rho_i$, m^i – the volume fraction of the i^{th} phase (here for the sake of brevity we denote gas density $\rho \equiv \rho_0$). The value ξ^i is the minimum admissible fraction of particles, d_i – the diameter of particles, λ – the gas thermal conductivity, μ – the gas viscosity, a – the speed of sound in gas, Re_i, Nu_i, M_i, and Pr_i are Reynolds, Nusselt, Mach, and Prandtl numbers, E_a – the activation energy of the reaction, T^i – the ignition temperature, and the τ_ξ^i – the characteristic time of burning.

Conversion of the serial code using OpenMP technology

The existing serial program in Fortran is based on the Harten TVD scheme with the Roe Riemann solver for the carrier phase and the Gentry-Martin-Daly upwind difference scheme for the disperse phase on Cartesian grids. It is a well-tested and verified code, which, however, puts serious limitations on the grid resolution and the size of the problem due to its serial nature. There are different approaches to parallelization of the computational codes. One of the simplest way to increase the efficiency on the program is to employ the OpenMP technology [11], which allows to create computational threads running in parallel on CPU cores. Each thread in theory can execute its own set of instructions. However, one of the more popular approaches is to use OpenMP for parallel execution of various loops in computational codes. Loop iterations are split evenly between OpenMP threads and each performs its part of the computations. Thus the process of parallelization is relatively straightforward: one needs to use special OpenMP notation to declare computationally intensive loops as sections for parallel execution. It should be noted, that often loop cannot be split into non-overlapping sets of data (*e. g*. computation affects few adjacent cells, so the parallel computation in two neighboring cells can lead to a data corruption). In this case such routines cannot be parallelized using aforementioned simplistic approach. The occurrence of serial code naturally leads to a decrease in parallelization efficiency.

For the program under consideration the proposed OpenMP parallelization was carried out for the most part of computationally intensive loops. Special attention was paid to make as few changes in the code as possible. The verification was made by comparing resulting flowfields to the result obtained using the serial code. After that speedup and parallelization efficiency were measured. For this purpose, the series of computations using increasing number of CPU cores were carried out. The fig. 1 shows the wall-clock time per step and parallel speed-up as functions of the number of CPU cores for the parallelized program. It can be seen, that for number of cores greater than four, computational time decreases very slowly. As for speed-up S_p, it is usually defined as a ratio of the wall-clock time of computation carried out on a single CPU t_1 to the wall-clock time of computation carried out on p CPU cores t_p:

$$S_p = \frac{t_1}{t_p}$$

By definition, the ideal speed-up should be directly proportional to the number of CPU cores. Figure 1 demonstrates the measured speed-up growth for the PtcDeton code. It is clearly seen that speed-up growth slows down for large number of cores, and S_p does not exceed three even when using 16 cores of a CPU. This effect can be explained by the serial fragments of code, reducing the efficiency of parallel computations. The theoretical speed-up when using multiple processors is usually estimated using the Amdahl's law [12], which can be formulated as follows. If $\alpha \in (0, 1)$ is the fraction of operations in a program, which can performed only in serial way, and $1 - \alpha$ is the fraction of operations that can be performed in a parallel way (without any overheads), then maximum theoretical speed-up S_p^* will be:

$$S_p^* = \frac{1}{\alpha + \dfrac{1-\alpha}{p}}$$

Figure 1 also shows the Amdahl's law dependency for the value of $\alpha = 0.3$. It is seen that this value yields a good agreement with measured speed-up. This result indirectly confirms that the fraction of unparallelized code in PtcDeton is about 30% and running it on more than four core of a CPU leads to an inefficient usage of computational resources.

Further parallelization of PtcDeton solver will require a massive code refactoring, including introduction of new data structures, reworking the numerical algorithm, introduction of MPI routines for data exchange, *etc*. Therefore, terms of the development time it would be more efficient to re-implement the solver using the HyCFS code, which was designed for the numerical simulation of chemically reacting flows on a number of different architectures, including hybrid CPU/GPU clusters.

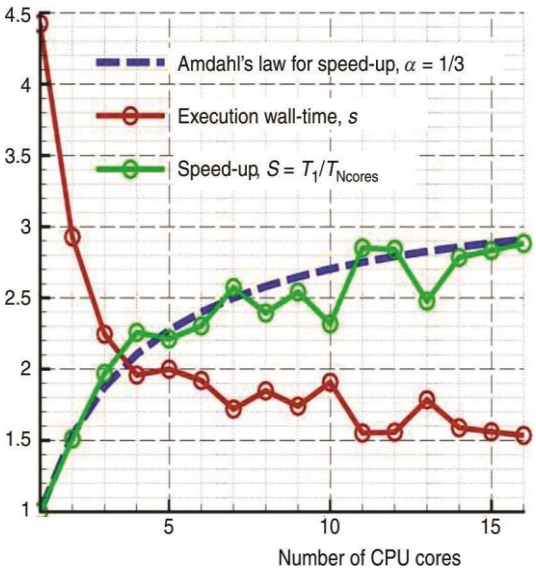

Figure 1. Wall-clock time per step in seconds and parallel speed-up as functions of number of CPU cores

***Test case 1*: Shock wave propagation in 2-D channel with baffle plates**

Shock wave propagation in 2-D channel with three baffle plates was taken as a test case, fig. 2. The reactive stoichiometric mixture of oxygen O_2 and Al particles with diameters d = 2 and 3.5 μm was considered. The volume concentration of particles was set equal to $6 \cdot 10^{-4}$. The initial shock wave Mach number was equal to 1.6. All computations are carried out on a uniform Cartesian grid consisting of 5000×330 points.

Geometry with baffle plate obstacles is treated in a following simple way. At each time step, the computation is performed in the entire region. After that, the boundary values in the respective cells between baffle plate surface are re-imposed according to the specular wall conditions conditions. The conditions at the inlet (left) boundary are imposed according to problem specifics. For the shock-wave diffraction problem, the condition of supporting piston (the values of a final equilibrium state behind the shock wave in the gas-particle mixture) is specified. The initial state is specified at the right boundary, the computation is continued until this boundary is reached by the front. The main goals of the paper are to carry out methodical work on the specification of boundary conditions and to define the influence of shock wave intensity on wave behavior.

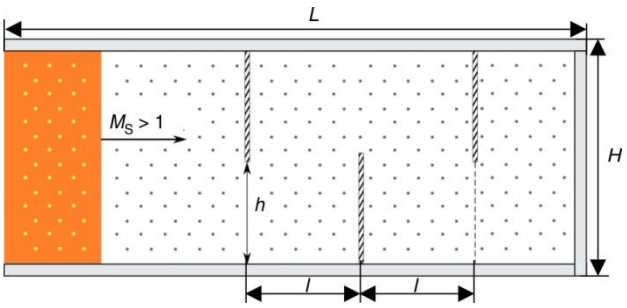

Figure 2. Computational domain geometry and initial conditions set-up

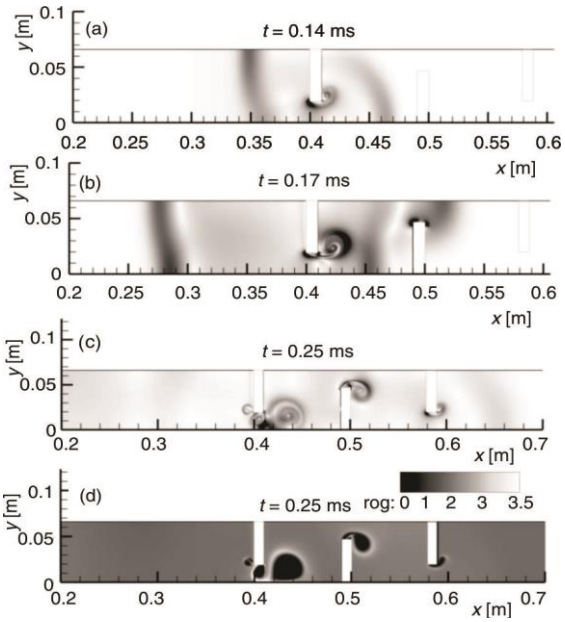

Figure 3. Shock wave propagation in the 2-D channel with three baffle plates; schlieren visualization of gas phase (a)-(c) and particle phase density field (d) for different time moments

In fig. 3 Schlieren visualization of gas and particle phases for different time moments is shown. When interacting with baffle plates the initial plain shock wave becomes distorted and its multiple reflections result in significant decrease of the shock wave intensity. More detailed illustration of the attenuation process is shown in fig. 4, where evolution of centerline gas density distribution can be seen. The grey bars indicate the locations of baffle plates. One can see the shock wave intensity decreases several-fold, and its profile widens. It effectively means that the initial shock wave has degraded into smooth pressure rise.

Figure 3(d) shows particle phase density field at the time, appropriate fig. 3(c). The wave pattern leads to a significant redistribution of particles in the flow. The particle free zones inside the vortices generated by the baffle plate edges are formed.

***Test case 2*: Detonation propagation in 2-D channel**

In the second test case the numerical simulation of detonation shock wave propagation in plain 2-D channel for the conditions from the [13] was performed. Mixtures of three fractions of particles with the diameters of 1, 2, and 3.5 μm with three different fractional compositions are considered. The composition is characterized by the so-called fraction saturation parameter, defined as: $\eta_i = \rho_i / \tilde{\rho}$, where $\tilde{\rho} = \sum_i \rho_i$. The total mass fraction of particles is equal to 0.55, *i. e.*, close to the stoichiometric composition and fractions saturations for different compositions are given in tab. 1.

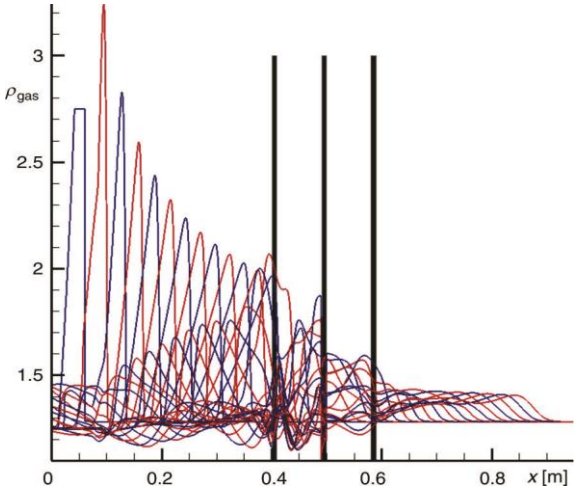

Figure 4. Evolution of centerline gas density distribution; grey bars show the locations of baffle plates

Table 1. Particle fractions composition used in computations

	η_1, 2 μm particles	η_2,1 μm particles	η_2, 3.5 μm particles
Composition 1	0.8	0.1	0.1
Composition 2	0.6	0.2	0.2
Composition 3	0.4	0.3	0.3

The initial conditions and the computation domain were taken the same as in *Test 1*. In the numerical simulation of cellular detonation intial small perturbations are imposed in a form of particle density fluctuations in the cloud filling the plane channel [8].

In fig. 5 the maximum pressure story images, which indicate the trajectories of the triple points, are shown. The same as in [13] the cellular detonation degeneration is observed as the saturation of the mean fraction decreases. As a result, the maximum value of pressure in triple points also becomes smaller, see figs. 5(a) and 5(b). When the saturation of the second fraction becomes smaller than 0.4, complete degeneration of cellular detonation is observed and the detonation wave propagates in a plane regime.

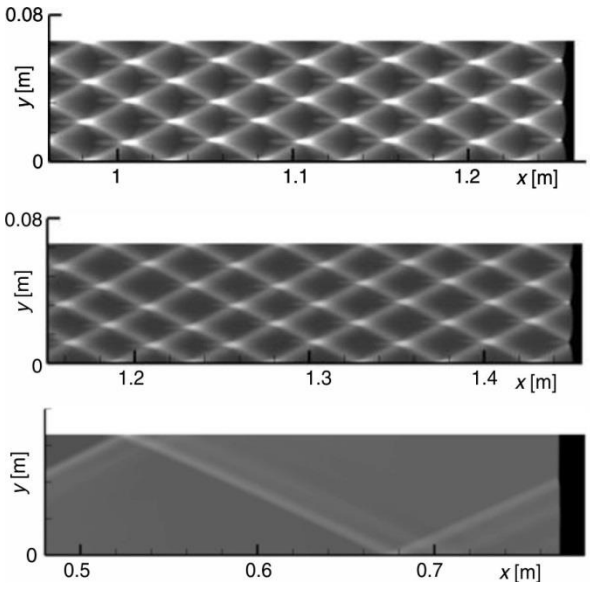

Figure 5. Effect of the fractional composition on detonation propagation; (a) 80% saturation of the second fraction, (b) 60 %, (c) 40 %

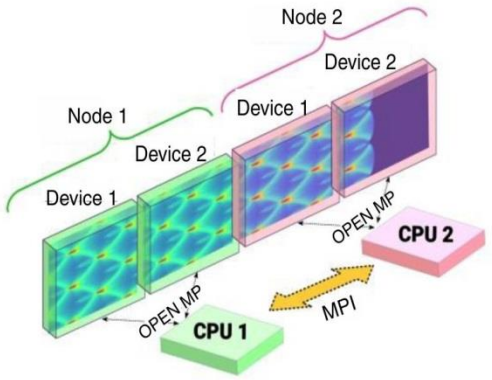

Figure 6. Scheme of HyCFS parallelization

Conclusion and future work

Serial Fortran code for solving unsteady 2-D Euler equations for the mixture of compressible gas and polydisperse particles was parallelized using OpenMP. Parallel code was used for the numerical simulation of two problems: shock wave propagation in 2-D channel with obstacles filled with reactive Al-O_2 gas particle mixture and heterogeneous detonation propagation in polydisperse suspensions. It was shown that obtained parallel speed-up is in a good agreement with the Amdahl's law, which gives the estimate for a serial code fraction of about 30%.

The future work will include re-implementation of the mathematical model of chemically reacting gas-particle suspension flows using the HyCFS-R numerical code [14, 15] originally developed for simulation of non-equilibrium chemically reacting flows. This code was designed specifically for efficient execution on a variety of computational architectures, including GPU, CPU, and Intel Xeon Phi co-processors. Parallel control of the GPUs is ensured by OpenMP, and the data transfer between the nodes is provided by MPI, fig. 6. OpenMP threads are used for computations on multicore CPU and Intel Xeon Phi co-processors. Data structures in this code allow for storing in each cell arbitrarily large set of variables. It becomes a key feature when implementing the many-velocity many-temperature models of heterogeneous media. The computation of the carrier phase is performed using standard HyCFS-R routines based on the shock-captuting TVD schemes and explicit Runge-Kutta TVD time integration schemes. The disperse phase evolution is done using additional routines for computation of convective and source terms.

Acknowledgment

This work was supported by Russian Foundation for Basic Research grants No. 18-08-01442.

References

[1] Nettleton, M. A., Recent Work on Gaseous Detonations, *Shock Waves, 12* (2002), 1, pp. 3-12
[2] Novozhilov, V., Fire Suppression Studies, *Thermal Science, 11* (2007), 2, pp. 161-180
[3] Roy, G. D., *et al*., Pulse Detonation Propulsion: Challenges, Current Status, and Future Perspective, *Prog. Energ. Combust. Sci., 30* (2004), 6, pp.545-672
[4] Wang, Z., *et al*., Numerical Simulation of the Nozzle and Ejector Effect on on the Performance of a Pulse Detonation Engine, *Thermal Science, 22* (2018), 3, pp. 1227-1237
[5] Fedorov, A. V., *et al*., Non-Equilibrium Model of Steady Detonations in Aluminum Particles - Oxygen Suspensions, *Shock Waves*, 9 (1999), 5, pp. 313-318
[6] Strauss, W. A., Investigation of the Detonation of Aluminum Powder-Oxygen Mixtures, *AIAA J., 6* (1968), 12, pp. 1753-1761
[7] Dreizin, E. L., On the Mechanism of Asymmetric Aluminum Particlecombustion, *Combustion and Flame, 117* (1999), 4, pp. 841-850
[8] Fedorov, A. V., Khmel, T. A., Numerical Simulation of Formation of cellular Heterogeneous Detonation of Aluminum Particles In Oxygen, *Combustion, Explosion, and Shock Waves, 41* (2005), 4, pp. 435-448
[9] Fedorov, A. V., Khmel, T. A., Formation and Degeneration of Cellular Detonation in Bidisperse Gas Suspensions of Aluminum Particles, *Combustion, Explosion, and Shock Waves, 44* (2008), 3, pp. 343-353

[10] Boiko, V. M., *et al*., Interaction of a Shock Wave with a Cloud of Particles, *Combustion, Explosion, and Shock Waves*, *32* (1996), 2, pp. 191-203
[11] ***, OpenMP API Specification for Parallel Programming, https://www.openmp.org
[12] Amdahl, G. M., Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities, *AFIPS Conference Proceedings 30* (1967), Apr., pp. 483-485
[13] Kratova, Yu. V., *et al*., Specific Features of Cellular Detonation in Polydisperse Suspensions of Aluminum Particles in a Gas, *Combustion, Explosion, and Shock Waves, 47* (2011), 5, pp. 572-580
[14] Shershnev, A. A., *et al*., HyCFS, a High-Resolution Shock Capturing Code for Numerical Simulation on Hybrid Computational Clusters, *Proceedings*, 18th International Conference on the Methods of Aerophysical Research (ICMAR 2016) AIP Conference Proceedings 1770, (Ed. by V. Fomin), American Institute of Physics, Melville, N. Y., USA, 2016, 030076
[15] Kudryavtsev, A. N., *et al*., A Numerical Code for the Simulation of Non-Equilibrium Chemically Reacting Flows on Hybrid CPU-GPU Clusters, *Proceedings,* XXV Conference on High-Energy Processes in Condensed Matter (HEPCM 2017) AIP Conference Proceedings 1893, (Ed. by V. Fomin), American Institute of Physics, Melville, N. Y., USA, 2017, 030054

Paper submitted: September 14, 2018
Paper revised: November 10, 2018
Paper accepted: November 30, 2018