

doi 10.26089/NumMet.v24r423

MARPLE: software for multiphysics modelling in continuous media

Vladimir A. Gasilov

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0001-7574-4061, e-mail: dipm132013@gmail.com

Alexey S. Boldarev

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0001-8344-646X, e-mail: boldar@imamod.ru

Olga G. Olkhovskaya

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0002-1318-4596, e-mail: olkhovsk@gmail.com

Dmitry S. Boykov

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0001-8587-4994, e-mail: boykovds@gmail.com

Yulia S. Sharova

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0003-2125-3085, e-mail: yulia-shar@mail.ru

Nikita O. Savenko

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0000-0002-0142-6375, e-mail: savenkonkt@gmail.com

Alexey M. Kotelnikov

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
ORCID: 0009-0000-1026-4788, e-mail: kotelnikov@phystech.edu

Abstract: The research code MARPLE was originally created to model high-speed dynamic processes caused by the action of high-intensity energy fluxes on matter. At present, it is a universal tool enabling to solve various continuum mechanics problems. In the code, the following physical models were implemented: single-fluid two-temperature MHD model of plasma dynamics, including electron-ion energy exchange and generalized Ohm's law; model of electrical and thermal conductivity taking into account the anisotropy in the magnetic field; radiative heat transfer (pertaining to both optically thin and optically thick media); techniques for taking into account radiative cooling losses, spectral multigroup diffusion transfer, laser radiation propagation etc.; model of multicomponent flow. The MARPLE code utilizes modern computational technologies based on block structured and unstructured meshes. MARPLE works as the MPI application for modern HPC systems. This paper presents examples of problems in plasma dynamics, magnetohydrodynamics, astrophysics and solid thermomechanics solved by means of the MARPLE code.

Keywords: 3D hydrodynamics, high-performance computing, unstructured meshes, multi-scale physics, solid state models.

Acknowledgements: The investigation was funded by the Russian Science Foundation, project No. 21-11-00362, <https://rscf.ru/project/21-11-00362/>. The numerical results were obtained using the equipment of Shared Resource Center of KIAM RAS (<http://ckp.kiam.ru>). The authors are grateful to Dr. G. Bagdasarov for fruitful discussions.

For citation: V. A. Gasilov, A. S. Boldarev, O. G. Olkhovskaya, D. S. Boykov, Yu. S. Sharova, N. O. Savenko, A. M. Kotelnikov, "MARPLE: software for multiphysics modelling in continuous media", Numerical Methods and Programming. 24 (4), 316–338 (2023). doi 10.26089/NumMet.v24r423.



MARPLE: программное обеспечение для мультифизического моделирования в механике сплошных сред

В. А. Гасилов

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0001-7574-4061, e-mail: dipm132013@gmail.com

А. С. Болдарев

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0001-8344-646X, e-mail: boldar@imamod.ru

О. Г. Ольховская

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0002-1318-4596, e-mail: olkhovsk@gmail.com

Д. С. Бойков

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0001-8587-4994, e-mail: boykovds@gmail.com

Ю. С. Шарова

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0003-2125-3085, e-mail: yulia-shar@mail.ru

Н. О. Савенко

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0000-0002-0142-6375, e-mail: savenkonkt@gmail.com

А. М. Котельников

Институт прикладной математики имени М. В. Келдыша (ИПМ РАН), Москва, Российская Федерация
ORCID: 0009-0000-1026-4788, e-mail: kotelnikov@phystech.edu

Аннотация: Исследовательский код MARPLE изначально создавался для моделирования высокоскоростных динамических процессов, вызванных воздействием на вещество интенсивных потоков энергии. В настоящее время это универсальный инструмент, способный решать различные задачи механики сплошных сред. В коде были реализованы следующие физические модели: одножидкостная двухтемпературная МГД-модель динамики плазмы, включающая электрон-ионный обмен энергией и обобщенный закон Ома; модель электро- и теплопроводности с учетом анизотропии в магнитном поле; радиационный теплообмен (модели, относящиеся как к оптически тонким, так и к оптически толстым средам); методы учета радиационных потерь на охлаждение, спектральный многоразовый диффузионный перенос, распространение лазерного излучения и т. д.; модель многокомпонентного течения. В коде MARPLE используются современные вычислительные технологии, основанные на блочно-структурированных и неструктурированных сетках. MARPLE работает как приложение MPI для современных высокопроизводительных вычислительных систем. В данной работе приведены примеры задач динамики плазмы, магнитогидродинамики, астрофизики и термомеханики твердого тела, решенных с помощью кода MARPLE.

Ключевые слова: трехмерная гидродинамика, высокопроизводительные вычисления, неструктурированные сетки, многомасштабная физика, твердотельные модели.

Благодарности: Исследование выполнено за счет гранта Российского научного фонда № 21-11-00362, <https://rscf.ru/project/21-11-00362/>. Численные результаты получены с использованием оборудования ЦКП ИПМ имени М. В. Келдыша РАН (<http://ckp.kiam.ru>). Авторы благодарят Г. А. Багдасарова за плодотворные дискуссии.

Для цитирования: Гасилов В. А., Болдарев А. С., Ольховская О. Г., Бойков Д. С., Шарова Ю. С., Савенко Н. О., Котельников А. М. MARPLE: программное обеспечение для мультифизического моделирования в механике сплошных сред // Вычислительные методы и программирование. 2023. 24, № 4. 316–338. doi 10.26089/NumMet.v24r423.

1. Introduction. In this paper we present an object-oriented parallel multidimensional Eulerian multi-physics code MARPLE. The code was designed in Keldysh Institute of Applied Mathematics Russian Academy of Sciences¹ for scientific numerical simulations using high-performance computing [1, 2].

The code MARPLE [2] was developed as a scientific code. Initially it was designed for modelling of the processes appearing in the experimental pulsed power energetic and high energy density plasmas. Therefore its abbreviated name originates from “Magnetically accelerated radiative plasma explorer”. During the years of the code usage, it has been developed to a universal code which can be applied to different problems of various object domains. The code contains a lot of data structures and algorithms independent on the object domain, for example, the system of unstructured mesh handling. Processing of unstructured meshes is not a trivial problem, especially in the case of a distributed code. In the code MARPLE, the mesh data structures which allow easy mesh handling were developed. Thus, nowadays MARPLE represents itself a numerical platform enabling to conduct multipurpose numerical experiments.

After writing the code, it has been used for solution of problems arising in different scientific fields, viz.:

- physics of high-energy-density plasmas, pinches;
- capillary discharge;
- astrophysics (modelling of evolution of neutron stars);
- solid mechanics, influence of strong energy fluxes on condensed matter.

Below we will consider some of these applications in details.

2. Models. In the MARPLE package, the complexity of the studied physical problems and presence of various physical effects is considered within the total approximation principle or splitting over groups of physical processes. A system of equations for a set of unknowns u

$$\frac{\partial u}{\partial t} = F_1 u + F_2 u + \dots + F_N u$$

with operators (usually differential) F_1, \dots, F_N , describing physical processes of different nature, can be solved as

$$\frac{\hat{u}_1 - u}{\tau} = \mathcal{F}_1(u, \hat{u}_1), \quad \frac{\hat{u}_2 - \hat{u}_1}{\tau} = \mathcal{F}_2(\hat{u}_1, \hat{u}_2), \quad \dots, \quad \frac{\hat{u} - \hat{u}_{N-1}}{\tau} = \mathcal{F}_N(\hat{u}_{N-1}, \hat{u}),$$

where \mathcal{F}_i is a discrete analogue (approximation) of the operator F_i . Such an approach allows us to consider various groups of processes independently and apply the most appropriate numerical technique (explicit or implicit, etc.) to a given group of processes.

The MARPLE package is designed for simulations in three-dimensional domains of complex geometric shapes on irregular grids, including block ones, consisting of elements of various types.

The main physical models realized in the MARPLE software package are given below.

2.1. Single-fluid two-temperature MHD model of plasma dynamics. The MARPLE code implements the MHD model, which has been used in many studies of fast plasma flows arising under the action of an intense energy input from an external source (plasma in a short-pulse electric discharge, laser-generated plasma, etc.). The model takes into account the possible temperature difference between the ionic and electronic components of the plasma. The corresponding system of equations that represents the conservation laws (see [3, 4, 5] etc.) has the following form:²

$$\frac{\partial}{\partial t} \rho + \nabla \cdot (\rho \mathbf{w}) = 0, \tag{1}$$

$$\frac{\partial}{\partial t} \rho w_l + \sum_k \frac{\partial}{\partial x_k} \Pi_{lk} = 0, \quad \Pi_{lk} = \rho w_l w_k + P \delta_{lk} - \frac{1}{4\pi} \left(B_l B_k - \frac{1}{2} B^2 \delta_{lk} \right), \tag{2}$$

$$\frac{\partial}{\partial t} \left(\rho \varepsilon + \frac{1}{2} \rho w^2 + \frac{B^2}{8\pi} \right) + \nabla \cdot \mathbf{q} = 0, \quad \mathbf{q} = \left(\rho \varepsilon + \frac{1}{2} \rho w^2 + P \right) \mathbf{w} + \frac{1}{4\pi} \mathbf{B} \times (\mathbf{w} \times \mathbf{B}), \tag{3}$$

$$\frac{\partial \rho \varepsilon_e}{\partial t} + \nabla \cdot (\rho \varepsilon_e \mathbf{w}) + P_e \nabla \cdot \mathbf{w} = 0, \tag{4}$$

$$\frac{\partial}{\partial t} \mathbf{B} - \nabla \times (\mathbf{w} \times \mathbf{B}) = 0, \tag{5}$$

¹<https://keldysh.ru/index.en.shtml>

²Here and below, the Gaussian system of units is used.



$$\delta_{lk} = \begin{cases} 1, & l = k \\ 0, & l \neq k, \end{cases}$$

$$\varepsilon = \varepsilon_i + \varepsilon_e, \quad P = P_i + P_e, \quad P_i = P_i(\rho, \varepsilon_i), \quad P_e = P_e(\rho, \varepsilon_e).$$

For the used notations see Table of symbols in the end of this work.

2.2. Model of electromagnetic field diffusion. The modeling of the diffusion of the electromagnetic field is carried out taking into account the possible anisotropy of the electrical conductivity caused by the influence of the magnetic field. The investigation of this problem was started in [3, 4] and continued in subsequent works on this topic. The main system of equations (under the condition that the displacement current is neglected) reads

$$\text{rot } \mathbf{B} = \frac{4\pi}{c} \mathbf{j}, \quad \mathbf{j} = \hat{\sigma} \mathbf{E}, \tag{6}$$

$$\frac{1}{c} \cdot \frac{\partial \mathbf{B}}{\partial t} = -\text{rot } \mathbf{E}, \tag{7}$$

$$\mathbf{E} = \frac{\mathbf{j}_{\parallel}}{\sigma_{\parallel}} + \frac{\mathbf{j}_{\perp}}{\sigma_{\perp}}. \tag{8}$$

2.3. Plasma heat balance model. The energy balance equations for the electron and ion components of the plasma are written taking into account the anisotropy of electron heat conduction in a magnetic field, electron-ion energy exchange, Joule heating and radiative heat transport. The system of energy balance equations [3, 4] has the following form

$$\frac{\partial(\rho\varepsilon_e)}{\partial t} = \text{div}(\hat{k}_e \text{grad } T_e) + Q_{ei} + Q_J + Q_R, \tag{9}$$

$$\frac{\partial(\rho\varepsilon_i)}{\partial t} = \text{div}(k_i \text{grad } T_i) - Q_{ei}. \tag{10}$$

The electron-ion energy exchange is equal to

$$Q_{ei} = \kappa_{ei}(T_i - T_e),$$

where $\kappa_{ei}(\rho, T_e)$ is the relaxation coefficient, while Joule heating is specified by the expression

$$Q_J = \frac{\mathbf{j}_{\parallel}^2}{\sigma_{\parallel}} + \frac{\mathbf{j}_{\perp}^2}{\sigma_{\perp}}.$$

The full system of the MHD equations, including (1)–(5), (6)–(8) and (9)–(10), is obtained by supplementing the equations of state:

$$\varepsilon_e = \varepsilon_e(\rho, T_e), \quad \varepsilon_i = \varepsilon_i(\rho, T_i).$$

§ 2.3.1. Radiative heat transfer models.

Multigroup spectral model. This model was developed in accordance with the recommendations [6]. The entire frequency range under study is divided into M spectral groups: $\nu_1 = 0 < \nu_2 < \dots < \nu_m < \dots < \nu_M < \infty$. The radiative energy transfer equation is solved for each group. The computations within each group use the average values of k^P , k^R and J (k^P and k^R are the absorption coefficients averaged according to Planck and Rosseland respectively, J stands for the emissivity; all of them depend on the temperature and density of the medium and on the photon energy). The integral values of the source term Q_R in the energy balance equation are calculated by summing over all spectral groups. In the one-group (“gray matter”) approximation the averaging of k^P , k^R and J is performed over the entire spectrum [7].

Diffusion model of radiative transport. The main model of radiative heat transfer implemented in MARPLE is the model of radiation diffusion in the multigroup approximation over the spectrum. In the diffusion approximation [6, 8], the balance equation for radiative energy in a radiating and absorbing medium has the form

$$-\text{div} \frac{c}{3k^R} \text{grad } U + k^P c U = k^P c J,$$

where U is the radiative energy density, $\mathbf{W} = -\frac{c}{3k^R} \text{grad } U$ is the radiative energy flux.

The boundary condition for a normal flow at the outer boundary (radiation output) is fixed by the requirement

$$W_n = -\frac{c}{3k^R} (\text{grad } U \cdot \mathbf{n}) = -\frac{cU}{2},$$

where \mathbf{n} is the normal vector to the boundary. It is also possible to set boundary conditions with a given incoming radiative flux.

The expression for the radiation energy absorbed by the plasma is written down as

$$Q_R = \text{div } \mathbf{W} = c(k^P J - k^P U).$$

Model of radiative heat conduction in optically thick media. In the case of an optically dense medium in the local thermodynamic equilibrium, one can use the radiative heat conduction model, in which the radiative energy flux is expressed by the following formula:

$$\mathbf{W} = \frac{16\sigma_{SB}T_e^3}{3k^R} \text{grad } T_e, \quad Q_R = \text{div } \mathbf{W}.$$

Volume energy losses due to bremsstrahlung. Plasma cooling due to volume energy losses is described by the relations [8]

$$Q_R = -1.764 \cdot 10^6 \sqrt{T_e} n_e n_i Z^2 \quad [10^5 \text{ J}/(\text{cm}^3 \cdot \mu\text{s})],$$

$$n_i = \frac{\rho}{m_i}, \quad n_e = \frac{\rho}{m_i} Z(\rho, T_e).$$

This model describes the situation adequately in the case of an optically thin medium.

2.4. Two-fluid MHD model of plasma dynamics. In some problems of astrophysics related, for example, to the study of the interaction of galactic gas with supernova remnants, stellar winds, etc., it becomes necessary to analyze separately the motion of the ionized part of the gas and its neutral component. The model of magnetohydrodynamics realized in MARPLE for solving such problems includes the following system of equations [9, 10], which contains two groups of hydrodynamic equations: for ions and for neutrals. The dynamics of the ion component (denoted by index α) is described by the following system of equations:

$$\rho_\alpha = m_\alpha n_\alpha,$$

$$\partial_t \rho_\alpha + \partial_i (\rho_\alpha w_{\alpha,i}) = 0,$$

$$\partial_t (\rho_\alpha w_{\alpha,i}) + \partial_j \left(\rho_\alpha w_{\alpha,i} w_{\alpha,j} + \delta_{ij} \left(p_\alpha + \frac{B^2}{8\pi} \right) - \frac{1}{4\pi} B_i B_j \right) = R_{\alpha\beta,i},$$

$$\partial_t E_\alpha + \partial_i \left(w_{\alpha,i} \left[E_\alpha + p + \frac{B^2}{8\pi} \right] - \frac{1}{4\pi} (\mathbf{w}\mathbf{B}) B_i \right) = Q_{\alpha\beta} + Q_R + Q_{\text{heat}},$$

$$\partial_t B_i + \partial_j (w_j B_i - B_j w_i) = 0,$$

$$E_\alpha = \rho_\alpha \varepsilon_\alpha + \frac{\rho_\alpha \mathbf{w}_\alpha^2}{2} + \frac{B^2}{8\pi},$$

where $Q_R = -\Lambda(T)n_e n_H$ are the radiative losses, $Q_{\text{heat}} = \Lambda(T_0)n_{e0}n_{H0}$ stands for the heating by background matter, $\Lambda(T)$ is the cooling function, n_e and n_H are concentrations of electrons and hydrogen nuclei correspondingly, $\Lambda(T_0)$ is the heating-related additive which is necessary to remain the background plasma unchanged, the parameter T_0 stands for the environment temperature, n_{e0} and n_{H0} — its electron and hydrogen densities, $\mathbf{R}_{\alpha\beta}$ is the friction force between fluids caused by collisions. For a partially ionized medium of complex composition, it is expedient to use a tabular cooling function (for instance, as in [11]), which corresponds to an equilibrium plasma with solar metallicity, or an own function of cooling calculated in advance using data for radiative processes [12].

The system of equations describing the evolution of the neutral component has the following form:

$$\rho_\beta = m_\beta n_\beta,$$

$$\partial_t \rho_\beta + \partial_i (\rho_\beta w_{\beta,i}) = 0,$$



$$\begin{aligned} \partial_t(\rho_\beta w_{\beta,i}) + \partial_j(\rho_\beta w_{\beta,i} w_{\beta,j} + \delta_{ij} p_\beta) &= R_{\alpha\beta,i}, \\ \partial_t E_\beta + \partial_i(w_{\beta,i} [E_\beta + p]) &= Q_{\alpha\beta}, \\ E_\beta &= \rho_\beta \varepsilon_\beta + \frac{\rho_\beta \mathbf{w}_\beta^2}{2}. \end{aligned}$$

§2.4.1. *Collisions between the components.* Energy and momentum are transferred between the components of the medium through collisional processes. To calculate the characteristic collision time $\tau_{\alpha\beta}$, the interaction cross section σ is assumed to be constant [11, 13]:

$$\tau_{\alpha\beta} = \frac{1}{n_\beta \sigma (kT_\alpha/m_\alpha)^{1/2}}.$$

In its turn, the frictional force can be written as

$$\mathbf{R}_{\alpha\beta} = -\frac{m_\alpha n_\alpha}{\tau_{\alpha\beta}} (\mathbf{w}_\alpha - \mathbf{w}_\beta).$$

Then, the energy transfer takes the following form:

$$Q_{\alpha\beta} = -\sum_\beta 3 \frac{m_\alpha}{m_\alpha + m_\beta} \frac{n_\alpha}{\tau_{\alpha\beta}} k_B (T_\alpha - T_\beta) - \sum_\beta \frac{m_\beta}{m_\alpha + m_\beta} (\mathbf{w}_\alpha - \mathbf{w}_\beta) \mathbf{R}_{\alpha\beta}.$$

The heating caused by the frictional force \mathbf{R} is taken into account. It is necessary to maintain the energy balance.

2.5. Multicomponent plasma flow model. The considered model describes the dynamics of a gas or gas-plasma medium consisting of N components (for example, a mixture of different substances; mixtures of thermodynamically identical components are also considered in order to mark a part of the substance in process of studying the flow structure). This model can be used both in problems of hydrodynamics and MHD, and in problems of mechanics of a deformable solid body (Section 2.6). The relations for the components read:

$$\sum_{n=1}^N \alpha_n = 1, \quad \sum_{n=1}^N \rho_n = \rho, \quad \rho_n = \alpha_n \rho.$$

For each component of the mixture, a separate continuity equation is solved, but the momentum and energy equations are common for the entire medium. There is mechanical ($\mathbf{w}_n = \mathbf{w}$) and thermodynamic ($T_n = T$) equilibrium between the components. The model of interpenetrating continua is used, which refers to mixtures of gases, where in any macroscopic volume all components are present in a ratio determined by mass concentrations. Each component fills the volume and is not affected by other components in the same volume. The thermodynamic model of such a mixture is considered in [14].

2.6. Solid state mechanics. The initial system of equations describing thermomechanical effects in an object exposed to an intense energy flow (particles, ionizing radiation) is based on continuum mechanics models. The complete system includes equations and constitutive relations (kinematic, physical) describing the hydrogasdynamics of the evaporated material, elastic, plastic waves, viscoplastic deformations and destruction in the solid residue of the target. Dynamic processes in the material of an object in a wide range of state parameters are modeled in the approximation of local thermodynamic equilibrium. When describing the state of a medium at a sufficiently high rate of change in its structure, a non-equilibrium at phase transitions or at changes in aggregate state of the material is allowed to be taken into account [15]. The equations of dynamics of a continuous medium are written in the form of basic conservation laws. In accordance with the aforesaid, in the region of the evaporated material the computations are performed with help of the equations of the gas-dynamic model:

$$\begin{aligned} \frac{\partial}{\partial t} \rho + \nabla(\rho \mathbf{w}) &= 0; & \frac{\partial}{\partial t} \rho w_i + \sum_k \frac{\partial}{\partial x_k} \Pi_{ik} &= 0; & \Pi_{ik} &= \rho w_i w_k + P \delta_{ik}; \\ \frac{\partial}{\partial t} \left(\rho \varepsilon + \frac{1}{2} \rho w^2 \right) + \nabla \mathbf{q} &= 0; & \mathbf{q} &= \left(\rho \varepsilon + \frac{1}{2} \rho w^2 + P \right) \mathbf{w}; & P &= P(\rho, \varepsilon). \end{aligned}$$

The governing system of equations, used to calculate wave processes in a deformable solid material, corresponds to the model of elastoplastic deformations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \rho \nabla_i w_i &= 0; \\ \rho \frac{\partial w_i}{\partial t} &= F_i + \nabla_j \sigma_{ij}; \\ \rho \frac{\partial E}{\partial t} &= \sigma_{ij} \dot{\epsilon}_{ij} - \nabla_i q_i; \\ q_i &= -\chi(\rho, T) \nabla_i T; \\ \dot{\epsilon}_{ij} &= \frac{1}{2} (\nabla_i w_j + \nabla_j w_i); \\ \frac{\partial D_{\sigma_{ij}}}{\partial t} + 2G \dot{\lambda} D_{\sigma_{ij}} &= 2G \left(\dot{\epsilon}_{ij} + \frac{1}{3\rho} \frac{\partial \rho}{\partial t} \delta_{ij} \right) - 3\alpha K (T - T_0) \delta_{ij}; \\ p &= p(\rho, T); \\ \sigma_{ij} &= -\nabla_i p + D_{\sigma_{ij}} \Psi + ((3\eta - 2\mu) \dot{\epsilon}_{ij} + 2\mu \dot{\epsilon}_{ij}) \Psi; \\ \dot{\lambda} &= \frac{3}{2\sigma_T^2} \sigma_{ij} \dot{\epsilon}_{ij}^{(p)}; \\ \dot{\epsilon} &= \dot{\epsilon}_{ij} \delta_{ij} / 3; \\ \Psi &= \begin{cases} 0, & \text{for } E \geq L_S \text{ or } \sigma_{ij} \geq \sigma^c; \\ 1, & \text{for } E < L_S \text{ and } \sigma_{ij} < \sigma^c. \end{cases} \end{aligned}$$

The pressure is calculated by means of the wide-range equations of state. The modelling of wave processes in a solid material can also be performed in the approximation of a barotropic medium. In this case, we use the equation of state in an analytical form.

§ 2.6.1. *Destruction criteria.*

Maximum stress criterion. The maximum stress criterion corresponds to the maximum ratio of actual stresses to limit stresses, defined in the coordinate system of the layer material. The strength criterion function is written as:

$$\left\{ \left| \frac{\sigma_{11}}{X} \right|, \left| \frac{\sigma_{22}}{Y} \right|, \left| \frac{\sigma_{33}}{Z} \right|, \left| \frac{\sigma_{12}}{S} \right|, \left| \frac{\sigma_{13}}{R} \right|, \left| \frac{\sigma_{23}}{Q} \right| \right\}_{\max} \leq 1.$$

Quadratic Mises–Hill criterion. To simulate the destruction of composite materials, a quadratic criterion is used:

$$\begin{aligned} F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{11} - \sigma_{33})^2 + H(\sigma_{22} - \sigma_{11})^2 + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 &\leq 1; \\ F &= \frac{1}{2} \left(\frac{1}{(\sigma_{22}^c)^2} + \frac{1}{(\sigma_{33}^c)^2} + \frac{11}{(\sigma_{11}^c)^2} \right); \quad G = \frac{1}{2} \left(\frac{1}{(\sigma_{11}^c)^2} + \frac{1}{(\sigma_{33}^c)^2} + \frac{1}{(\sigma_{22}^c)^2} \right); \\ H &= \frac{1}{2} \left(\frac{1}{(\sigma_{22}^c)^2} + \frac{1}{(\sigma_{11}^c)^2} + \frac{1}{(\sigma_{33}^c)^2} \right); \quad L = \frac{1}{2(\tau_{23}^c)^2}; \quad M = \frac{1}{2(\tau_{31}^c)^2}; \quad N = \frac{1}{2(\tau_{12}^c)^2}. \end{aligned}$$

Tuler–Butcher failure criterion. In composite materials, failure upon reaching the maximum stress can lead to rupture at lower stresses. The process of destruction with the influence of the time parameter is interpreted as an integral one, which is set on a certain structural-time range. To determine the strength of composite materials, the integral criterion of destruction of Tuler–Butcher is used:

$$D = \int_0^t (\sigma_0 - \sigma(t))^\lambda dt. \tag{11}$$

Here $\sigma(t)$ is fracture stress, σ_0 is maximum principal stress, λ is a material constant, t is the time when solid cracks, D is another material constant which is referred to as damage integral. The destruction happens at the moment t when the integral in (11) reaches the value D of the damage integral.



2.7. Material equations. For the completion of the system of equations, some expressions for the matter properties parameters (EOS, transport coefficients etc.) should be added. In the code, it is possible to use various types of material equations. For example, in some computations we used the wide-range equations of state (EOS) of matter, transport coefficients and optical properties. The pressures and specific energies of individual plasma components, as well as the average charge of ions depending on the density and temperature, are determined based on the quantum-statistical model described in the book [7] (the Hartree–Fock–Slater model for an average atom, the generalized Thomas–Fermi model). Data on the equations of state and optical properties of matter are preliminarily calculated using the THERMOS program [16] and utilized in computations in the form of data tables. Explicit formulas for calculating the kinetic coefficients that determine the expressions for dissipative flows are given in [17]. Some computations have been performed with help of the thermodynamic model of Jones–Wilkins–Lee [18].

3. Methods. The MARPLE code implements computational technologies using block structured and unstructured meshes consisting of tetrahedral, hexahedral, prismatic elements and their combinations. To perform calculations taking into account the symmetry of the problem, periodic and symmetric boundary conditions are provided.

The solution of the complete system is carried out according to the physical splitting scheme. The main approaches to the construction of numerical algorithms realized in the MARPLE code are described below.

3.1. Non-dissipative MHD. For the modelling of ideal two-temperature MHD, an explicit two-stage (predictor–corrector) finite difference scheme on unstructured meshes is used. This scheme has the second order of approximation in both spatial coordinates and time. The increased order TVD approximation is based on the local spatial reconstruction of the solution by its values at the mesh cells centers [19]. As a base for the values fluxes over the mesh elements, a variant of the 1D modified Lax–Friedrichs scheme is applied.

§3.1.1. Piecewise parabolic method. Although the MARPLE code deals, in general, with unstructured meshes, some models, which work with traditional (i, j, k) -structured meshes, are successfully tested and applied within MARPLE. For example, the piecewise parabolic method is implemented for ideal MHD and Euler equations.

The Euler equations are discretized in space and time using a Godunov-type scheme and a predictor–corrector scheme. The PPM method is employed to interpolate quantities from the cell centers to their faces with subsequent solution of the Riemann problem for the discontinuity evolution. The code implements the Roe, HLL, HLLC, and HLLD methods [20, 21, 22].

3.2. Dissipative processes. Discontinuous Galerkin approximations are applied to the second-order partial differential equations with a variable coefficient in tensor form [23]. These schemes possess the conservativity property (they approximate the energy balance in the computational domain accordingly to the chosen form of flux approximation) and retain the self-adjoint property inherent in the original differential operator.

§3.2.1. Diffusion of the magnetic field. The equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\frac{c^2}{4\pi} \operatorname{rot} \hat{\sigma}^{-1} \operatorname{rot} \mathbf{B}$$

is solved.

For integration over time, an absolutely stable Crank–Nicolson scheme is used, which provides the second order of accuracy. The discontinuous Galerkin scheme is applied to approximate the operator $\operatorname{rot} \hat{\sigma}^{-1} \operatorname{rot} \mathbf{B}$.

To ensure end-to-end calculation in areas with low density, numerical stabilizers are introduced into the magnetic field diffusion equation.

1) Artificial displacement current:

$$\operatorname{rot} \mathbf{B} = \frac{4\pi}{c} \mathbf{j} + \frac{\alpha}{c} \frac{\partial \mathbf{E}}{\partial t}.$$

Here $\alpha > 1$ is a tuning factor that effectively reduces the propagation velocity of electromagnetic wave in a low-density plasma region. The coefficient α is chosen from the condition $\frac{\alpha}{c} \ll \frac{T}{L}$, where T is the characteristic time of the problem, L is the characteristic size of the computational domain.

2) Hyperbolization of the diffusion equation by introducing the second time derivative with a small parameter ω :

$$\frac{\partial \mathbf{B}}{\partial t} + \omega \frac{\partial^2 \mathbf{B}}{\partial t^2} = -\frac{c^2}{4\pi} \operatorname{rot} \hat{\sigma}^{-1} \operatorname{rot} \mathbf{B}.$$

The coefficient ω is chosen to satisfy the condition $\omega \leq \frac{h}{c}$ (with h is the minimum characteristic linear size of the cell over to the computational mesh). For the computations, a three-layer explicit difference scheme of time integration with a limitation on the time step

$$\Delta t \leq \sqrt{\frac{\omega}{\sigma}} h = \frac{h^{3/2}}{\sqrt{c}}$$

is used.

To ensure the solenoidality of the magnetic field $\text{div } \mathbf{B} = 0$, which can be violated due to a numerical error, the magnetic field is calibrated by solving the Poisson equation for the magnetic potential φ :

$$\mathbf{B}^* = \text{grad } \varphi + \text{rot } \mathbf{A}; \quad \text{div}(\text{grad } \varphi) = 0; \quad \mathbf{B} = \mathbf{B}^* - \text{grad } \varphi.$$

Here \mathbf{B}^* is the result of numerical solution of the magnetic field diffusion equation and \mathbf{B} is the calibrated solution. Taking into account the calculated field \mathbf{B} , one can find $\mathbf{j} = \frac{c}{4\pi} \text{rot } \mathbf{B}$ and $\mathbf{E} = \frac{\mathbf{j}_{\parallel}}{\sigma_{\parallel}} + \frac{\mathbf{j}_{\perp}}{\sigma_{\perp}}$. The obtained values are used to evaluate the Joule heating and to analyze the electric current density and electric field strength.

§3.2.2. *Thermal conductivity and electron-ion energy exchange.* An implicit scheme is utilized for time integration and the projection scheme is used to approximate the operator $\langle \text{div } k \text{ grad } T \rangle$. At this splitting step, we fix ρ and set

$$\frac{\partial \varepsilon}{\partial t} = \frac{\partial \varepsilon}{\partial T} \cdot \frac{\partial T}{\partial t}$$

for ions and electrons. The heat capacity $C_v = \frac{\partial \varepsilon}{\partial T}$ is calculated analytically or from EOS tables for the values of T and ρ from the previous layer in time. The thermal conductivity coefficient (taking into account the magnetization) and the relaxation coefficient κ_{ei} are taken from the same layer.

3.3. Mesh ray tracing. The algorithms, based on the tracing of rays over the computational mesh, allow to obtain the numerical solution of the radiation transport equation with respect to the direction of every ray. Let us mention the main tasks where such a technique is useful:

- postprocessing of the computational results for their direct comparison with experimental data. For example, a numerical model of the pinhole camera gives numerical pinhole images which can be easily compared with ones obtained in experiment;
- modelling of substance heating by laser pulses in laser plasma experiments;
- modelling of the radiative heat transfer in plasmas. Solution of the radiative heat transfer problem with use of ray tracing (the so-called long characteristics method) requires enormous computational power and large operating memory. Therefore, it is practically never performed in 3D, but such a technique has been successfully applied in previous years to the 2D problems.

The problem of ray tracing over an unstructured mesh is complicated by a considerable probability of degenerated cases, when a ray passes directly through an edge or a node (or very close to them, then we cannot distinguish these cases because of the limited accuracy of our computations). These degenerated cases, when they are not handled properly, can result in lost or multiple accounting of a ray segment, which leads to a serious error in results of such a modelling. Another complication is concerned with parallel solution of the whole task and the distribution of the computational mesh over the processors — a ray can pass through different parts of the mesh, exit and re-enter a part. As a result, the distribution of computational load between the processors becomes inhomogeneous.

In the code MARPLE, a ray tracing technique based on rational numbers arithmetics is implemented [24, 25]. The rational numbers arithmetics [26] leads to a considerably more time-consuming algorithm compared to traditional floating-point arithmetic. However, this approach allows to obtain the exact solution of the geometrical problem of intersection of a ray with a mesh element. As a result, all degenerated cases become absolutely distinguishable.

3.4. Solid mechanics model. The solution of the equations of hydrodynamics/elastoplasticity over the entire area of the evaporated and non-evaporated material is performed according to explicit difference schemes for different predetermined aggregate states of matter. For this purpose, a special parametrization is introduced



in accordance with the following algorithm: if the energy in the considered region of the sample is less than the sublimation energy and the stresses do not exceed the ultimate strength, then the region belongs to a solid body. The momentum equation for the non-evaporated material takes into account the total stress tensor σ and for the evaporated material, only its spherical part $S = \sigma - D$.

Computation of gas-dynamic waves in the evaporated material and dynamic processes in the area of elastoplastic material is carried out by means of the high-order difference scheme which is a modification of the well-known Lax–Friedrichs scheme for general unstructured grids. Integration over time is implemented using the predictor–corrector scheme. This method has a sufficient level of resolution of wave structures in space and time at a moderate consumption of computational resources.

The developed technique enables to model the processes of interaction between direct and reflected waves and to take into account correctly various kinds of boundary conditions and contact discontinuities. Using appropriate models and failure criteria, this technique makes it possible to simulate the failure of the material and the wave response from the destroyed regions as well.

4. Unstructured meshes and parallel mesh processing. As a rule, computational domains in the studied problems have comparatively complex geometry and there may be some difficulties in discretization of these regions by simple structured computational meshes. Moreover, the adaptation of the mesh to the peculiarities of the solution is needed in some problems. There are different approaches to the problem of discretization of the computational domains and each approach involves its own type of the mesh, the technique of the mesh generation and handling [27, 28] (see Figure 1).

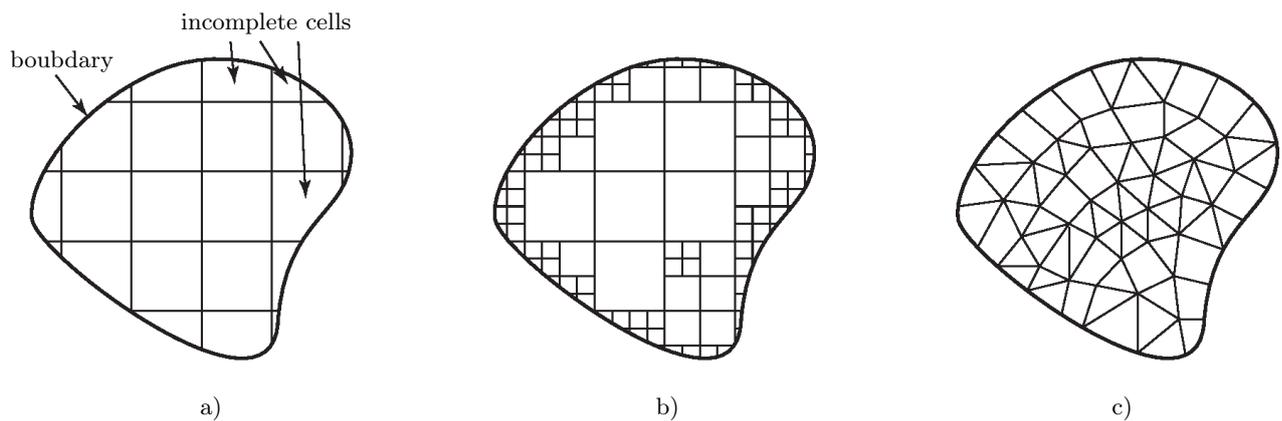


Figure 1. Various types of computational meshes: a) rectangular (with incomplete cells along the boundary);
 b) octree (with possibility of adaptation to the boundary and solution peculiarities);
 c) unstructured (polyhedral complex).

Initially, we have chosen unstructured meshes (Figure 1 c), because this is a very flexible technique which allows to easy adapt the mesh to the geometry of the computational domain and to the solution. More simple mesh techniques, for example, presented at Figure 1 a, can be considered as particular cases of unstructured meshes. The technique does not include the case shown at Figure 1 b. The restriction related to the fact that mesh elements can only intersect by their common edge (mesh *conformity*) may be important for some approximations of the equations, e.g. for finite-element ones. The development of data structures for support of the case Figure 1 b requires special approaches to the problems of approximation of equations and parallel computations organization, [29].

Theoretically, an unstructured mesh can be defined as a *polyhedral complex*, which is a set of the mesh elements of different dimensionalities (i. e. nodes, edges, faces and cells), where the intersection of each two mesh elements (unless it is empty) is the element of the complex. The description of the complex may be constructed with help of the incidence relations (two mesh elements are incident to each other if one of them is a bound of another). For a three-dimensional mesh, there are 12 incidence relations between the elements of different dimensions, but they are not all independent: some of them can be expressed through others. If we denote by In_{nm} the incidence relation between the mesh elements of dimensions n and m (for example, $In_{30}(c)$

gives the set of nodes which are the vertices of the cell c), we may write

$$\text{In}_{nm} = \text{In}_{lm} \circ \text{In}_{nl}$$

for any l such that $n \leq l \leq m$ and

$$\text{In}_{nm} = \text{In}_{mn}^{-1}.$$

Thus, to process an unstructured mesh in a code, one should store a sufficient set of incidence relations (and the nodes coordinates). In this case, for a given mesh element all its neighboring elements can be found.

To make a work with the unstructured mesh more convenient, one should consider the incidence relations between two adjacent dimensionalities as the *oriented* relations. We suppose that each edge has two extremities (i. e. incident nodes) and one of them is interpreted as the beginning, another one is the end. In this case, the pair (edge, node) enters in the incidence relation In_{10} or In_{01} with additional attribute (sign) ‘+’ for the beginning and ‘-’ for the end. The pair (face, edge) enters in the relation In_{21} or In_{12} with the sign depending on whether the natural direction of the edge (starting from its beginning to the end) coincides with the direction of tracing of the contour of the face. The pair (cell, face) enters in In_{32} and In_{23} relations with the sign depending on whether the cell is adjacent to the face from the side, which corresponds to the right-hand screw rule direction with the direction of the tracing of the contour of the face.

When a mesh is used for approximation and solution of some problem, it is also necessary to store the physical values associated with the mesh elements (nodes or cells depending on the chosen discretization). In principle, one can imagine three different ways to maintain the correspondence between the mesh elements and the values (attributes):

- the attributes are stored inside the structure (for example, C++ class), which describes the mesh element. The number and types of the physical attributes depend on the system of equations and on the numerical algorithm. Therefore, the structure for storing a mesh element becomes dependent on solver. It is impossible to separate the geometrical data from physical ones.
- a modification of the first point, where now a special container is inside the mesh element structure, which can contain an arbitrary number of attributes of various types. The principal drawback is that the access to these attributes requires the procedure of search in this container any time when an attribute associated with a given mesh element is needed. Meanwhile, access to the attributes is a critical operation.
- the attributes are stored separately from the mesh, for instance, in an array of attributes (presuming that the mesh elements are numbered). This is the most common way of storing the attributes, but it has a fundamental disadvantage as well: the correspondence between a mesh element and the attribute is not established firmly and is based on the assumption that the numbers of the mesh elements are constant.

The incidence relations can be also considered as the attributes of the mesh elements (more precisely, the list of incident elements constitutes an attribute of a mesh element) and it is natural to apply the same approaches.

In the code MARPLE, the latter approach to storing the attributes (and binary relations) was chosen. However, since the mesh structures were developed for the dynamically changing meshes (which may be used both in solvers with dynamical mesh adaptation and in the codes for mesh generation and editing), there were established special C++ classes for the arrays of the attributes. In addition, the concept of the *numbered set* (implemented as the class “setnumbers”, see Figure 4) was naturally arisen, for instance, the set of mesh elements of one dimension is the numbered set. Each array of attributes is linked to a numbered set, which stores the list of attribute arrays connected to this numbered set. If some changes occur in a numbered set, for example, some mesh elements are deleted and some other elements change their numbers for keeping the continuous numeration, the corresponding changes in all arrays of attributes are performed automatically. Indeed, the size of array is changed and the attributes which correspond to the elements with varying numbers are moved to the new positions in the array.

In this case, the binary relations are naturally considered as the relations between two numbered sets. Special data structures for storing binary relations, like in the case of an array of attributes, automatically perform all necessary changes when one of the numbered sets changes. In fact, one may consider the binary relation as an array of the lists of the corresponding elements. It is convenient to store the binary relations as arrays of lists.

To represent an unstructured mesh (a polyhedral complex, class “OrGcomplex0<3>”, see Figure 4), we should store the array of nodes coordinates and some of the incidence relations. From the user’s point of view,

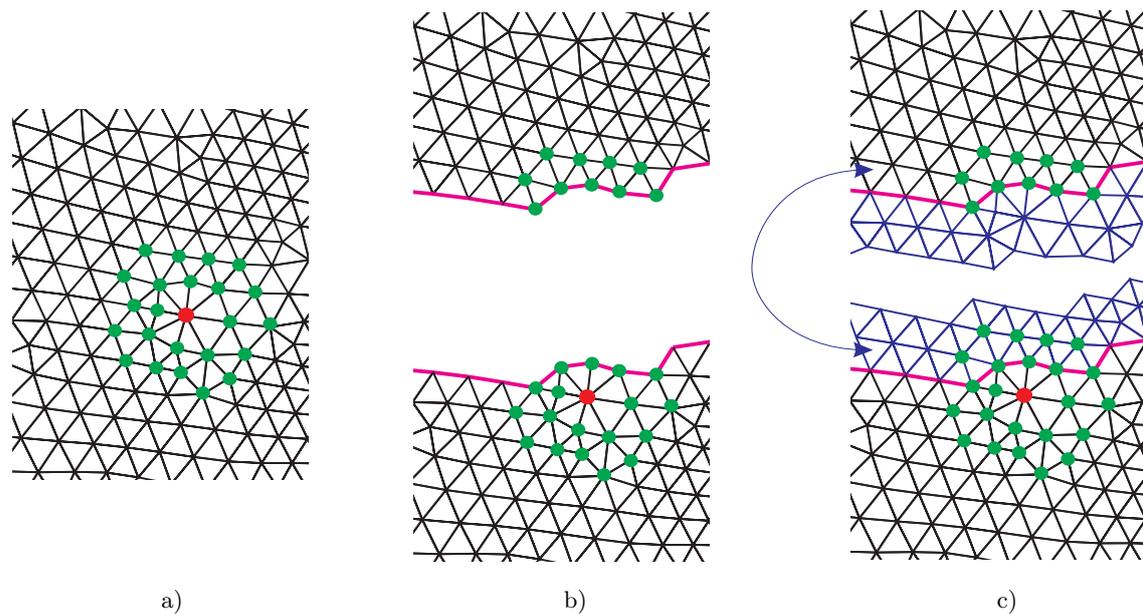


Figure 2. Fragment of the unstructured mesh and approximation stencil of a node (a), the same mesh fragment after mesh splitting between the processes (b), the fictive mesh elements are added and the stencils of all real nodes are completed (c).

it is not important what incidence relations are stored. Thus, the user interface to the unstructured grid implies the following request operations: for a given element (or a group of elements) of dimension n it is necessary to get the incident elements of dimension m . It does not matter whether the incidence relation In_{nm} is stored, the set of elements can be computed via other incidence relations.

All modern solvers, especially 3D ones, treat the meshes and the mesh data in a distributed manner. In this regard, the mesh is subdivided into a number of parts which are stored and handled by different processors (or processor cores, command streams etc.). It is not even supposed that the whole mesh can be fit into the memory of one processor. In the process of solution of a problem, some inter-processor data exchange is required to treat the data on the mesh elements that are close to the bounds of the parts (MPI programming model). The most common technique is based on the usage of the fictive mesh elements (also called “ghost” or “halo” mesh elements), Figure 2. A fictive mesh element represents a real element contained in another part of the mesh. The physical data corresponding to such an element can be obtained by the interprocessor communications, and then the computations in the mesh elements near the decomposition boundaries can be performed regularly, like in all other elements. The ghost elements technique can be also used for the periodical problems. If the computational domain and the problem in a whole is invariant with respect to some space transformation (usually translation, rotation or plane symmetry), then it allows to solve the problem only in a part of the computational domain, for instance, in one period. In this case, the fictive mesh elements are the images of some real elements in the corresponding symmetry transformation, see Figure 3.

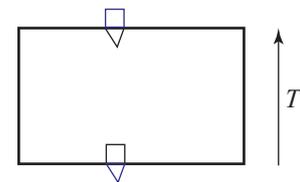


Figure 3. Example of usage of the fictive mesh elements for a periodical problem. The studied problem is invariant with respect to the translation T , and the lower and upper boundaries should be considered as a periodical ones.

Thus, for processing the distributed meshes with periodical boundaries, the mesh data structures should contain the descriptors for the fictive mesh elements and the tools for establishing the correspondence between each fictive element and its real preimage. Such a distributed mesh is represented by the class “Mesh3D”, Figure 4.

5. Software design.

5.1. Architecture overview. In accordance with the ideas discussed above and principles of object-oriented design, the code was divided into modules: approximation, work with grids, solvers, boundary conditions and equations of state. The hierarchy of classes in modules and the relationships between them are presented in Figure 4 (there are shown the essential classes and relationships only, modules are highlighted in frames).

- MARPLE_Driver is the main module that controls the entire computing process, sends messages to other modules and does not perform any actions that require computational resources.
- Main_Solver carries out top-level operations with physical subdomains (controlled by MARPLE_Driver).
- PhysicalDomainSolver is a physical solver that performs operations at the physical subdomains level and is controlled by MainSolver. Each PhysicalDomainSolver object fulfills operations on one computing node only and in one physical subdomain only. The physical subdomain is assigned to the specific PhysicalDomainSolver object respectively.
- ElementalSolver conducts calculations to solve a concrete subsystem of equations or just one equation in the corresponding physical subdomain. Many ElementalSolver objects provide a total approximation of all equations that should be solved. For instance, if the equations of heat conduction, Joule heating and radiation transfer are to be solved in a specific physical subdomain, then one elementary solver can be used to solve the heat conduction equation numerically, and the other one can solve the remaining two equations together. In this example, two elementary solvers provide a total approximation of all three equations defined in a particular physical subdomain. Each ElementalSolver object performs operations on one computational node only and in one physical subdomain only. In addition, it solves one specific equation or a subset of the system of equations. Elementary solvers are completely controlled by the solvers of the physical subdomains described above.
- Properties of matter MatterProperties provide the unified access to equations of state for each physical subdomain.

The data structures used in the code MARPLE are original, but they support functionality similar to those utilized in other continuous-media codes like OpenFOAM, FlowVision, Ansys Fluent etc.

5.2. Parallel performance. Implementation of a multiphysics model entails calculating dozens of physical quantities in each cell of the computational mesh at each time step. Simulation of inherently three-dimensional processes accounting the actual geometry of devices and/or multiscale processes requires meshes containing from millions to hundreds of millions of cells. Solution of such tasks is possible only when using the distributed computations based on modern high-performance computers. The code MARPLE can be run on computers with shared or distributed memory as a parallel MPI application relying on domain and mesh decomposition.

5.3. Third party software and open-source tools. We employ an open-source CAD-CAE platform SALOME [30] to preprocess computational domains of any complex shape for the numerical simulations, including geometry description, boundary and subregions attributes settings, mesh generation and refinement.

For data visualization, an open-source multi-platform application ParaView [31] is used.

To solve large systems of linear algebra equations with sparse matrices arising in implicit approximations of dissipative processes equations, we use Aztec [32, 33]. This is a massively parallel library for solving sparse linear systems. It includes a set of iterative solvers, preconditioners and matrix-vector multiplication routines. Aztec is a robust tool, which has demonstrated high scalability up to 10^9 equations. It can operate with matrices that have a completely unstructured sparsity pattern. This fact is crucial in the case of unstructured meshes. One of the external libraries, used in Aztec, is LAPACK [34]. Routines of this library call to another library, namely the Basic Linear Algebra Subprograms (BLAS). Aztec allows the incorporation of user-supplied matrix-vector product routines, including those for GPGPU.

For the decomposition of meshes in parallel computations we use open-source libraries METIS/ParMETIS [35]. METIS libraries carry out graph decomposition with good load balance, which may be critical in the case of large unstructured meshes. The library is written in ANSI C and has no problem in linking with our C++ codes.

CMake [36] was chosen as the standard MARPLE project build system. Thus, the project developers and users can employ their custom development tools (development environment, compilers, etc.) with project files



Figure 4. UML class diagram: architecture of MARPLE.

independently on each other. The target platforms are independent on the development platforms. CMake-format project files are just a simple format plain text files. It is a suitable tool for maintaining a large tree structure code.

Open-source software is a fundamental element of our computing technology. The open libraries are incorporated in the code and free tools are used for organizing computational experiments: cross-platform assembly, pre- and postprocessing. The adoption of open-source software facilitates the development and debugging of the code, helps to provide the portability and standardization of created codes, allows coupling our codes with the codes of other developers.

6. Examples of applications.

6.1. Modelling the dynamics of supernovae remnants. The explosions of supernovae are significant events in the life of any galaxy. Over the course of a few dozen days — a short time on astronomical scales — their luminosity becomes comparable to the one of a galaxy. This property enables to use them in studying the properties of distant regions of the Universe. Furthermore, these objects are incredibly rich in various types of physics. During the explosion, the parameters of the environment undergo changes in a wide range, from superdense matter to low densities of interstellar medium, from temperatures of thermonuclear fusion (from hundreds of keV to MeV) to several thousand kelvins and below. This diversity renders supernovae natural laboratories in which numerous physical processes can be observed. An example of 3D modelling of the supernova explosion caused by the so-called pair instability is considered in [37] by means of MARPLE.

The two-fluid model [9, 10] implemented in the code is utilized to solve the problem of supernova remnant evolution. At this stage, the specific details of the supernova explosion mechanism are not essential, as the focus is on the dynamics at spatial scales significantly exceeding the initial size of the star. The magnitude of the total energy release E_{expl} plays a substantial role. In the calculations, this energy is released as internal energy of the matter in a small volume at the center of the computational domain with characteristic dimensions of L . Initially, the region is filled with a homogeneous medium consisting of ions with density of ρ_{0i} and neutrals with density of ρ_{0n} , which are of the order of the interstellar medium density (approximately $1 m_p/\text{cm}^3$). The calculations were performed for different ratio ρ_{0i}/ρ_{0n} , while the total density of the medium remains constant $\rho_{0i} + \rho_{0n} = \text{const}$.

At the initial moment, the environment is filled with homogeneous ionic and neutral components. Figure 5 presents the results of computations when the proportions of the initial densities of the components were varied

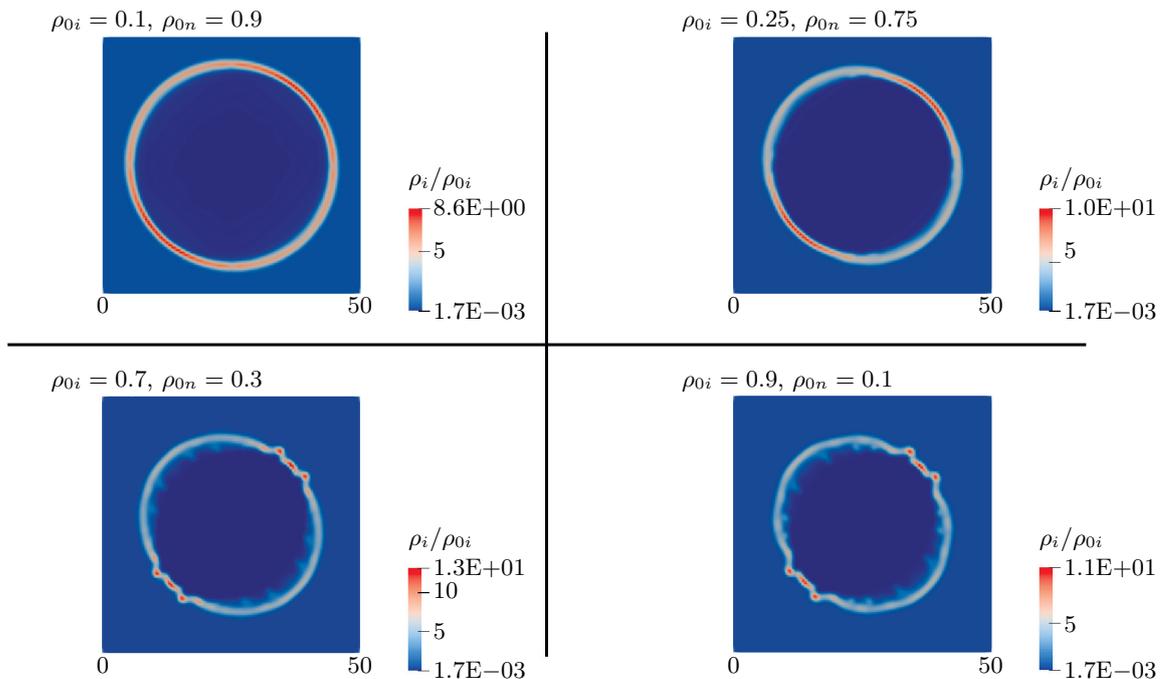


Figure 5. The ion density normalized to the initial value ρ_i / ρ_{0i} for calculations with different ratios of ion and neutral densities at the time of $t = 400$ thousand years.



leaving the relationship $\rho_i + \rho_n = 1 \times m_p/\text{cm}^3$ unchanged. The energy release $E_i = 1.28 \times 10^{51}$ erg in the ionic component occurs instantaneously at the center of the region with a size of $R = 4$ pc. The temperature of the surrounding medium outside the energy release region for ions and in the total region for neutrals is $T_i = T_n = 8000$ K. The magnitude of the magnetic field was set to $B = 5 \mu\text{G}$. To mitigate the numerical effects of the influence of preferred directions along the axes, the magnetic field was assumed to be oriented along the line $X = Y$.

When the number of neutrals is less than the number of ions, they do not affect the instability. The opposite situation is observed when neutrals dominate in density.

6.2. Research on the physics of current-carrying plasma (Z-pinch). The physics of electrical discharge plasma involves nonlinear wave processes, kinetic, thermal and hydrodynamic instabilities. Plasma structures vary greatly in scale and the full-scale computational experiments with plasma dynamic models require petascale and higher performance supercomputers. Here we present some results of the modeling of radiation-plasma processes accompanying experiments on the high-current generator Angara-5-1 (Troitsk Institute for Innovation and Fusion Research, Troitsk, Moscow) aimed at creating new designs of high-brightness ultraviolet and X-ray radiation sources.

In recent years, a promising pinch configurations were studied at Angara-5-1 facility providing higher energy concentration related to the possibility of 3D compression of $\sim 10 \mu\text{m}$ wire arrays [38]. Multiwire array implosion was simulated in a series of numerical experiments with 3D plasma flow models. The aim was to explore plasma instabilities arising at the end of plasma ablation and developing up to the final stage of the wire array implosion. Different configurations of wire arrays were investigated. There were studied cylindrical and quasispherical wire arrays with single-array and nested-array design. Plasma emission was reproduced using a prolonged plasma ablation model including spatial nonuniformity of plasma production rate consistent with experimental X-ray images of lower plasma emission areas [39]. The distinction of wires in nested arrays was implemented. Figure 6 shows an example of the experimental setup that was reproduced in numerical experiments.

The geometry of the wire array and the computational mesh (~ 4 millions cells) were constructed by SALOME. Domain decomposition for 256 computing cores was carried out using open libraries ParMETIS.

To estimate the energy balance, one can use the one-group (“gray”) approximation for radiative heat transfer. To calculate the time shape and power of the X-ray pulse of nested arrays, it is advisable to utilize from 8 to 20 spectral groups.

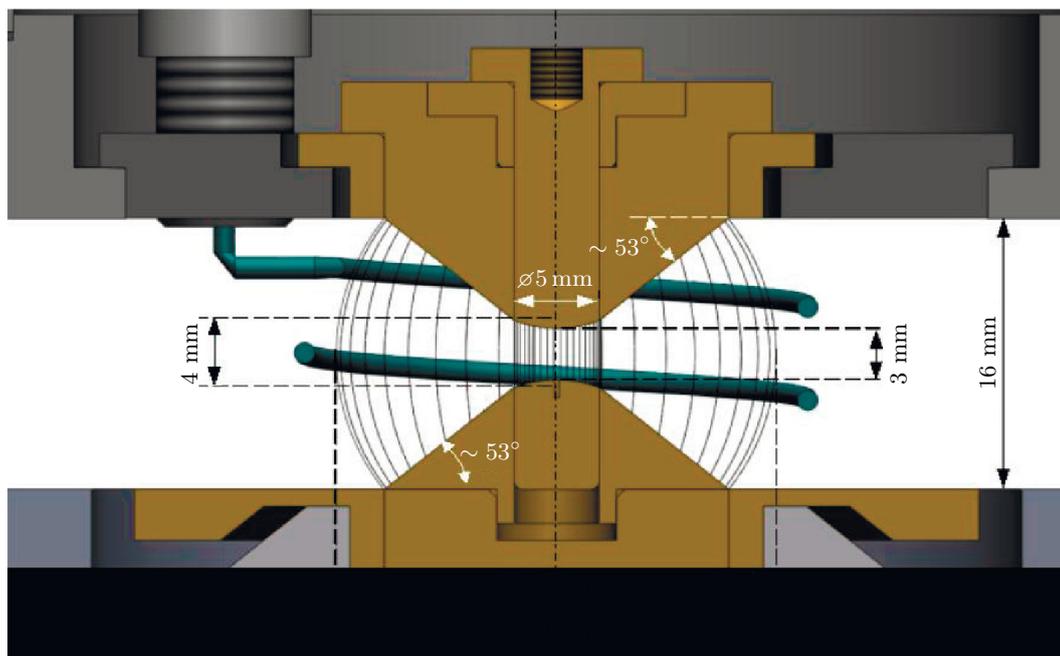


Figure 6. Design of an experimental quasi-spherical nested assembly (outer cascade — 40 tungsten wires, inner cascade — 80 tungsten wires).

In a series of the compression of multiwire arrays of the mixed composition modeling, the “gray” approximation was used for the emission of polypropylene plasma (averaging over the spectrum $0.1 \text{ eV} < h\nu < 50 \text{ keV}$). For the emission of tungsten, 8 spectral groups were applied adapted to the spectrum of the tungsten assembly recorded in the experiments. Group averaging and the construction of the corresponding tables were performed using the THERMOS program [16].

Up to 2000 spectral groups can be utilized for post-processing and building detailed emission spectra.

It was demonstrated numerically that a special design of the electrodes, the wire array and the mass distribution along the wires result in very compact spherical bright radiation source in the center of the array. Linear mass distribution $m(\theta) \sim 1/\sin^2 \theta$ (where θ is the poloidal angle) along the length of the wires in a quasi-spherical array provides the increase in the radiation power density by a factor of ~ 3 in comparison with quasi-spherical compression without mass profiling and by almost 7.5 times compared to the cylindrical array. Figure 7 demonstrates the details of plasma flow derived in computations. The presented picture was created by ParaView.

The qualitative and quantitative agreement of numerical results with the experimental data obtained at Angara facility is discussed in [38]. Thus, the results of modeling with the usage of supercomputers are of fundamental importance for understanding the physics of nonlinear processes in pinch plasma.

6.3. Modelling of the capillary discharge for laser acceleration of particles. Capillary discharges are widely applied in many experiments as a simple and convenient tool for generating “quiet” plasma with reliably controlled parameters. Compact laser-plasma particle accelerators are considered as one of the most promising applications of capillary plasma.

The capillary discharge is used to create a plasma waveguide for an accelerating laser pulse. The electrons are accelerated under the action of femtosecond laser pulses passed through the discharge plasma. Currently, the frequency modes of operation of electron accelerators based on capillary discharges are being actively studied.

An example is the results of a three-dimensional magnetohydrodynamic simulation of a capillary discharge cycle, which includes the stages of filling the capillary with a working gas (hydrogen), forming a plasma channel, and restoring the working medium before the start of the next discharge. The simulation was performed for a capillary with a circular cross section of the diameter of $300 \mu\text{m}$, connected to the gas supply channels. The Figure 8 demonstrates a part of the computational mesh and the velocity distribution near the open end of the capillary. This velocity field is reached at the end of the initial filling stage.

The simulation was carried out mainly to obtain such a characteristic, which affects the efficiency of the accelerator as a whole, as the characteristic time required to obtain steady flows at individual stages of the accelerator system operation:

- the stage of filling the capillary with gas from the supply channels until the steady state establishment;
- the stage of electric discharge in the capillary and the return of the parameters to the level of the parameters at the filling stage.

In the discussed problem, the free parameters are: the length of the capillary, its diameter, the shape of its end, the pressure of the gas conveyed through the supply channels. It was assumed that the capillary was 2 cm long, $300 \mu\text{m}$ in diameter, had a round cross-section and the inlet pressure was 125 mbar at the temperature of 293 K. Varying the free parameters, there was made an attempt to provide the symmetry of the plasma channel along the axis of the capillary and to save its profile (with decreasing density on the axis). Saving the profile

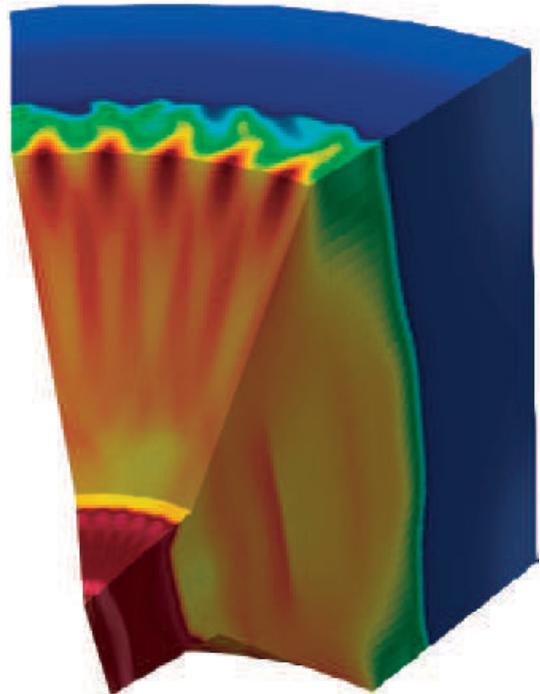


Figure 7. Computed plasma density distribution inside a nested array (1/8 of the array in Figure 6 is shown).

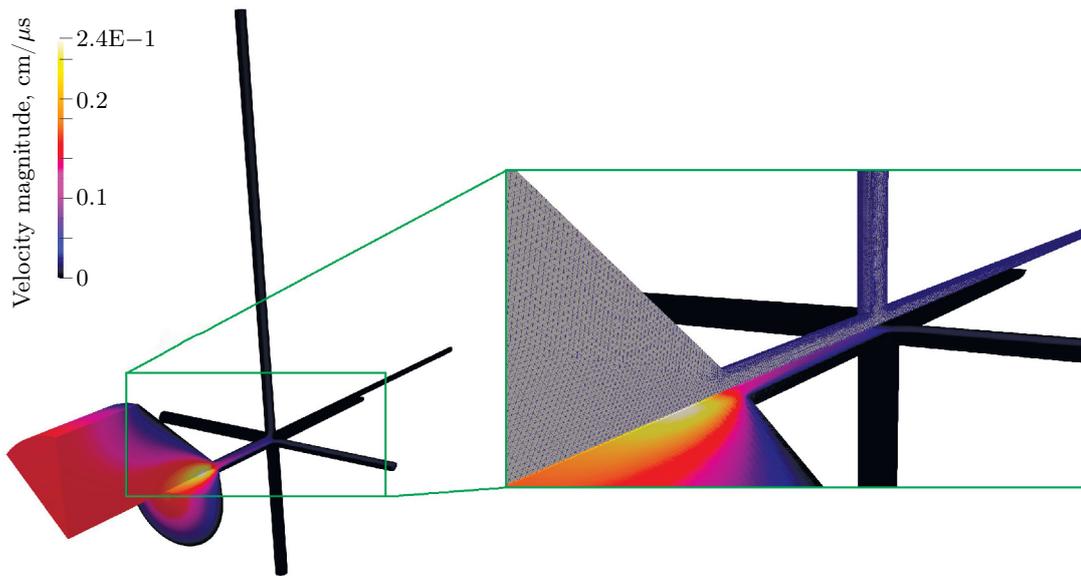


Figure 8. Velocity field at the end of the initial filling stage.

as far as possible is necessary to focus the laser beam. The opportunity of symmetrization of the channel using four inlet channels located crosswise is shown. The possibility of using a capillary discharge with a frequency of about 1 kHz is demonstrated as well.

Similar experiments are conducted in:

- in the framework of the project BELLA (Berkeley Laboratory Laser Accelerator) (LBLN, USA);
- at ELI Beamlines (Czech Republic);
- at Laboratori Nazionali di Frascati (INFN) (Italy);
- at Physics Department, Saint Petersburg State University (Russian Federation);
- at Budker Institute of Nuclear Physics of Siberian Branch Russian Academy of Sciences (BINP SB RAS) (Novosibirsk, Russian Federation);
- and etc.

6.4. Modelling of the laser pulse interaction with the solid-state target. The experiment was performed within the TurboHEDP project [40]. The interaction of the short laser pulse with the solid-state two-component target was investigated. The experimental installation was described in details in [40]: the laser pulse (with wavelength $\lambda = 1.315 \cdot 10^{-4}$ cm) is directed at the 45° angle to the target, which consists of the

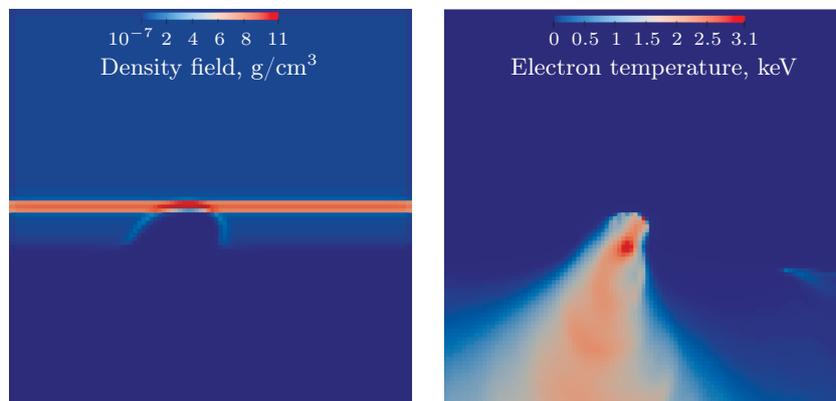


Figure 9. Spatial distribution of the density (g/cm^3) of the evaporated target material and its electron component temperature (keV) at the moment of the laser pulse termination.

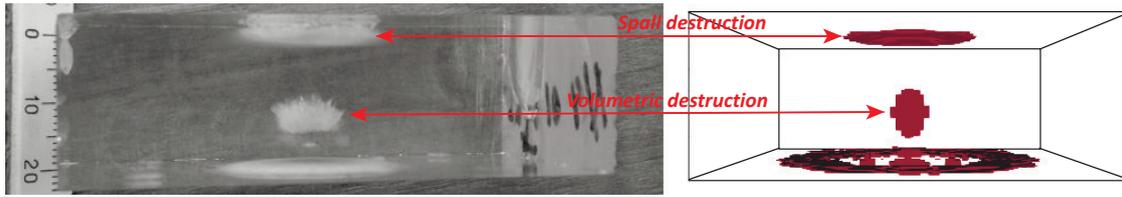


Figure 10. Numerical results and photo of the sample.

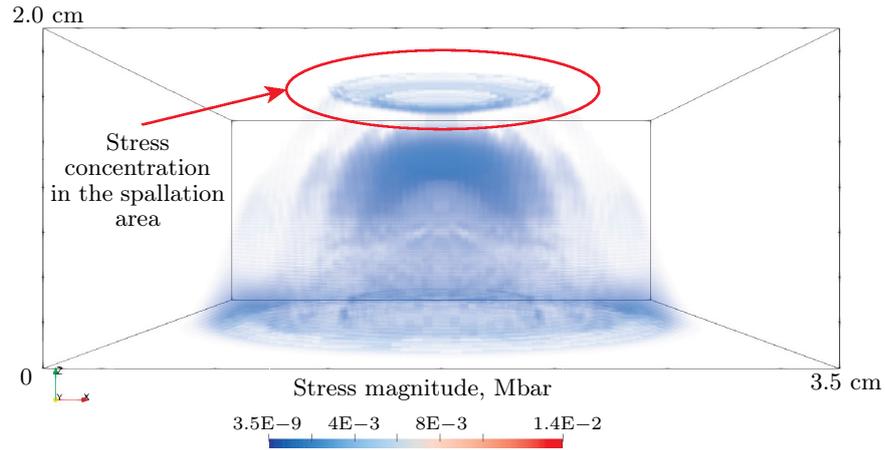


Figure 11. Tensor stress field in Mbar in a fractured sample at a time of 5.2 μ s.

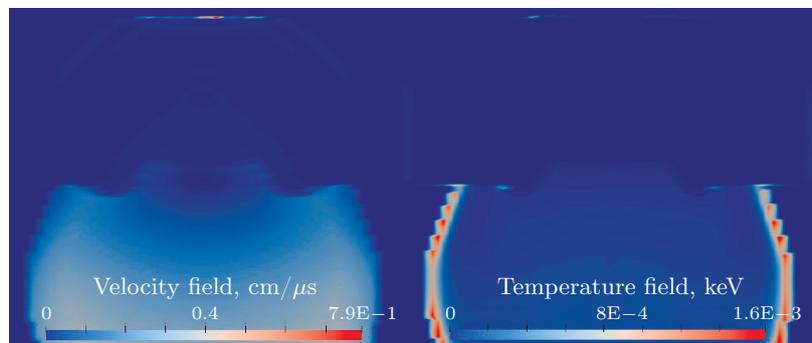


Figure 12. Wave process in the entire computational domain, including the destroyed sample. Left picture is the velocity distribution in $\text{cm}/\mu\text{s}$, right picture is the temperature field in keV.

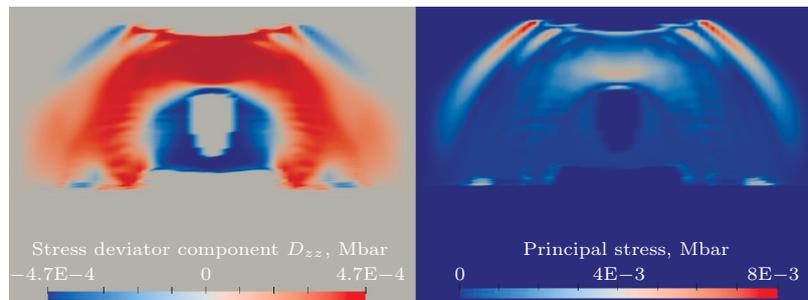


Figure 13. Wave process in the entire computational domain, including the destroyed sample. Left picture is the distribution of the σ_{zz} component of the stress deviator in Mbar, right picture is the vector field of the principal stresses in Mbar.



10 μm polymer layer, the 1 μm copper layer and a thick polymer layer at the target's rear side. The pulse duration is $1.8 \cdot 10^{-10}$ s, the laser intensity at the focal spot reaches the value of $3 \cdot 10^7$ W/cm².

The numerical modelling was carried out with help of the MARPLE code using the ray tracing model. To reproduce the geometrical shape of the laser beam, 961 rays were taken. The Figure 9 represents the computed spatial distributions of the density of evaporized target substance and its electron component temperature at the moment of the laser pulse termination.

6.5. Effects of the strong energy fluxes on condensed matter. Nowadays, there are no experimental methods that could be used to conduct direct measurements of the parameters of the material state directly in the zone of the volumetric and spall fracture. Therefore, for its description a joint analysis of the results of a physical and numerical experiment is now widely applied.

The epoxy resin sample is a cylinder with a diameter of 35 mm and a height of 20 mm. It is exposed to a relativistic electron beam. The photo of the sample after this experiment is presented at Figure 10. The results of the numerical modelling of this process given on the right part of the Figure 10 show a good reproduction of the destruction zones. Numerical distributions of some important physical quantities in the studied process are illustrated at Figures 11–13. The recoil impulse of the flying vaporized material was directly taken into account in the end-to-end method in the process of calculating the stress-strain state in the non-vaporized remnant of the sample [15]. To verify the computation, theoretical estimates and experimental data obtained at the Kurchatov Institute Research Center on a high-current electronic accelerator “Kalmar” were used.

7. Conclusion. The code MARPLE, which was originally created for the modelling of radiative plasma processes, has become a universal tool for solving a wide range of continuum mechanics problems relating to different object domains. The code demonstrates the ability to incorporate new physical models due to its flexible module structure. The use of unstructured computational mesh technique facilitates the solution of problems involving the geometrically complex computational domains. The technique of the mesh decomposition makes the code applicable to solution of computationally expensive problems on supercomputers, while the problems with moderate computational power consumption can be solved with use of an ordinary computer.

References

1. V. A. Gasilov, A. S. Boldarev, S. V. D'yachenko, et al., “Program Package MARPLE3D for Simulation of Pulsed Magnetically Driven Plasma Using High Performance Computing,” *Matem. Mod.* **24** (1), 55–87 (2012). <http://mi.mathnet.ru/mm3198>. Cited September 1, 2023.
2. A. S. Boldarev, V. A. Gasilov, O. G. Olkhovskaya, et al., “Object-Oriented Code MARPLE3D: Simulations of Radiative Hydrodynamic/MHD Effects at High-Performance Computer Systems,” in *Proc. 6th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2012, Vienna, Austria, September 10–14, 2012*. https://www.researchgate.net/profile/Elizaveta-Dorofeeva/publication/283302809_Object-oriented_code_MARPLE3D_Simulations_of_radiative_hydrodynamicMHD_effects_at_high-performance_computer_systems/links/56321a3a08ae13bc6c373070/Object-oriented-code-MARPLE3D-Simulations-of-radiative-hydrodynamic-MHD-effects-at-high-performance-computer-systems.pdf. Cited September 7, 2023.
3. S. I. Braginsky, “Transport Phenomena in Plasma,” in *Problems of Plasma Theory*, Ed. by M. A. Leontovich (Gosatomizdat, Moscow, 1963), Vol. 1, pp. 183–272 [in Russian].
4. V. B. Baranov and K. V. Krasnobaev, *Hydrodynamic Theory of Space Plasma* (Nauka, Moscow, 1977) [in Russian].
5. N. A. Bobrova and P. V. Satorov, “MHD-Equations for Completely Ionized Plasma of Complex Composition,” *Fiz. Plazmy* **19** (6), 789–795 (1993).
6. B. N. Chetverushkin, *Mathematical Modeling of Problems in the Dynamics of a Radiating Gas* (Nauka, Moscow, 1985) [in Russian].
7. A. F. Nikiforov, V. G. Novikov, and V. B. Uvarov, *Quantum-Statistical Models of High-Temperature Plasma and Methods for Computing Rosseland Mean Free Paths and Equations of State* (Fizmatlit, Moscow, 2000) [in Russian].
8. Ya. B. Zel'dovich and Yu. P. Raizer, *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena* (Academic Press, New York, 1967). doi 10.1016/B978-0-12-395672-9.X5001-2.
9. Yu. S. Sharova, S. I. Glazyrin, and V. A. Gasilov, “Study of the Influence of the Background Neutral Component on the Dynamics of the Envelope in Supernova Remnants,” *Astron. Lett.* **47** (11), 746–753 (2021). doi 10.1134/S1063773721110050.

10. Yu. S. Sharova, “MHD Simulation of the Supernova Remnant Dynamics Taking into Account the Neutral Plasma Component,” *Mat. Model.* **34** (1), 47–58 (2022) [*Math. Models Comput. Simul.* **14** (4), 654–661 (2022)]. doi [10.1134/S2070048222040111](https://doi.org/10.1134/S2070048222040111).
11. D. R. Schultz, P. S. Krstic, T. G. Lee, and J. C. Raymond, “Momentum Transfer and Viscosity from Proton-Hydrogen Collisions Relevant to Shocks and Other Astrophysical Environments,” *Astrophys. J.* **678** (2), 950–960 (2008). doi [10.1086/533579](https://doi.org/10.1086/533579).
12. Y. S. Sharova and D. S. Shidlovski, “Numerical Study of Radiatively Cooling Partially Ionized Plasma Expansion in Neutral Environment,” in *Proc. 8th Int. Congress on Energy Fluxes and Radiation Effects, Tomsk, Russia, October 2–8, 2022* (TPU Publishing House, Tomsk, 2022), pp. 206–212. doi [10.56761/EFRE2022.S2-P-033502](https://doi.org/10.56761/EFRE2022.S2-P-033502).
13. J. D. Huba, *NRL Plasma Formulary* (Naval Research Laboratory, Washington D.C., 2013). https://library.psfc.mit.edu/catalog/online_pubs/NRL_FORMULARY_13.pdf. Cited September 2, 2013.
14. A. S. Boldarev, O. G. Olkhovskaya, V. V. Val’ko, and V. S. Solovyova, *Thermodynamic Models of Gas Mixtures for Computational Fluid Dynamics*, Preprint No. 54 (Keldysh Institute of Applied Mathematics, Moscow, 2021) [in Russian]. <https://library.keldysh.ru/preprint.asp?id=2021-54>. Cited September 2, 2023.
15. D. S. Boykov, O. G. Olkhovskaya, and V. A. Gasilov, “Coupled Simulation of Gasdynamic and Elastoplastic Phenomena in a Material under the Action of an Intensive Energy Flux,” *Mat. Model.* **33** (12), 82–102 (2021) [*Math. Models Comput. Simul.* **14** (4), 599–612 (2022)]. doi [10.1134/S2070048222040044](https://doi.org/10.1134/S2070048222040044).
16. THERMOS: Software Package & Database. <https://www.keldysh.ru/cgi/thermos/navigation.pl?en,home>. Cited September 2, 2023.
17. P. V. Sasorov, N. A. Bobrova, and O. G. Olkhovskaya, *The Two-Temperature Equations of Magnetic Hydrodynamics of the Plasma*, Preprint No. 21 (Keldysh Institute of Applied Mathematics, Moscow, 2015) [in Russian]. <https://library.keldysh.ru/preprint.asp?id=2015-21>. Cited September 2, 2023.
18. V. V. Valko, V. A. Gasilov, N. O. Savenko, and V. S. Solovyova, “Simulation of an Air Shock Wave Using the Equations of State for the Jones–Wilkins–Lee Detonation Products,” *Mat. Model.* **34** (4), 3–22 (2022) [*Math. Models Comput. Simul.* **14** (6), 875–888 (2022)]. doi [10.1134/S2070048222060163](https://doi.org/10.1134/S2070048222060163).
19. T. J. Barth and D. C. Jespersen, *The Design and Application of Upwind Schemes on Unstructured Meshes*, AIAA Paper No. 89-0366 (1989).
20. P. Batten, N. Clarke, C. Lambert, and D. M. Causon, “On the Choice of Wavespeeds for the HLLC Riemann Solver,” *SIAM J. Sci. Comput.* **18** (6), 1553–1570 (1997). doi [10.1137/S1064827593260140](https://doi.org/10.1137/S1064827593260140).
21. T. Miyoshi and K. Kusano, “A Multi-State HLL Approximate Riemann Solver for Ideal Magnetohydrodynamics,” *J. Comput. Phys.* **208** (1), 315–344 (2005). doi [10.1016/j.jcp.2005.02.017](https://doi.org/10.1016/j.jcp.2005.02.017).
22. P. L. Roe, “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *J. Comput. Phys.* **135** (2), 250–258 (1997). doi [10.1006/jcph.1997.5705](https://doi.org/10.1006/jcph.1997.5705).
23. O. G. Olkhovskaya, *Grid-Projection Schemes for Approximation of the Second Order Partial Differential Equations on Irregular Computational Meshes*, Preprint No. 226 (Keldysh Institute of Applied Mathematics, Moscow, 2018) [in Russian]. <https://library.keldysh.ru/preprint.asp?lg=e&id=2018-226>. Cited September 2, 2023.
24. O. Olkhovskaya, A. Kotelnikov, M. Yakobovsky, and V. Gasilov, “Parallel Ray Tracing Algorithm for Numerical Analysis in Radiative Media Physics,” in *Advances in Parallel Computing* (IOS Press, Amsterdam, 2021), Vol. 32, pp. 137–146. doi [10.3233/978-1-61499-843-3-137](https://doi.org/10.3233/978-1-61499-843-3-137).
25. A. Kotelnikov, I. Tsygvintsev, M. Yakobovsky, and V. Gasilov, “Parallel Ray Tracing Algorithm for Numerical Analysis of Laser Radiation Absorption in a Plasma,” in *Communications in Computer and Information Science* (Springer, Cham, 2019), Vol. 1129, pp. 110–120. doi [10.1007/978-3-030-36592-9_10](https://doi.org/10.1007/978-3-030-36592-9_10).
26. R. N. Shakirov, “C++ Class for Integers of Unlimited Range,” <http://www.imach.uran.ru/cbignum/index.html>. Cited September 3, 2023.
27. P. J. Frey and P.-L. George, *Mesh Generation: Application to Finite Elements* (ISTE Ltd and John Wiley & Sons, 2008).
28. V. A. Gasilov, E. L. Kartasheva, and O. V. Fryazinov, “Discretization of Inhomogeneous Geometric Objects,” in *Encyclopedia of Low-Temperature Plasmas* (Yanus-K, Moscow, 2009), Vol. VIII-1, pp. 298–318.
29. D. Boykov, S. Grigoriev, O. Olkhovskaya, and A. Boldarev, “Implementing a Mesh-Projection Schemes Using the Technology of Adaptive Mesh Refinement,” in *Lecture Notes in Computer Science* (Springer, Cham, 2020), Vol. 11958, pp. 576–583. doi [10.1007/978-3-030-41032-2_66](https://doi.org/10.1007/978-3-030-41032-2_66).
30. Salome Platform — The Open-Source Platform for Numerical Simulation. <http://www.salome-platform.org/>. Cited September 3, 2023.



31. ParaView — Open-Source, Multi-Platform Data Analysis and Visualization Application. <https://www.paraview.org/>. Cited September 3, 2023.
32. AztecOO, Trilinos. <https://trilinos.github.io/aztecoo.html>. Cited September 3, 2023.
33. M. A. Heroux, *AztecOO User Guide* (Sandia National Laboratories, Albuquerque, 2007). <https://trilinos.github.io/pdfs/AztecOOUserGuide.pdf>. Cited September 3, 2023.
34. LAPACK — Linear Algebra PACKage. <http://www.netlib.org/lapack>. Cited June 13, 2023.
35. Karypis Lab@UMN-CS&E. <https://github.com/KarypisLab>. Cited September 3, 2023.
36. CMake. <http://www.cmake.org/>. Cited September 3, 2023.
37. N. Smirnova, *Simulation multidimensionnelle de l'explosion d'une supernova par production de paires*, Thèse de doctorat en Physique théorique (Université Grenoble Alpes, France, 2019). <https://theses.fr/2019GREAY089>. Cited September 3, 2023.
38. K. N. Mitrofanov, V. V. Aleksandrov, A. V. Branitski, et al., “Progress in Researching the Implosion of Nested Arrays of Mixed Composition and the Generation of Soft X-Ray Power Pulse,” *Plasma Phys. Control. Fusion* **64** (4), Article Identifier 045007 (2022). doi 10.1088/1361-6587/ac49f9.
39. V. V. Aleksandrov, A. V. Branitskii, G. S. Volkov, et al., “Dynamics of Heterogeneous Liners with Prolonged Plasma Creation,” *Plasma Phys. Rep.* **27** (2), 89–109 (2001). doi 10.1134/1.1348487.
40. A. Casner, G. Rigon, B. Albertazzi, et al., “Turbulent Hydrodynamics Experiments in High Energy Density Plasmas: Scientific Case and Preliminary Results of the TurboHEDP Project,” *High Power Laser Sci. Eng.* **6** (3), Article Number 03000e44 (2018). doi 10.1017/hpl.2018.34.

Table of symbols

\mathbf{A} — vector potential of magnetic field	n — concentration	η, μ — viscosity coefficients
A — atomic weight	P, p — pressure	$\dot{\lambda}$ — specific power of plastic deformation
\mathbf{B} — magnetic induction	\mathbf{q} — energy flux density vector	ρ — density
$c = 3 \cdot 10^{10}$ cm/s — light velocity in vacuum	Q_{ei} — electron-ion energy exchange	σ — electrical conduction; section of particles interaction (sect. 2.4.1); stress tensor (sect. 2.6)
$D_{\sigma ij}$ — stress deviator	Q_J — Joule heating	σ^c, τ^c — damage thresholds at normal and shear stressing
\mathbf{E} — electric field strength	Q_R — energy source caused by radiation absorption	σ_{SB} — Stefan—Boltzmann constant
E — total energy	$Q_{\alpha\beta}$ — collisional energy exchange between ions and neutrals	σ_T — yield stress
\mathbf{F} — external forces	$\mathbf{R}_{\alpha\beta}$ — friction force between ions and neutrals caused by collisions	Ψ — indicator of state of aggregation (phase state)
G — shear modulus	t — time	
\mathbf{j} — electric current density	T — temperature	
J — emittance	U — radiant energy density	
k, χ — thermal conductivity coefficient	U — radiant energy density	
k_B — Boltzmann constant	\mathbf{w} — velocity of medium	Indices:
k^P, k^R — absorption coefficients averaged according to Planck and Rosseland	\mathbf{W} — radiant energy flux	e, i — electron and ion component of plasma
K — bulk modulus	Z — average charge number of an ion	\parallel, \perp — direction parallel and perpendicular to magnetic field
L_S — sublimation heat	α — thermal expansion coefficient	α, β — ionized and neutral components of plasma in two-fluid model, sect. 2.4
m — mass (of a particle)	α_n — mass concentration of the n -th component of mixture, sect. 2.5	
	ε — specific internal energy	
	$\dot{\varepsilon}$ — strain velocity tensor	

Received
July 8, 2023

Accepted for publication
August 14, 2023

Information about the authors

Vladimir A. Gasilov — Dr. Sci., Prof., Head of Department; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Alexey S. Boldarev — Ph.D., Senior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Olga G. Olkhovskaya — Ph.D., Senior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Dmitry S. Boykov — Junior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Yulia S. Sharova — Ph.D., Junior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Nikita O. Savenko — Junior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.

Alexey M. Kotelnikov — Junior scientist; Keldysh Institute of Applied Mathematics RAS, Miusskaya ploshchad' 4, 125047, Moscow, Russia.