

DE GRUYTER

Yu. N. Grigoryev, V. A. Vshivkov, M. P. Fedoruk

NUMERICAL „PARTICLE-IN-CELL“ METHODS



Numerical "Particle-in-Cell" Methods
theory and applications

NUMERICAL "PARTICLE-IN-CELL" METHODS

THEORY AND APPLICATIONS

Yu.N. Grigoryev, V.A. Vshivkov and M.P. Fedoruk

///VSP///

Utrecht • Boston, 2002

VSP BV
P.O. Box 346
3700 AH Zeist
The Netherlands

Tel: +31 30 692 5790
Fax: +31 30 693 2081
vsppub@compuserve.com
www.vsppub.com

© VSP BV 2002

First published in 2002

ISBN 90-6764-368-8

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

Preface

Algorithms known as "particle" methods are becoming more and more widespread in modern mathematical modeling. The characteristic feature of these methods is the discretization technique when the set of discrete objects is introduced that are model "particles" considered as some mesh of moving nodes.

Numerical models based on particle methods are widely used for full-scale computational experiments. The results of such experiments are of great significance for science and practice. For example, the modifications of these methods are applied in the simulation of complex physical processes in Tocomacs and other unique plasma facilities of thermonuclear fusion. The effect on continuum media of powerful explosions causing great volume deformations, strong shock waves, shears and discontinuities was successfully modeled by particle methods in nuclear centers of USA and Russia.

The aerodynamics of rarefied gases is an extensive field for the application of such methods, which were used for the calculations of the flow past complex spacecrafts, such as "Shuttle" and "Buran".

Until recently, particle methods have been developed mainly by specialists in applied calculations. They were looking for an alternative to classical numerical methods, first of all, to solve problems of plasma physics, deformed continuum media, rarefied gases, etc. As a result no specialized monographs and textbooks on particle methods are available, as for finite-difference or finite-element methods.

There are certainly some manuals on the application of "particle" methods in different areas of physics and mechanics. Most of them deal with plasma physics problems. All of these books contain much methodological information that is useful for specialists in computations, but, strictly speaking, they are not books on computational mathematics. From this point of view, their common drawback is dealing too deeply with the specifics of the field. Besides, the mathematical part often suffers from verbal descriptiveness, and is presented as certain "know-how". It is clear that all this makes grasping the essence of computational methods difficult.

This book deals with combined Lagrangian-Eulerian schemes of the "particle-in-cell" type, most widespread among particle methods. The authors describe a universal approach to the construction of such algorithms. The approach is based on splitting the initial problem by which the auxiliary problem with a hyperbolic (divergent) operator is separated. After special discretization of the solution, such a splitting naturally leads to the well-known schemes of "particle-in-cell" methods.

The Introduction gives a general characteristic of particle methods. It contains characteristic properties of this group of numerical algorithms. It briefly shows their development in retrospective, and contains

a review of supplements.

Chapter 1 describes a general scheme of the construction of algorithms of the particle-in-cell type, the particle models used, and their interpolating and approximating properties.

Chapter 2 deals with the implementation of particle-in-cell methods on unstructured grids the development of which is started only during the last ten years.

Chapter 3 presents particle-in-cell methods for problems of inviscid gas dynamics problems, that originated from the well-known paper of F. H. Harlow.

Chapter 4 is devoted to schemes that can be described as "vortex-in-cell" methods used in the calculations of vortex flows. Alongside with the well-known scheme for two-dimensional flows, a new original scheme for three-dimensional flows based on the concept of Lamb's vortex impulse, which provides a convenient vorticity measure, is presented.

Chapter 5 contains particle-in-cell algorithms for the modeling of processes in collisionless plasma.

Chapter 6 introduces statistical particle-in-cell methods used mostly for numerical research of rarefied gas dynamics, disperse coagulate systems, and solid plasma.

The subroutines of universal blocks used in "particle-in-cell" methods written in Fortran 77, which are ready for use by the readers, are presented in the Supplement.

The chapters devoted to concrete applications of particle-in-cell methods begin with a brief statement of a mathematical problem. Then the corresponding algorithm is introduced on the basis of the general scheme and standard discretization. After this, the specific character of its implementation for a certain class of problems is considered.

Examples of calculations give the readers an idea of the capabilities of particle-in-cell methods, their requirements to computers, and the degree of precision that can be achieved. Modifications of the presented schemes used in other supplements are briefly annotated.

In selecting and presenting the material, we used our experience in mathematical modeling by particle methods as well as methodological approaches we developed in special courses of lectures for students of Novosibirsk State University.

To make the book more compact, we tried to separate the general elements of the presented algorithms and to find an optimal combination between the rigor and heuristics in style.

The contribution of the authors in the book is as follows. Chapters 1,2 and 3 were written by the authors jointly. Chapters 4,6 were written by Yu. N. Grigoryev, and Chapter 5 was written by V. A. Vshivkov and M. P. Fedoruk who also prepared the subroutines given in Supplement.

The book is primarily intended for specialists in calculations who want to get a general idea of numerical particle-in-cell methods and the sphere of their application. As a methodological guide it will be of

interest to undergraduate and postgraduate students, mathematicians and physicists specializing in mathematical modeling. We hope that the book will also be of interest to specialists familiar with the methods.

Authors

Contents

Introduction. Computational particle methods	5
General characteristics	5
Some applications	9
1. Particle-in-cell methods	15
1.1. Introduction	15
1.2. General scheme	16
1.3. Model particles and their properties	22
1.3.1. General properties of the particles	22
1.3.2. Widespread models of the particles	25
1.3.3. The improvement of the interpolation smoothness .	32
1.4. Errors of the particle-in-cell schemes	35
1.4.1. The self-force effect	35
1.4.2. The influence of the mesh periodicity	41
1.4.3. The errors of interpolation	43
1.5. The continuity equation in the particle method	48
2. Particle-in-cell methods on unstructured meshes	56
2.1. Introduction	56
2.2. Finite-elements bases	58
2.3. The Lagrangian step on unstructured meshes	62
2.3.1. Particle localization algorithms	63
2.3.2. The interpolation procedures	67
2.3.3. Numerical examples	72
2.3.4. The Lagrangian step with auxiliary Cartesian mesh	75
2.4. The Euler step. The finite-volume method	78
3. The particle methods in gas dynamics	85
3.1. Introduction	85
3.2. Basic equations	86
3.2.1. Gas dynamics equations in divergent form	86
3.2.2. The splitting scheme	87
3.3. The realization of the method	88
3.3.1. Harlow scheme	88
3.3.2. The "coarse" particle method and FLIC-method .	94
3.4. The combined particle method	97
3.5. The example of application	99
4. Vortex-in-cell methods	102
4.1. Introduction	102
4.2. Vorticity dynamics in two-dimensional flows	103
4.2.1. Basic equations	103
4.2.2. The conservation laws	104

4.3. The vortex-in-cell method in two-dimensional case	104
4.3.1. Splitting on physical processes	104
4.3.2. The systems of discrete vortices and their properties	105
4.3.3. The computational cycle of the method	109
4.4. The dynamics of vortices in three-dimensional flows	111
4.4.1. The vorticity evolution in the three-dimensional case	111
4.4.2. The equation for the Lamb impulse density	112
4.5. The vortex-in-cell scheme for three-dimensional flows	114
4.5.1. The splitting scheme	114
4.5.2. The Lagrangian vortex mesh	114
4.5.3. Computational cycle of the three-dimensional scheme	116
4.6. The examples of applications	119
5. Particle-in-cell methods in collisionless plasma dynamics	124
5.1. Introduction	124
5.2. Collisionless plasma basic equations	126
5.2.1. Kinetic equation with a self-consistent field	126
5.2.2. Particle motion equations	127
5.3. General scheme and computation cycle of the method	129
5.3.1. Initial data	129
5.3.2. Computation cycle	131
5.3.3. Numerical schemes for dynamic equations of model particles	133
5.4. Conservation laws in model plasma	138
5.5. Examples of applications	143
5.5.1. The dynamics of ion-acoustic waves of the arbitrary amplitude in non-isothermal plasma	143
5.5.2. Shock waves in plasma with the magnetic field	151
5.5.3. Numerical simulation of explosion phenomena in collisionless plasma at super-alfvenic speed	157
5.5.4. Interaction of the laser impulse with plasma	166
6. Statistical particle-in-cell methods	173
6.1. Introduction	173
6.2. Kinetic equations of rarefied gas	174
6.2.1. Boltzmann equations	174
6.2.2. The splitting up scheme	177
6.2.3. The master kinetic equation	179
6.2.4. Initial and boundary conditions	180
6.3. Some procedures of Monte Carlo methods	182
6.3.1. The inversion method	182
6.3.2. Simulation of joint distributions	184
6.3.3. The rejection method (by J. Neumann)	186
6.3.4. The superposition method	187
6.4. Statistical particle methods	187
6.4.1. General characteristics	187

6.4.2. The scheme and calculation cycle for the Boltzmann equation	189
6.4.3. The scheme and calculation cycle for the master kinetic equation	196
6.4.4. The Bird algorithm	201
6.5. Examples of the application	203
Supplements	208
A. Subroutine of initial data preparation	208
B. Subroutines of interpolation between the Lagrangian and Eulerian meshes	210
B1. Interpolation of the mesh vector-function to the Lagrangian mesh of particles	210
B2. Interpolation of the scalar function from the Lagrangian mesh of particles to nodes of the Eulerian mesh	211
B3. The subroutine of interpolation of generalized fields to the particle location on unstructured grids	214
B4. The subroutine for assignment of the particle charge on unstructured grids	215
B5. The subroutine for the determination of the scalar density in the nodes of unstructured grids	216
C. Subroutine for the particle dynamics	217
C1. Subroutine for calculation of the particles dynamics in fields of mass forces	217
C2. The subroutine for relativistic particle pusher according to Boris	220
D. The subroutines of a localization of particles on the unstructured grid	221
D1. The subroutines of particle localization on two-dimensional triangular grid (Löhner's algorithm)	222
D2. The subroutines of particle localization on three-dimensional tetrahedrons grid (Assous algorithm)	223
E. The subroutines for calculation of linear shape-functions on unstructured grids	227
E1. The subroutine for calculation shape-functions with respect to the particle locations in two-dimensional case	227
E2. The subroutine of calculation of the shape-functions with respect to the particle locations for three-dimensional case	228
F. The auxiliary subroutines	229
F1. The subroutine of determination of the local coordinates of point r (r -vector)	229
F2. The subroutine for determination of auxiliary vectors of tetrahedrons with nodes (k_1, k_2, k_3, k_4)	229
F3. The subroutine used in subroutine ploc3	230
F4. The subroutine for Gauss elimination	231

G. Subroutines for the solution of the Poisson equation (Poisson solvers)	232
G1. Direct method	232
G2. Combined iteration method	235
H. Subroutine of numerical integration of the full system of Maxwell equations	236
Bibliography	241

Introduction. Computational particle methods

General characteristics

The development of computer techniques in the last ten years has made mathematical modeling in a universal tool for solving applied problems and scientific researches open to wide auditorium of users.

The starting point of mathematical modeling is the construction of a computational model. It requires a transition from the initial problem in the form of differential or integral equations to a discrete algebraic approximation. The choice of discretization method significantly influences the type of computational model, its further algorithmization, and the computer program architecture on the basis of which the model is realized on the computer.

The best known is the discretization method that uses finite-difference approximations of differential operators or quadrature formulas for integral defined on the spaces of mesh functions, in other words, finite-difference methods or schemes.

"Particle" methods, as well as finite-difference methods, comprise a special group of computational algorithms combined by the discretization method of the initial mathematical problem. One can single out the following distinctive features of these algorithms.

Areas of applicability. Without loss of generality it is possible to say that particle methods are applicable to problems that deal with the time evolution of a certain medium or with the result of such evolution. In this case, the medium itself can be arbitrary. For example, it can be continuous or corpuscular, uniform or non-uniform, multicomponent, homogeneous or heterogeneous, etc. It can be an incompressible fluid, a dense or rarefied gas, a plasma of ions and electrons, a wave ensemble, a disperse system of aerosol type, an antagonistic coexisting populations, and so on.

Equations of continuum mechanics in various forms, kinetic equations of different level of closure, multiparticle dynamic systems, hyperbolic systems of equations of mathematical physics can be considered as typical mathematical models of evolution.

Discretization method. A common feature of particle methods is the concept of transition to discrete approximation. It consists in a model representation of the medium as a finite set of individual elements (particles) interacting with one another. Each particle becomes a carrier of certain characteristics — attributes of a given medium, such as electric charge, inertial or gravitating mass, mechanical momentum, kinetic energy, coordinates in some space, etc. Some of these individual properties are constant and do not change in the process of the medium

evolution. This is convenient for computations, because this conservatism corresponds to the conservation laws in the initial statement of a problem.

All medium characteristics, additive with respect to Lebesgue measure of the domain, are calculated by a simple summation of the corresponding values of all particles in the domain at a given time moment. By dividing this sum into the Lebesgue measure it is possible to determine density characteristics of the medium.

Classification of particle methods. Observing the time variation of the particle coordinates in a space filled with an evolving medium, we have a moving point set which it is usually called Lagrangian mesh in particle methods. The term "Lagrangian", as well as the term "Eulerian", which are often used in the present monograph, are borrowed from hydrodynamics [1]. They are based on two approaches to mathematical description of a moving fluid. In the approach connected with the name of Lagrange, the fluid is treated as a continuum of material points. In this case, all flow characteristics are presented as functions along trajectories (Lagrangian variables) of these points. The Eulerian approach consists in treating the fluid characteristics as functions in a given coordinate system fixed in the space in which the fluid is placed.

Purely Lagrangian and mixed Eulerian-Lagrangian or semi-Lagrangian algorithms are distinguished in particle methods. Leaving the details for future consideration, we can say that algorithms of the first group reduce to the numerical integration of systems of differential equations of the dynamic type [2], which describe the trajectories of the interacting particles.

Particle-in-cell methods described below belong to the group of mixed algorithms. It is characterized by the evolution of a system of particles at every time step being splitted in two stages. At the first step, under fixed states of particles their interaction and (or) the result of their joint effect on the medium are calculated preliminarily. The calculation is carried out on a stationary ("Eulerian") mesh. That is why this step is called Eulerian. At the next ("Lagrangian") step the integration of the dynamic system, the right hand side which was calculated at the Eulerian step, is performed for the next time step.

Elements of direct modeling. Purely Lagrangian schemes are more often used in cases when initially the medium consists of discrete particles. It can be ion or electron plasma, a galaxy of separate planets, or discrete vortices in an ideal fluid. Here model particles are called by the same names as particles of a real medium, i.e., ions, molecules, or vortices. They have the same characteristics, such as charge, mass, and circulation. Their interaction is described by the Coulomb, Newton, Biot-Savart laws. Therefore, such models are usually called imitative. Calculations by these methods are closest to direct mathematical modeling. In particle-in-cell methods particle interaction is described by means of "fields" of different nature calculated on the Eulerian mesh.

In some cases, the scattering problem is used to describe it. Nevertheless, some of direct modeling elements are also present in such schemes.

Properties of model particles. Even in purely Lagrangian models correspondence of model particles to their real prototypes is far from complete. That is why it would be a mistake to consider that they are identical. Very often model particles have geometric shapes of finite dimensions, inside of which the distribution of individual particle features, such as the charge, mass, and vorticity is given. The choice of a distribution in particles can be determined by the wish to smooth out the singularity of their interaction as well as non-physical fluctuations in the solutions due to a finite number of particles. It can be also determined by the necessity to increase the collective character of interaction or reproduce a certain effect. The shape of particles is connected with the symmetry of the problem and with the desire to simplify the calculation of their dynamics, etc. As a rule, both the shape of particles and the distribution of features in them remain constant in the process of modeling, that can contradict a real situation. For example, this is what is happening with model vortices in an ideal fluid where such stability does not comply to the "freezing-in" property of vorticity [2].

In all cases, there is a significant difference between the number of model particles and real particles. For instance, in the modeling collisionless plasma (see Chapter 5) each model ion corresponds to dozens of millions of physical ions. In other particle methods this ratio can be even higher.

All these differences from an original real medium indicate that computational particle methods require "tuning" in the process of numerical calculation, which is a "know-how". Its success is determined not only by the skill of a scientist-programmer who performs computations, but also by his understanding of the phenomenon being modeled.

Computational peculiarities. Particle methods are usually characterized by a relatively low accuracy. Usual errors constitute several percent. This is the result of a compromise between the reasonable volume of calculations and the possibility to model complicated phenomena. Therefore the accuracy of the auxiliary numerical procedures used in the programs of particle methods, for example, the finite-difference schemes for equations of motion, is not high order. This approach saves computational time considerably. Usually particle-in-cell methods have a high reserve of stability that makes it possible to move relatively quickly over the evolutionary variable. It also saves time when we deal with problems solved by iteration methods.

Although their accuracy of computations is not very high, particle-in-cell methods reproduce adequately many of the fine effects of original mathematical models which prove to be inaccessible to other numerical schemes. This interesting property can be explained only partially by the absence of scheme viscosity in them. It accounts for a certain intrigue of particle methods and their attraction for those scientists who

used the methods at least once.

Although the number of model particles is always by many orders of magnitude lower than that of particles forming the initial medium, their absolute number is rather high. This is connected with the fact that the accuracy of particle methods increases with the growth in the number of Lagrangian mesh nodes N though not linearly. With the development of computers, the typical number of particles used in ordinary calculations increased in the range $N = 10^3 \div 10^6$. The complexity of purely Lagrangian methods is estimated by the value of $O(N^2)$. Good algorithms of particle-in-cell methods are usually economical and their complexity is $O(N)$ or $O(N \ln N)$. From this one can conclude that all particle methods are characterized by a large amount of computational labour. Their implementation requires high-speed computers with a large memory capacity. The computational resources that are necessary for three-dimensional and two-dimensional non-stationary problems reach the level of modern supercomputers. That is why until recently particle methods have not been widely used in practice and were developed by separate groups scientists only in large scientific centers.

Software implementation. Particle methods are easily adapted to computers with parallel architecture. This allows us to hope that with the coming into use of parallel computers particle methods will become available to a broad range of users. The efficiency of modeling on their basis will also increase.

Particle-in-cell methods based on the idea of splitting can be especially easily converted into parallel computations. A major technological advantage of these methods is that the motion of particles in each cell of the Eulerian mesh is determined only by the local values of the "fields" and does not depend on the generalized "coordinates" and "momentums" of other particles. It is just this independence of the motion of individual particles that makes it possible to convert them into parallel algorithms. The parallelizing of algorithms makes it possible to use modern multiprocessor computers and save significantly the computing time per one particle. This, in turn, gives an opportunity to increase the number of model particles, and thus raise the computational accuracy which is necessary for the investigation of fine physical processes. However, when using multiprocessor computers for particle-in-cell methods there exists the problem of uniform loading of processors. It is connected with the fact that the data arrays in processors are divided into two groups. One group contains the data for particles, such as coordinates, velocity components, masses, charges, and other characteristics. The volume of those arrays is directly proportional to the number of model particles. The other group contains arrays of the values of mesh density functions, such as the pressure, strengths of electric and magnetic fields, etc. If in parallel method equal parts of the mesh domain are allotted to each processor, the number of computations can vary greatly due to a different number of particles in each processor. If at the starting

point the mesh domain is divided in such a way as to equalize the number of particles in each processor, the load of some processors during computation process can also deviate significantly from the mean value due to the motion of particles from one subdomain to another. That is why to optimize particle-in-cell methods of computation on multiprocessor computers, it is necessary to use special methods of load balancing [3, 4, 5, 6].

Computer programs for the mathematical modeling by particle methods have a large volume, and can contain several thousands of operators. The creation of such a program requires much work of a computer programmer, and can take much time. Under these conditions, one should use the experience accumulated in the known programs of this type [7, 8]. The consideration of particle-in-cell methods as uniform group of computational algorithms, which one can find in the this book, makes it possible to separate universal blocks and models, to give general recommendations on how to overcome computational difficulties, and to unify the general structure of program control. We hope it will help in the generalization and greater use of the results from those areas of computational mechanics and computational physics where these methods are more developed. Some specially selected universal subroutines given in Supplement serve the same purpose.

Some applications

Particle methods historically appeared and developed as an alternative to classical algorithms and, in the first place, to finite-difference schemes. Their use was frequently connected with hopes to reproduce complex effects in computing experiments on the basis of their imitative properties. Besides, particle methods have certain advantages of purely computational nature. These are, for example, the absence of approximating viscosity, the possibility of through computation when there are strong changes of solution, and some others.

Originally, particle methods were mostly being developed in those research and applied areas that required large-scale computational experiments. To conduct such computations, great intellectual and computer resources were involved. The studies on nuclear weapons and controlled thermonuclear fusion performed by Russia and USA in the 60-s and 80-s can serve as examples. Since then, their application has been constantly increasing. Here, however, we restrict ourselves to a brief survey of those methods that are treated in the subsequent chapters.

1°. Particle-in-cell methods for solving gas dynamics problems described in Chapter 3, appeared in 1955 (see papers [9, 10] of F.H.Harlow with collaborators from Los-Alamos Laboratory). From the very beginning the method was oriented to solving problems with great bulk deformations, strong shears, instabilities on the interfaces of various media where finite-difference schemes proved to be inapplicable. The possi-

bility to use the powerful computers the Laboratory had at that time played decisive role. Numerous computations obtained for problems of reduction of spherical bubble by cylindrical shock wave and powerful near surface shots causing crater formation were typical in those investigations.

In mid-sixties, there appeared in the literature a description of computer programs with use of Harlow type particle method, earlier developed in Russian nuclear centers [11]. In addition to above mentioned problems of explosions and shock waves, those of turbulent mixing at the surface of bounded double-layer fluid with unstable stratification, high-speed collision of bodies, and some other problems were solved. The school of computational scientists created by N. N. Yanenko in Russia made a fundamental contribution to the development of particle methods. Effective modifications of Harlow method used in the mathematical modeling of multicomponent and multiphase media under great deformations and complex intersections of boundary surfaces, in plasma physics problems, in the mechanics of viscous, viscoplastic and elastoplastic media were presented in the studies of his followers. Unfortunately, until recently the most of these results were known only to a limited number of Russian specialists.

In paper [12] N. N. Yanenko et al., for the first time, the idea of splitting in such algorithms was emphasized. Their work had played a key role in forming the modern representation of particle-in-cell methods. In presenting the material we tried to call attention to the important role of this approach.

An essential progress of particle-in-cell methods for solving computational aerodynamics problems was achieved thanks to another Russian school of computer scientists headed by O. M. Belotserkovsky. This school created, in particular, the economic modification of the Harlow scheme known in Russia as "coarse" ("large") particle method. This modification made it possible to calculate complex flows on the medium-power computers that Russian computational scientists had in the 60-ies. The modification has given an opportunity to calculate transonic flow past head parts of airplanes and missiles, flow past bodies of finite size with disruption zones and recirculation in the bottom area, two-phase flows, etc. in the period from mid 70-ies to mid 80-ies. A description of these and other results demonstrated the possibilities of the method one can find in the [13]. This book was the first guide in Russia on particle-in-cell methods for solving aerohydrodynamics problems that contained a sufficiently complete bibliography of the Russian and foreign literature on the subject.

2°. Numerical modeling of vortex flows by methods of discrete vortices [1, 14] were originally based on the numerous results obtained for point vortex dynamic systems in theoretical hydrodynamics. The Hamiltonian systems of point vortices on a plane were considered still by Kirchhoff [15]. Karman vortex streets and Karman's model of hy-

drodynamic drag of bodies are also well known [1]. For the first time, numerical experiments on instability of tangent discontinuity in fluid modeled by a street of point vortices was carried out in [16] in 1931. One may say that historically this was the very first computation by particle method. Some attempts to develop the idea on first digital computers were made late in the 50s — early in the 60s. In these studies purely Lagrangian algorithms were developed in which all vortices in the system interacted with one another accordingly with a long-range interaction model (Biot-Savart integral).

As to later studies on the subject, one should mention the algorithm for a system of vortices with a finite kernel [17]. In this paper grid free Lagrangian algorithm was suggested for the computation of weakly viscous flows with high Reynolds numbers. This particle method made possible the maximum reduction of the approximation viscosity. The computation of turbulent flow past bodies by discrete vortex method was also proposed first in this paper. The application of such methods for the modeling of turbulent flows received some development in succeeding years.

The development of vortex-in-cell algorithms (VIC) discussed in Chapter 4 was started in the early 70s. They were created on the basis of particle methods in rarefied plasma that had been already developed in detail by that time.

The direct analogies in the mathematical description of ideal fluid with vorticity and collisionless plasma (see Chapters 4 and 5) facilitated the development. That is why many computational procedures from plasma problems are used in VIC methods. Among these are the choice of particle kernels, fast Poisson solvers, some methods of interpolation between the Lagrangian and Eulerian meshes, and many others. As a result the known universal complex of programs VORTEX [8] was used not only for computation of plane flows with vorticity, but also for the computation the dynamics of currents in magnetized plasma in "guiding center" approximation.

The problems of vortex sheet roll-up [18], as well as those of the instabilities of a thin shear layer and compact areas of concentrated vorticity [19] were solved by VIC-methods.

It should be noted that all mentioned above refers only to two-dimensional flows. Until recently, there were no algorithms of the vortex-in-cell type for three-dimensional problems. In this case, the analogy with plasma is much more complicated than in the case of plane flows. Moreover, three-dimensional particle algorithms for plasma problems are much less developed, especially when currents are present.

On the whole, vortex-in-cell methods were not often used, despite their potential. First of all, this can be explained by the fact that in applied hydrodynamics no researches similar to the most important researches of controlled thermonuclear fusion are performed. Nevertheless, vortex-in-cell methods will undoubtedly occupy their proper place in the

software of the future computational "wind tunnel" as a supercomputer for solving aerohydrodynamic problems of any degree of complexity is figuratively called.

3°. In computational plasma physics, particle methods came into practice in mid-fifties, and were used in the calculations of electron and ion beams in high current vacuum devices [20]. These models were of purely Lagrangian type and contained a small number of particles interacting by the Coulomb law.

A particle method was first used to solve purely plasma problems at the end of the 50s — beginning of the 60s [21]. These studies were devoted to one-dimensional plasma vibrations. Model particles had the shape of charged planes ("sheets"). The algorithms were also purely Lagrangian.

The creation of the particle-in-cell algorithms for plasma considered in Chapter 5 was started under the direct influence of the Harlow group. Here, however, they found the most wide application and were justified from the theoretical point of view. First of all, many countries devoted much effort to the investigations of thermonuclear fusion problems in the 60s — 80s. It turned out, that many important processes of this problem can be easily described by Vlasov-Maxwell equations and the so-called hybrid models, for which particle-in-cell methods proved to be extremely effective.

The behavior of plasma is influenced by multiple effects, such as different types of instabilities, collisionless wave attenuation, superposition of collective and individual interactions. This forced investigators to study the computational aspects of the methods in order to separate physical effects and computation errors. The influence of the space mesh periodicity, particle kernels and interpolation methods, as well as the violation of conservation laws was investigated. As a result detailed plasma computational models were created on the basis of particle-in-cell methods. The computational experiments based on these models have given a possibility to select constructive decisions for experimental installations such as "Tokomak" and others which are unique from the point of view of their scale and cost. Effects impossible to model in laboratory experiments were reproduced by computations and this made it possible to advance the physical theory of plasma.

At present, particle-in-cell methods are successfully used in astrophysics for the numerical simulation of the interaction of Galaxies on space scales and that of solid state plasma in semiconductors on submicron scales, where direct physical experiments are simply impossible.

A number of monographs [7, 21, 22] dealing with particle methods in computational plasma physics is known. In particular, a fundamental computational code which can be used not only for collisionless plasma but for other applications, for example, to model the dynamics of Galaxies or for a discrete vortex system in a plane flow of an ideal fluid, is described in detail in [7].

Unfortunately, many methodological results that could be applied to other problems are strongly connected with the physics of concrete phenomena. To use these results, one must see physical analogies, for instance, between "phase" and real fluid flows or between electric currents, vortex filaments and gravitating planets in the ecliptic plane, etc. In this reason, it was not easy for computational scientists to access the results. Therefore it is necessary to unify the methodical results. To some extent, this is what the authors of this book are trying to do.

4°. Particle methods in rarefied gas dynamics developed simultaneously with the transformation of rarefied gas dynamics into a separate branch of mechanics. This happened during the sixties — seventies. At this time there was rapid development of space science and techniques, as well as vacuum technologies in microelectronics and chemistry, the powerful vacuum facilities were built for scientific researches. In these conditions, numerous problems were put forward which required large-scale computational experiments. Many scientists were involved in solving of these problems, and necessary computer capacities were accumulated. The physical model of rarefied gas naturally led to imitation algorithms which were varieties of particle methods. Thus, favorable conditions for the development of particle methods appeared in rarefied gas dynamics in the same way as in plasma physics.

Due to the statistical character of microscopic processes in rarefied gas all numerical models containing imitating elements are also statistical. That is why, in literature on the rarefied gas dynamics these models are traditionally referred to as Monte Carlo statistical methods.

Statistical particle methods in rarefied gas dynamics developed in two directions. One of them is based on the iteration of linear operators by Monte Carlo method [23]. A semi-empirical application of the method to solving some problems of rarefied gas flows was described in [24]. Mathematical proof of Monte Carlo iteration algorithm for the Boltzmann nonlinear kinetic equation was carried out in [25]. The survey of the further development of this approach and its applications to the typical problems of rarefied gas aerodynamics can be found in [26, 27].

The inventor of particle-in-cell statistical methods was G. A. Bird. In a series of his papers, in mid-sixties, he created the first working version of the method, and demonstrated its efficiency for solving one- and two-dimensional rarefied gas flow problems. As to Russian computation scientists, O. M. Belotserkovsky and B. Ye. Yanitsky [28, 29] made the most contribution into the development and proof of the method.

The wide use of the method was supported Bird's book [30]. The book written for applied computation scientists is still a convenient guide for the initial practical becoming familiar with the method. In particular, it includes Fortran subroutines, which can be used to compile programs for a variety of rarefied gas dynamics problems.

At present, many computations in aerodynamics of space crafts, from

the simplest ones to the most complicated, such as American "Shuttle" and Russian "Buran", were carried out on the basis of this method. One can say that the particle-in-cell statistical methods surpass all other numerical methods in the number of applications for solving rarefied gas dynamics problems. This, of course, does not mean that in rarefied gas dynamics particle method is any more universal than in other applications. The thing is, at this stage practical demands, in particular, those of rarefied gas aerodynamics, and the possibilities of modern computers created conditions propitious for their foremost development.

Methods belonging to the described group are often designated in the West literature on the subject by the abbreviation DMCS (Direct Monte-Carlo Simulation). In Russian publications little by little also started using the equivalent term. However, this terminology only reflects the specifics of realization and it should not confuse the reader intent on getting to know particle-in-cell methods. It should be mentioned that in their time G. A. Bird and O. M. Belotserkovsky gave its due to Harlow's idea and were consistent in representing their schemes as varieties of particle-in-cell methods. One can think that such an universal approach made possible further application of these methods outside the rarefied gas dynamics [31, 32]. That is why the presentation of these algorithms in the monograph on particle-in-cell methods is fully justified.

Chapter 1

Particle-in-cell methods

1.1. Introduction

Great advances in numerical particle-in-cell methods in such important branches as computational mechanics of continuous media and computational plasma physics stimulated their further spreading in other areas of mathematical modeling. In connection with a rapid increase in the power of personal computers in the last decade, these methods used earlier only in large scientific supercomputer centers became accessible to a wide range of computer users. However, there is no literature on the numerical methods yet where numerical schemes of particle-in-cell methods could be described as the subject of computational mathematics without any connection to physical models, as complicated and specific as, for example, gas plasma.

This approach is developed in this book, and the first chapter gives us the general methodical basis of the chapters that follow. To unify the interpretation, special terminology is introduced in p. 1.2. In general, it was already used in original studies of various authors. The terminology gives us the possibility to distract from concrete physical models and to concentrate on general computational characteristics of these methods. The scheme of construction of particle-in-cell algorithm is given in abstract form. The approach is based on approximate factorization of the initial problem. At factorization, an auxiliary problem with a divergent (hyperbolic) operator is selected. It is shown that the approximate solution of this problem is reduced, after special discretization, to numerical integration by steps with the evolutional variable of some dynamic system of model particles. This part represents the Lagrangian step which is universal for all particle-in-cell methods.

General requirements to particle models are formulated in p. 1.3. The fulfilment of these requirements determines the calculation errors. Commonly used models, with wide applications of methods in gas dynamics, plasma physics, vortex flows and astrophysics, are considered here. The particle model connection with smoothness of interpolation procedures between Lagrangian and Eulerian meshes is also shown which affects greatly the computational properties of this class of numerical schemes.

The mechanisms of appearance of some characteristic numerical errors of particle-in-cell methods are considered in p. 1.4. These are, in particular, the "self-force" effect of particles connected with the calcu-

lation on the Euler mesh of the force fields affecting the particles as well as the appearance of non-physical grid harmonics in the spectra of mesh functions. The methods for reducing these and other errors are presented. A systematic procedure of improving the particle models is suggested.

In p. 1.5 some opportunities for improving the difference generalized density conservation law typical for many problems are discussed.

1.2. General scheme

Let us consider the scheme of construction of particle-in-cell algorithm. Without loss of generality, we can assume that there is an evolutional variable t in the initial problem, and the problem is written in the following abstract operator form:

$$\frac{\partial \mathbf{q}}{\partial t} + A\mathbf{q} = 0. \quad (1)$$

Here $\mathbf{q}(t, \mathbf{u})$ is a vector-function with values in R^n , and \mathbf{u} is a vector of independent variables with the domain of definition $\Omega \subset R^m$, $t \in R_+^1$. The solution of the stationary problem is obtained by the method of establishment.

The splitting scheme. At the step τ of the evolutional variable t the solution of the problem is approximately factorized in the form

$$\mathbf{q}(t_p + \tau) \cong (I + \tau E)(I + \tau L)\mathbf{q}(t_p),$$

where $E + L = A$, I is the identity operator. The key point here is the separation of the divergent operator L in A .

As a result the solution Eq. (1) at the step τ is reduced to sequential solution of two auxiliary problems

$$\frac{\partial \tilde{\mathbf{q}}}{\partial t} + E\tilde{\mathbf{q}} = 0, \quad \tilde{\mathbf{q}}(t_p) = \mathbf{q}(p\tau), \quad (2)$$

$$\frac{\partial \mathbf{q}}{\partial t} + L\mathbf{q} = 0, \quad \mathbf{q}(t_p) = \tilde{\mathbf{q}}(t_p + \tau), \quad (3)$$

If operator A has divergent form or is reduced to it, then the algorithm can be constructed directly for the initial problem.

Let a set $\{\omega_\alpha\}$ of cells forms a partition of domain Ω . The collection of the cells and nodes chosen while partitioning, for instance, the centers of the cells $\{\mathbf{u}_\alpha\}$ forms the mesh domain Ω_h or the *Euler* mesh.

Euler step. It's customary to name solving of the auxiliary problem Eq. (2) by the Euler step. Rather various numerical schemes are used in different particle-in-cell methods for its realization. They all are united by the only fact that all calculations are carried out on the fixed Euler mesh. That is why it is inexpedient to discuss this part of the

algorithm within the framework of the general scheme. It will be done in the following chapters for the concrete algorithms. Besides, in this chapter we restrict ourselves by the case of regular Cartesian meshes. For them methods of particles developed traditionally up to nowadays. The problems of particle methods realization on unstructured meshes are separately considered in the following chapter.

Lagrangian step equations. The second step is universal for methods particle-in-cell and it is called Lagrangian. The equations corresponding to the problem Eq. (3), as a rule, can be present in the following divergent form

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial(\mathbf{q}\mathbf{U})}{\partial \mathbf{u}} = 0. \quad (4)$$

For the sake of simplicity we assume that $\mathbf{U} = \mathbf{U}(t, \mathbf{u})$ — a vector-function with values in R^n . We'll remark that equations of such a type in mathematical physics are named the system of differential conservation laws. The integration Eq. (4) in the domain Ω leads to relations

$$\frac{\partial \mathbf{Q}}{\partial t} + \int_{\partial\Omega} dS(\mathbf{n} \cdot \mathbf{q}\mathbf{U}) = 0, \quad (5)$$

$$\mathbf{Q} = \int_{\Omega} d\mathbf{u} \mathbf{q}, \quad (6)$$

where \mathbf{n} is a unit vector of the local outward normal to boundary $\partial\Omega$ of the domain Ω and the integral term has the meaning of the flows of vector \mathbf{q} components "transferring" by the vector field \mathbf{U} . If this term vanishes or in the unbounded domain Ω aims for zero asymptotically with $|\mathbf{u}| \rightarrow \infty$ then the integral characteristic (6) is conserved in the process of evolution.

In physical terms widely used in these methods the domain Ω is customarily called *phase space*, and a set of independent variables \mathbf{u} is called *dynamic variables*, and material medium or substance the evolution of which is the contents of Lagrangian step can be called *phase fluid*. It's quite natural that vector \mathbf{q} should be called vector *density* of medium features, correspondingly the system of Eqs. (4) one can call *continuity* equations or *transition* equations of these features in phase space Ω .

Now one part of these terms became established notions, others, in our opinion, are suitable for universal interpretation of the particle-in-cell methods because their concrete embodiments are various enough. Indeed, the phase space can be a domain of the real gas flow, the area of existence of some population, the space of coordinates and velocities for plasma and rarefied gas, etc. Respectively, in the above examples phase fluid will be real gas, certain population, plasma or rarefied gas.

Discretization. The solution of problem Eq. (4) is represented in the form of the following interpolation formula

$$\mathbf{q}(t, \mathbf{u}) = \sum_{j=1}^N \mathbf{Q}_j R(\mathbf{u}, \mathbf{u}_j(t)), \quad (7)$$

This transition it is usual to call a discretization (partition) of phase fluid into (model) particles. Vectors \mathbf{Q}_j define the sets of features which are individual characteristics of the particles. It should be emphasized that \mathbf{Q}_j don't depend on time and, hence, they remain constant in the process of evolution. Function $R(\mathbf{u}, \mathbf{v})$ is called the kernel of a model particle. It defines the form of the particle and the distribution of the features in it, that are components of vector \mathbf{q} . Coordinates $\mathbf{u}(t)$ determine the current position of the (symmetry) centers of the particles in Ω . The kernel usually has the following universal properties

$$R(\mathbf{u}, \mathbf{v}) = R(\mathbf{v}, \mathbf{u}) \geq 0, \quad \frac{\partial R}{\partial \mathbf{u}} = -\frac{\partial R}{\partial \mathbf{v}}. \quad (8)$$

If the value (6) remains invariable, then it's necessary that the representation (7) should satisfy the conservation law in the form

$$\mathbf{Q} = \sum_{j=1}^N \mathbf{Q}_j. \quad (9)$$

In this case, the substitution Eq. (7) in Eq. (6) gives the following norming condition for the core R

$$\int_{\Omega} d\mathbf{u} R(\mathbf{u}, \mathbf{v}) = 1. \quad (10)$$

The equations of the motion of particles. The representation $\mathbf{q}(t, \mathbf{u})$ in the form (7) allows to reduce the solution of the auxiliary problem (3) to the integrating of the dynamic system of particles. The transition is realized in the following way. Let $\varphi(\mathbf{u})$ be the arbitrary smooth finite function. Substituting (7) into Eq. (4) and integrating in the domain Ω with weight function $\varphi(\mathbf{u})$, we receive successively

$$\begin{aligned} & \int_{\Omega} d\mathbf{u} \varphi(\mathbf{u}) \left(\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial (\mathbf{q} \mathbf{U})}{\partial \mathbf{u}} \right) = \\ & = \sum_{j=1}^N \mathbf{Q}_j \int_{\Omega} d\mathbf{u} \left(\frac{\partial R}{\partial \mathbf{u}_j} \frac{d\mathbf{u}_j}{dt} \varphi(\mathbf{u}) - R \mathbf{U} \frac{\partial \varphi}{\partial \mathbf{u}} \right) = \\ & = \sum_{j=1}^N \mathbf{Q}_j \left\{ \frac{d\mathbf{u}_j}{dt} \int_{\Omega} d\mathbf{u} R(\mathbf{u}, \mathbf{u}_j) \frac{\partial \varphi}{\partial \mathbf{u}} - \int_{\Omega} d\mathbf{u} R(\mathbf{u}, \mathbf{u}_j) \mathbf{U} \frac{\partial \varphi}{\partial \mathbf{u}} \right\} = 0. \end{aligned}$$

Making a change of variables $\mathbf{u} \leftrightarrow \mathbf{u}_j$, in the integrals we obtain

$$\sum_{j=1}^N \mathbf{Q}_j \int_{\Omega} d\mathbf{u}_j R(\mathbf{u}, \mathbf{u}_j) \frac{\partial \varphi(\mathbf{u}_j)}{\partial \mathbf{u}_j} \left[\frac{d\mathbf{u}_j}{dt} - \mathbf{U}(\mathbf{u}_j(t)) \right] = 0. \quad (11)$$

The properties of the kernel symmetry (8) and relations

$$\frac{\partial R(\mathbf{u}, \mathbf{u}_j(t))}{\partial t} = \frac{\partial R}{\partial \mathbf{u}_j} \frac{d\mathbf{u}_j}{dt} = -\frac{\partial R}{\partial \mathbf{u}} \frac{d\mathbf{u}_j}{dt}.$$

were used here under transforms.

Besides an arbitrariness $\varphi(\mathbf{u})$ the integrals in the last equality (11) must turn into zero identically. For that it's necessary to fulfill the following conditions

$$\frac{d\mathbf{u}_j}{dt} = \mathbf{U}(\mathbf{u}_j(t)), \quad j = 1, \dots, N. \quad (12)$$

This dynamic system is called equations of motion of model particles. Integrating it on the step τ , we observe corresponding permutations of particles along the trajectories $\{\mathbf{u}_j(t)\}$ in Ω . In this case, by virtue of (7), (11) space-time evolution of the solution \mathbf{q} of the system (4) is reproduced approximately in the interval τ . One can notice that the generalized field \mathbf{U} in the system (12) is calculated in the centers \mathbf{u}_j of the particles.

In a lot of cases phase space of the problem expands into the direct product of two subspaces $\Omega = \Omega_r \times \Omega_p$. Under these conditions dynamic variables \mathbf{u} subdivide into two groups $\mathbf{r} = \{r_1, \dots, r_s\}$ and $\mathbf{p} = \{p_1, \dots, p_s\}$ which can be defined as generalized coordinates and generalized momentums corresponding to them.

In these variables the Eqs. (4), as a rule, have the following characteristic form

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial}{\partial \mathbf{r}} (\mathbf{p} \mathbf{q}) + \frac{\partial}{\partial \mathbf{p}} (\mathbf{F} \mathbf{q}) = 0, \quad (13)$$

where $\mathbf{F}(\mathbf{r}, \mathbf{p})$ is an s -component vector of the generalized force field.

The discretization of the solution is fulfilled in the form

$$\mathbf{q}(t, \mathbf{r}, \mathbf{p}) = \sum_{j=1}^N \mathbf{Q}_j R(\mathbf{r}, \mathbf{r}_j(t)) \delta(\mathbf{p} - \mathbf{p}_j(t)), \quad (14)$$

where $\delta(\mathbf{p})$ is s -dimension Dirac delta function. The function of the kernel $R(\mathbf{r}, \mathbf{z})$ has the properties (8) and satisfies the norming condition

$$\int_{\Omega_r} d\mathbf{r} R(\mathbf{r}, \mathbf{z}) = 1. \quad (15)$$

It is seen, that in this case model particles have a finite Lebesgue measure in the coordinate subspace Ω_r and a singular one in the subspace of generalized momentums Ω_p .

For the transition to the dynamic system let us substitute Eq. (14) in the Eq. (13) and integrate on Ω with weight smooth finite function $\varphi(\mathbf{r}, \mathbf{p})$. Integrating on generalized momentums gives us

$$\sum_{j=1}^N \mathbf{Q}_j \int_{\Omega_r} d\mathbf{r} \left\{ \frac{\partial}{\partial t} [\varphi(\mathbf{r}, \mathbf{p}_j) R(\mathbf{r}, \mathbf{r}_j)] + \mathbf{p}_j(t) \frac{\partial}{\partial \mathbf{r}} [\varphi(\mathbf{r}, \mathbf{p}_j) R(\mathbf{r}, \mathbf{r}_j)] - \right.$$

$$F(\mathbf{r}, \mathbf{p}_j) \frac{\partial \varphi(\mathbf{r}, \mathbf{p}_j)}{\partial \mathbf{p}_j} R(\mathbf{r}, \mathbf{r}_j) \Big\} = 0.$$

By differentiation with respect to t and by integration by parts with the use of the properties (8) of the kernel $R(\mathbf{r}, \mathbf{r}_j)$ after the regrouping we obtain

$$\sum_{j=1}^N Q_j \left\{ \int_{\Omega_r} d\mathbf{r} \left[\frac{d\mathbf{r}_j}{dt} - \mathbf{p}_j \right] \frac{\partial \varphi(\mathbf{r}, \mathbf{p}_j)}{\partial \mathbf{r}} R(\mathbf{r}, \mathbf{r}_j) + \int_{\Omega_r} d\mathbf{r} \left[\frac{d\mathbf{p}_j(t)}{dt} - F(\mathbf{r}, \mathbf{p}_j) \right] \frac{\partial \varphi(\mathbf{r}, \mathbf{p}_j)}{\partial \mathbf{p}_j} R(\mathbf{r}, \mathbf{r}_j) \right\} = 0. \quad (16)$$

If we do the substitution of variables $\mathbf{r} \leftrightarrow \mathbf{r}_j$ in the second group of integrals, as we have done under transition to Eq. (11) then because of the arbitrariness of the weight function $\varphi(\mathbf{r}, \mathbf{p})$ the integrand expressions must vanish identically. This will be fulfilled on the solutions of the following dynamic differential system

$$\frac{d\mathbf{r}_j}{dt} = \mathbf{p}_j(t), \quad \frac{d\mathbf{p}_j}{dt} = \mathbf{F}(\mathbf{r}_j, \mathbf{p}_j) \quad j = 1, \dots, N. \quad (17)$$

Let us note that we can obtain some other form to the subsystem for generalized momentums $\{\mathbf{p}_j\}$. With this aim let us consider the conditions of vanishing of the second group of integrals having assumed that the weight function φ doesn't depend on the generalized coordinates.

$$\sum_{j=1}^N Q_j \int_{\Omega_r} d\mathbf{r} \left[\frac{d\mathbf{p}_j(t)}{dt} - F(\mathbf{r}, \mathbf{p}_j) \right] R(\mathbf{r}, \mathbf{p}_j) = 0.$$

This supposition is not too restricting because there remains enough store of arbitrariness in the kernel $R(\mathbf{r}, \mathbf{r}_j)$. Taking into account the norming conditions (15) for generalized momentums we receive the subsystem

$$\frac{d\mathbf{p}_j}{dt} = \bar{\mathbf{F}}(\mathbf{r}_j, \mathbf{p}_j) \quad j = 1, \dots, N; \quad (18)$$

where

$$\bar{\mathbf{F}}(\mathbf{r}_j, \mathbf{p}_j) = \int_{\Omega_r} d\mathbf{r} F(\mathbf{r}, \mathbf{p}_j) R(\mathbf{r}, \mathbf{r}_j). \quad (19)$$

Now the generalized force field acting on the model particle is summarized with the weight $R(\mathbf{r}, \mathbf{r}_j)$. Nonnegativity and normalization (15) of the kernel allows to consider the integration in (19) as the averaging with respect to distribution R . We can note that the transition from the subsystem Eqs. (19) to the subsystem in Eqs. (17) can be achieved by formally assuming that $R(\mathbf{r}, \mathbf{r}_j) = \delta(\mathbf{r} - \mathbf{r}_j)$. Though, as it is shown above, the system Eqs. (17) is obtained for the arbitrary kernel.

While numerical integration of the system Eqs. (12) permutations of particles in the limits of domain Ω and across boundary $\partial\Omega$ are usually

calculated separately at every time step τ . In this case, the fulfillment of the conservation law is required for every new state of the set of model particles. Therefore the norming condition of (10) of the kernel R must always be satisfied. It's also valid for the kernels with norming (15).

Interpolation "particles-grid". After finishing the Lagrangian step before next step τ it is necessary to project solution \mathbf{q} in the form (7) into the space of mesh functions $\{\tilde{\mathbf{q}}_\alpha\}$. This is done not only for calculations with helping of finite-difference schemes used at the Euler step, but also just for convenience of data processing. Under calculation of mesh functions according to the given state of set N of particles the fulfillment of the conservation law (6) must be provided. Using the generalization for R^m of the quadrature formula of middle rectangles with accuracy up to its error we can write the following

$$\mathbf{Q} = \int_{\Omega} d\mathbf{u} \mathbf{q} = \sum_{(\alpha)} |\omega_\alpha| \tilde{\mathbf{q}}_\alpha. \quad (20)$$

where $|\omega_\alpha|$ is Lebesgue measure of the cell ω_α .

Obviously the conservation law (6) will be carried out if we assume that the values of the mesh functions are calculated according to formulas

$$\tilde{\mathbf{q}}_\alpha = \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} d\mathbf{u} \mathbf{q}.$$

Substituting the solution representation (7) at the Euler step into the integral over cell ω_α we obtain the following formula

$$\tilde{\mathbf{q}}_\alpha = \sum_{j=1}^N \mathbf{Q}_j \bar{R}(\mathbf{u}_\alpha, \mathbf{u}_j), \quad (21)$$

where the newly introduced function named by the mesh kernel $\bar{R}(\mathbf{u}_\alpha, \mathbf{u}_j)$ is defined by the relation

$$\bar{R}(\mathbf{u}_\alpha, \mathbf{u}_j) = \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} d\mathbf{u} R(\mathbf{u}, \mathbf{u}_j). \quad (22)$$

Let us pay attention to the evident similarity of the formulas (7) and (21). It means that the solutions on the Lagrangian and Eulerian meshes are defined by formulas of the same structure, but with changing of the kernel $R(\mathbf{u}, \mathbf{u}_j)$ by the mesh kernel $\bar{R}(\mathbf{u}_\alpha, \mathbf{u}_j)$.

Interpolation "mesh-particles". Finite-difference schemes dealing with mesh functions $\{\tilde{\mathbf{q}}_\alpha\}$ are used in a lot of particle-in-cell methods for calculations at the Euler step. Therefore before the beginning of the next Lagrangian step it is necessary to interpolate the values of generalized force field $\{\tilde{\mathbf{U}}_\alpha\}$ or $\{\tilde{\mathbf{F}}_\alpha\}$ calculated in the Euler mesh into the locations of particles.

Recalculation usually is fulfilled according to following interpolation formulas

$$\mathbf{U}(\mathbf{u}_j) = \sum_{(\alpha)} \tilde{\mathbf{U}}_\alpha S(\mathbf{u}_\alpha - \mathbf{u}_j), \quad j = 1, \dots, N \quad (23)$$

where $S(\mathbf{u})$ is some interpolating function. From the condition of interpolation of the constant field $\tilde{\mathbf{U}}_\alpha = \mathbf{C}$ it is clear that interpolating function $S(\mathbf{u})$ must satisfy the requirement of norming

$$\sum_{(\alpha)} S(\mathbf{u}_\alpha - \mathbf{u}) = 1 \quad (24)$$

for all values of variables \mathbf{u} .

If the problem at the Lagrangian step is set up in the form (13), then the generalized force fields $\{\tilde{\mathbf{F}}_\alpha(\mathbf{r}, \mathbf{p})\}$ are calculated on the Euler mesh in the coordinate space Ω_r . Therefore formulas (23) take the form

$$\mathbf{F}(\mathbf{r}_j, \mathbf{p}_j) = \sum_{(\alpha)} \tilde{\mathbf{F}}_\alpha S(\mathbf{r}_\alpha - \mathbf{r}_j), \quad j = 1, \dots, N, \quad (25)$$

where the interpolating function $S(\mathbf{r})$ satisfy the requirement of norming analogous (24).

The examples of programs of interpolation of mesh functions into Lagrangian particles and inversely for orthogonal Cartesian mesh are given in Supplement B.

1.3. Model particles and their properties

1.3.1. General properties of the particles

Let us consider the main properties of model particles. In spite of the fact that in different applications they are called analogously to real objects such as ions, vortices, molecules their dynamics is very schematic. In particular, the form of particles and their orientation along coordinate axes don't change during movement.

In fact, if we do the substitution $\mathbf{v} = \mathbf{u} - \mathbf{u}_j(t)$ and consider the kernel in the coordinate system which moves together with particles, then we'll receive

$$R(\mathbf{u}, \mathbf{u}_j(t)) = R(\mathbf{v}, 0),$$

which means the unvariance of the form, and in the multidimensional case it means also the absence of rotations. The mutual intersection of particles of finite dimensions doesn't also influence their dynamics.

Let us suppose that the Lagrangian particles migrate in the domain Ω with the regular Euler mesh of nodes \mathbf{u}_α , formed by centers of cells $\{\omega_\alpha\}$. Approximative properties of the solution in the form (7) are defined considerably by the choice of the function of the kernel $R(\mathbf{u}, \mathbf{v})$. In particular, the form, the characteristic size of the kernel with respect to $|\omega_\alpha|$, and the smoothness of R play a very important role.

The using of singular kernels on the basis of Dirac delta functions is the limit case. Then formula (7) has the form

$$\mathbf{q} = \sum_{j=1}^N \mathbf{Q}_j \delta(\mathbf{u} - \mathbf{u}_j(t)) \quad (26)$$

Though such kernels are used in practice, however, if the average number of particles \bar{n} in one Euler cell ω_α is close to unit, then such choice leads to strong computational oscillations of the mesh solution. Indeed, the mesh kernel according to (22) in this case is

$$\tilde{R}(\mathbf{u}_\alpha, \mathbf{u}_j) = \frac{\xi_\alpha(\mathbf{u}_j)}{|\omega_\alpha|}, \quad (27)$$

where $\xi_\alpha(\mathbf{u})$ is a characteristic function (identifier) of the cell ω_α ,

$$\xi_\alpha(\mathbf{u}) = \begin{cases} 1, & \mathbf{u} \in \omega_\alpha, \\ 0, & \mathbf{u} \in \Omega \setminus \omega_\alpha \end{cases}$$

According to (21)

$$\tilde{\mathbf{q}}_\alpha = \sum_{j=1}^N \mathbf{Q}_j \frac{\xi_\alpha(\mathbf{u}_j)}{|\omega_\alpha|}, \quad (28)$$

and the value of the mesh solution in the node α is determined only by the particles which are in ω_α . This model is called NGP (nearest-grid-point) model. It is quite clear that with small numbers of particles in the cell the coming and leaving of every particle from it can change $\tilde{\mathbf{q}}_\alpha$ into the value of the same order. Such oscillations reflect the imperfection of the algorithm.

In order to avoid this the smoothed kernels are considered which are derived as a result of the action of some "smoothing" linear integral operator

$$R(\mathbf{u}, \mathbf{v}) = \int_{\Omega} d\mathbf{u}' G(\mathbf{u}, \mathbf{u}') \delta(\mathbf{u}' - \mathbf{v}) \equiv G(\mathbf{u}, \mathbf{v}).$$

The kernel of such a transformation evidently must satisfy all requirements imposed on the kernel R . Specifically, $G(\mathbf{u}, \mathbf{v})$ must be non-negative, symmetric, etc.

As functions $G(\mathbf{u}, \mathbf{v})$ one can consider degenerate kernels

$$G(\mathbf{u}, \mathbf{v}) = \sum_{m=1}^K \psi_m(\mathbf{u}) \psi_m(\mathbf{v}),$$

where $\{\psi_m(\mathbf{u})\}$ is a system of linearly independent functions in Ω . Simple functions are usually used here, that are connected with the partition of the domain Ω into Euler cells. For examples, such functions one can take from finite-element basis [33]. If we choose the piecewise constant functions $\psi_m(\mathbf{u}) = \xi_m(\mathbf{u})/\sqrt{|\omega_m|}$, where $|\omega_m|$ is Lebesgue measure of the cell, then the smoothed kernel accepts the following form

$$R(\mathbf{u}, \mathbf{v}) = \frac{\xi_\alpha(\mathbf{u})\xi_\alpha(\mathbf{v})}{|\omega_\alpha|}. \quad (29)$$

According to the classification used in the computational plasma physics [7, 22] this model is called particle-in-cell (PIC). As it will be shown below, such a choice of the kernel already allows to reduce significantly the fluctuations.

In general case the transition from distribution (7) to functions $\{\mathbf{q}_\alpha\}$ is brought out with the help of some projection operator P_ω . Then from Eq. (7) we receive

$$\mathbf{q}_\alpha = P_\omega \mathbf{q} = \sum_{j=1}^N \mathbf{Q}_j P_\omega R(\mathbf{u}, \mathbf{u}_j) = \sum_{j=1}^N \mathbf{Q}_j \bar{R}(\mathbf{u}_\alpha, \mathbf{u}_j). \quad (30)$$

Formula (30) determines the general form of interpolation from Lagrangian particles into nodes of Euler mesh. It's seen that the mesh kernel \bar{R} is the projection of the kernel $R(\mathbf{u}, \mathbf{v})$ into the space of the mesh functions. As a rule, in practice simple projectors are used, if it is possible. In particular, one can notice that the mesh kernel in (22) is determined by the projection operator calculated as averaging according to Steklov [34] over Euler cells ω_α .

But the use of the projector

$$P_\omega R(\mathbf{u}, \mathbf{u}_j) = \frac{\xi_\alpha(\mathbf{u}_j)}{|\omega_\alpha|} \quad (31)$$

gives us

$$\mathbf{q}_\alpha = \sum_{j=1}^N \mathbf{Q}_j \frac{\xi_\alpha(\mathbf{u}_j)}{|\omega_\alpha|}.$$

Though in this case the conservation law (20) is fulfilled

$$\sum_{(\alpha)} |\omega_\alpha| \mathbf{q}_\alpha = \sum_{j=1}^N \mathbf{Q}_j \sum_{(\alpha)} |\omega_\alpha| \frac{\xi_\alpha(\mathbf{u}_j)}{|\omega_\alpha|} = \sum_{j=1}^N \mathbf{Q}_j,$$

but the contribution into node \mathbf{u}_α of the Euler mesh is given only by the particles the centers of which are in the cell ω_α . Thus, the use of the projector (31) equivalently to returning on the singular particles (26).

If the action of generalized fields on the particles is calculated according to the formulas (19), then the interpolation "mesh-particles" also will depend on the choice of the model of the particles. Really, let us take the simplest fulfilment of the mesh force field $\{\mathbf{F}_\alpha\}$ with the help of piecewise constant interpolating function $S_\alpha(\mathbf{r}_\alpha - \mathbf{r}) = \xi_\alpha(\mathbf{r})$. Then for the singular kernel $R(\mathbf{r}, \mathbf{r}_j) = \delta(\mathbf{r} - \mathbf{r}_j)$ the field interpolation into the location of the particles gives us

$$\{\bar{\mathbf{F}}(\mathbf{r}_j)\} = \int_{\Omega_r} d\mathbf{r} \sum_{(\alpha)} \mathbf{F}_\alpha \xi_\alpha(\mathbf{r}) \delta(\mathbf{r} - \mathbf{r}_j) = \mathbf{F}_\beta \xi_\beta(\mathbf{r}_j),$$

where $\mathbf{r}_j \in \omega_\beta$. It means that the field acting upon the particle in a cell ω_β equals the value of mesh function \mathbf{F}_β in a node β . Evidently, such an interpolation coincides with NGP-model and has shortcomings inherent in it.

Subspace of generalized coordinates Ω_r is often splitted into cells $\{\omega_\alpha\}$, sides of which are parallel to coordinate planes of Cartesian coordinate system. If we take a piecewise constant kernel of the type (29)

$$R(\mathbf{r}, \mathbf{r}_j) = \frac{\xi_j(\mathbf{r})\xi_j(\mathbf{r}_j)}{|\omega_j|},$$

then for the generalized field acting upon j -particle, we obtain the formula

$$\{\bar{\mathbf{F}}(\mathbf{r}_j)\} = \int_{\Omega_r} d\mathbf{r} \sum_{(\alpha)} \mathbf{F}_\alpha \xi_\alpha(\mathbf{r}) \frac{\xi_j(\mathbf{r})\xi_j(\mathbf{r}_j)}{|\omega_j|} = \sum_\beta \mathbf{F}_\beta \frac{|\omega'_\beta|}{|\omega_j|},$$

where $\sum_{\{\beta\}} |\omega'_\beta| = |\omega_j|$, $|\omega'_\beta|$ is Lebesgue measure of the intersection of j -particle with cell ω_β . Summing up is carried out over all cells that have nonempty intersection with this particle. Evidently, the interpolation will be the more smooth the more cells will intersect with given particle. The latter naturally depends on the ratio of the dimensions of the cells $|\omega_\alpha|$ and the particles $|\omega_j|$.

1.3.2. Widespread models of the particles

In order to do our presentation more concrete let us consider the simplest case when $q(u) \equiv \rho(x)$ is a scalar function, $x \in [0, L]$ and the Euler mesh $\{x_\alpha\}$ is formed by the centers of cells with spacing h . Having simplified our computation in this way, we can closely approach the models of the particles used in practical calculations. The latter are received from the considered below as evident generalizations on two- and three-dimensional cases.

Let the one-dimensional analog of distribution (7) which we'll call as some density (of the charge, mass) has the form

$$\rho(x) = \sum_{j=1}^N m_j R(x - x_j). \quad (33)$$

The norming condition (10) turns into the relation

$$\int_0^L dx R(x - x_j) = 1.$$

We give two examples of one-dimensional kernels [22]. The analog of the kernel (29) is the function

$$R(x) = \begin{cases} \frac{1}{2\Delta}, & |x| \leq \Delta, \\ 0, & |x| > \Delta. \end{cases} \quad (34)$$

The kernel in the form (34) describes the particle of the width 2Δ with uniform density distribution. The density calculated according to formula (33) will be in this case piecewise constant function. At $\Delta = \frac{h}{2}$ this kernel corresponds to the one-dimensional PIC-model (see (29)).

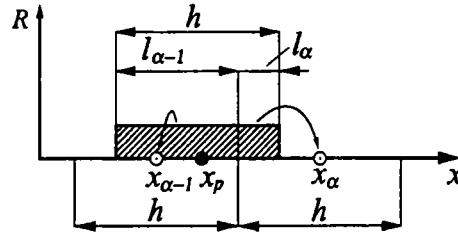


Fig. 1. The initial kernel of the one-dimensional PIC model. The scheme of the interpolation of the carried feature upon the Euler mesh.

The other simple kernel is

$$R(x) = \begin{cases} \frac{1}{\Delta} \left(1 - \frac{|x|}{\Delta}\right), & |x| \leq \Delta, \\ 0, & |x| > \Delta. \end{cases} \quad (35)$$

In the case (35) the particle has width 2Δ and linear density distribution inside it. The particle has maximum density $1/\Delta$ in the center, at the edges its density equals zero. Restored according to formula (33) the density is continuous piecewise-linear function. We should emphasize that in general case the characteristic dimension of the particle Δ is not connected with the length h (Lebesgue measure) of cell (compare with (29)). In computational plasma physics this model is known as "cloud-in-cell" (CIC).

The conservation law (20) has the following form here

$$\sum_{j=1}^N m_j = h \sum_{\alpha} \rho_{\alpha}.$$

It is satisfied if values ρ_{α} in the mesh nodes will be determined by the formula

$$\rho_{\alpha} = \frac{1}{h} \int_{x_{\alpha} - \frac{h}{2}}^{x_{\alpha} + \frac{h}{2}} \rho(x) dx,$$

or in the form Eq. (23)

$$\rho_{\alpha} = \sum_{j=1}^N m_j \bar{R}(x_{\alpha} - x_j), \quad (36)$$

where the mesh kernel (compare with (24)) is

$$\bar{R}(x_\alpha - x_j) = \frac{1}{h} \int_{x_\alpha - \frac{h}{2}}^{x_\alpha + \frac{h}{2}} R(x - x_j) dx.$$

Let us note that for $\bar{R}(x)$ the norming condition is also fulfilled

$$\int_0^L dx \bar{R}(x) = 1.$$

If we sum up the contributions of one particle with number j in all nodes of the mesh then we obtain

$$\begin{aligned} h \sum_{(\alpha)} \rho_\alpha &= h \sum_{\alpha} m_j \bar{R}(x_\alpha - x_j) = \\ &= hm_j \sum_{(\alpha)} \frac{1}{h} \int_{x_\alpha - \frac{h}{2}}^{x_\alpha + \frac{h}{2}} R(x' - x_j) dx' = \\ &= m_j \int_0^L R(x' - x_j) dx' = m_j. \end{aligned}$$

That means that the sum of contributions into the nodes of the mesh equals the total feature carried by the particle.

Let us calculate the mesh kernels for some functions $R(x)$ in one-dimensional case. For the point-like particle with the singular kernel $R(x - x_p) = \delta(x - x_p)$ we derive

$$\begin{aligned} \bar{R}(x_\alpha - x_p) &= \frac{1}{h} \int_{x_\alpha - \frac{h}{2}}^{x_\alpha + \frac{h}{2}} \delta(x - x_p) dx = \\ &= \frac{1}{h} \int_{x_\alpha - x_p - \frac{h}{2}}^{x_\alpha - x_p + \frac{h}{2}} \delta(y) dy \begin{cases} \frac{1}{h}, & |x_\alpha - x_p| \leq \frac{h}{2}, \\ 0, & |x_\alpha - x_p| > \frac{h}{2} \end{cases} \quad (37) \end{aligned}$$

because the integral differs from zero in the last equality only if the point $y = 0$ belongs to the integration interval. It is seen that this kernel is similar to the mesh kernel (28) in multidimensional case and represents the one-dimensional NGP-model (nearest grid point model) [35, 36]. The meaning of it can be understood from the formula (28).

For the PIC-model from the kernel (34) at $\Delta = h/2$ the mesh kernel is derived as follows

$$\bar{R}(x_\alpha - x_p) = \frac{1}{h} \int_{x_\alpha - \frac{h}{2}}^{x_\alpha + \frac{h}{2}} R(x - x_p) dx = \frac{1}{h} \int_{x_\alpha - x_p - \frac{h}{2}}^{x_\alpha - x_p + \frac{h}{2}} R(y) dy.$$

From (34) it directly seen that the last integral at $|x_\alpha - x_p| > h$ equals zero because the integrand vanishes in the corresponding limits of the

integration. As one can see from Fig. 1 the particle with the center in point x_p in this case as a whole is out of the cell with center x_α and it doesn't contribute into this node. At $|x_\alpha - x_p| \leq h$ taking into account (34) it may be written

$$\bar{R}(x_\alpha - x_p) = \frac{1}{h} \int_{y_i}^{y_s} \frac{dy}{h} = \frac{1}{h^2} (y_s - y_i),$$

where $y_s = (x_\alpha - x_p)H(x_p - x_\alpha) + \frac{h}{2}$, $y_i = (x_\alpha - x_p)H(x_\alpha - x_p) - \frac{h}{2}$, $H(x)$ is the Heaviside function, $H(x) = 1$ for $x > 0$, $H(x) = 0$ for $x < 0$. Finally we deduce

$$\bar{R}(x_\alpha - x_p) = \begin{cases} \frac{1}{h} \left(1 - \frac{|x_\alpha - x_p|}{h} \right), & |x_\alpha - x_p| \leq h, \\ 0, & |x_\alpha - x_p| > h. \end{cases} \quad (38)$$

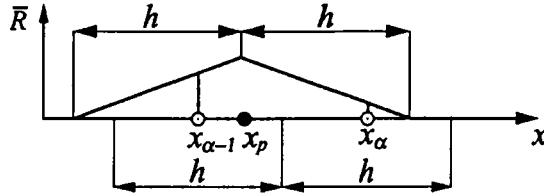


Fig. 2. The mesh kernel function of the one-dimensional PIC-model. Values $\bar{R}(x - x_p)$ in the nodes of the mesh equal the part of the particle feature interpolated into this node.

In the formula (36) the summation is carried out on all the particles. However, the contribution into the mesh density in the node x_α is made only by the particles which are in certain neighborhood of this node. The geometry of the neighborhood is determined by the dimensions and form of the model particle. As it is seen from (37) the contribution into the mesh density in NGP-model is given only by the particles which are at the distance less than $h/2$ of the node. Thus, every particle contributes into one nearest node. In PIC-model (38) the contribution into considered node is given by the particles which are at the distance less than h . It means that every particle, as it is shown in the scheme Fig. 1, can contribute into the two nearest nodes. According to the formula (34) at $\Delta = h/2$ mass (charge) m_p of the particle with dimension h is uniformly distributed in it proportionally with the density $\rho_p = m_p/h$. Let the center of the particle be located in point x_p of the segment $[0, L]$ between neighbouring nodes $x_{\alpha-1}, x_\alpha$, $x_\alpha - x_{\alpha-1} = h$. Then its contributions into the mesh density in these nodes according to (33) and (38) are calculated as

$$\begin{aligned}\rho_{\alpha-1} &= \frac{m_p}{h} \left(1 - \frac{x_p - x_{\alpha-1}}{h} \right) = \frac{m_p}{h} \cdot \frac{x_\alpha - x_p}{h}, \\ \rho_\alpha &= \frac{m_p}{h} \left(1 - \frac{x_\alpha - x_p}{h} \right) = \frac{m_p}{h} \cdot \frac{x_p - x_\alpha}{h},\end{aligned}\tag{39}$$

That means that the uniform density m_p/h is distributed by the *inverse linear interpolation* between the two nearest nodes of a mesh.

The geometric interpretation of this interpolation model is still more clear. As one can see from Fig. 1 and the formula (39), the following relations take place

$$\begin{aligned}\frac{m_p}{h} \cdot \frac{l_{\alpha-1}}{h} &= \frac{m_p}{h} \cdot \frac{1}{h} \left[\left(x_{\alpha-1} + \frac{h}{2} \right) - \left(x_p - \frac{h}{2} \right) \right] = \\ &= \frac{m_p}{h} \cdot \frac{x_\alpha - x_p}{h} = \rho_{\alpha-1}, \\ \frac{m_p}{h} \cdot \frac{l_\alpha}{h} &= \frac{m_p}{h} \cdot \frac{1}{h} \left[\left(x_p + \frac{h}{2} \right) - \left(x_{\alpha-1} + \frac{h}{2} \right) \right] = \\ &= \frac{m_p}{h} \cdot \frac{x_p - x_{\alpha-1}}{h} = \rho_\alpha,\end{aligned}$$

Thus, the uniform density is divided between the nodes, i.e. centers of the neighbouring cells proportionally to parts $l_{\alpha-1}/h$, l_α/h of the length (measure) of the particle belonging to the corresponding cells.

It can be noted that the using of initial kernels for calculating of the contribution of the p -particle into the node x_α needs the fulfillment of some operations, such as the determination of the cell in which the particle locates (NGP-model), the inverse interpolation (PIC-model), etc. On the contrary, the corresponding mesh kernels give the ready algebraic relations for this purpose. For one-dimensional PIC-model it is seen from the comparison Fig. 1, 2.

In the two-dimensional case the particle in the PIC-model has a form of a rectangle with sides h_x , h_y which are parallel to coordinate axes (x, y) . Its Lebesgue measure is the area $s_p = h_x h_y$ on which the feature carried by the particle is distributed uniformly with the density $\rho_p = m_p/s_p$. The corresponding mesh kernel analogous to the mesh kernel Eq. (38) has the form

$$\bar{R}(x_\alpha - x_p, y_\beta - y_p) = \begin{cases} \frac{1}{h_x h_y} \left(1 - \frac{|x_\alpha - x_p|}{h_x} \right) \left(1 - \frac{|y_\beta - y_p|}{h_y} \right), \\ |x_\alpha - x_p| \leq h_x \text{ and } |y_\beta - y_p| \leq h_y, \\ 0, \quad |x_\alpha - x_p| > h_x \text{ or } |y_\beta - y_p| > h_y, \end{cases}$$

That means that it is the product of two one-dimensional kernels in the form of Eq. (38). In the two-dimensional PIC-model the density of every

particle is distributed with the help of the *inverse bilinear interpolation* among the four nearest nodes of the mesh.

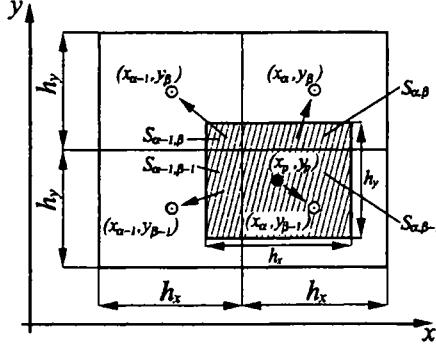


Fig. 3. The initial kernel of the two-dimensional PIC-model. The interpolation scheme of the carried feature on the Euler mesh.

Really, let the particle center be in the point with coordinates (x_p, y_p) and $x_{\alpha-1} < x_p < x_\alpha, y_{\beta-1} < y_p < y_\beta$. Then, for example, for the contribution into the node (α, β) using the formula from Eq. (39) along every Cartesian coordinate one can deduce

$$\rho_{\alpha\beta} = \rho_p \frac{x_p - x_{\alpha-1}}{h_x} \cdot \frac{y_p - y_{\beta-1}}{h_y}.$$

Respectively, in the geometric interpretation the contribution of the particle into every node from the four ones of the nearest neighborhood is proportional to the area part overlapping this cell. In particular, from Fig. 3 taking into account the previous formula it follows

$$\begin{aligned} \rho_p \frac{s_{\alpha\beta}}{s_p} &= \rho_p \left[\left(x_p + \frac{h_x}{2} \right) - \left(x_{\alpha-1} + \frac{h_x}{2} \right) \right] \left[\left(y_p + \frac{h_y}{2} \right) - \left(y_{\beta-1} + \frac{h_y}{2} \right) \right] = \\ &= \rho_p \frac{x_p - x_{\alpha-1}}{h_x} \cdot \frac{y_p - y_{\beta-1}}{h_y} = \rho_{\alpha\beta}. \end{aligned}$$

Contributions into the other nodes are calculated similarly. The subroutine of the interpolation into the nodes of the Euler mesh for the two-dimensional PIC-model is presented in Supplement B2.

The particles in the two-dimensional CIC-model [37] represent rectangles with the sides $2\Delta_x$ and $2\Delta_y$, which are parallel to the Cartesian coordinate axes that in general case are not connected by any relations with space mesh steps h_x, h_y . The density of carried feature in the rectangle is assumed as uniform, the particle location is determined by the coordinates of its center. The contribution of the particle feature into the mesh nodes is proportional to the part of the particle area occurring in the cell surrounding the corresponding node (compare with

Fig. 2). If we fix the mesh steps h_x , h_y , then the number of the nodes into which the particle contributes its feature depends in general case on the particle dimensions $2\Delta_x$, $2\Delta_y$ and its location.

The mentioned models of the particles are among the most commonly used in the methods of the particles considered in the following chapters. They can be referred to single class because their mesh kernels are generated by one and the same piecewise constant initial kernel of the type Eq. (29).

The kernel CIC with arbitrary dimensions $2\Delta_x$, $2\Delta_y$ obviously represents the general case. From it at $\Delta_x = h_x/2$, $\Delta_y = h_y/2$ the PIC-model follows, and at $\Delta_x, \Delta_y \rightarrow 0$ — the NGP-model is derived .

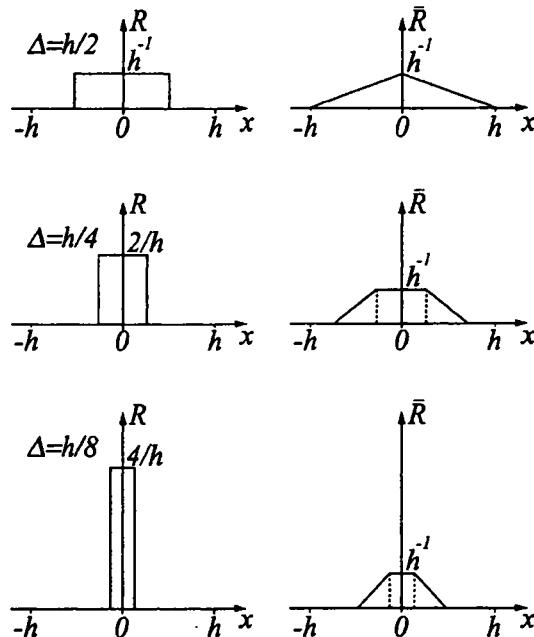


Fig. 4. The transformation of the kernels of the one-dimensional CIC-model depending on the particle dimension.

The dependence of the form of the initial and mesh kernels of the one-dimensional model CIC on the relation of the particle dimension 2Δ to the mesh step is shown in Fig. 4. One can observe how with the decreasing of the particle dimension the approximation approach towards the NGP-model takes place.

As it has been noted above, the NGP-model where every particle contributes into a single node leads to strong fluctuations of the mesh functions. The PIC-model in the n -dimensional case smoothes these fluctuations distributing the carried feature among 2^n nodes nearest to its center. The CIC-model allows to increase the number of the nodes involved into the interpolation.

1.3.3. The improvement of the interpolation smoothness

One can indicate the following possibility of systematic construction of mesh kernels with the aim of the optimization of the interpolation procedure. Let us turn to the one-dimensional case again. Let some model particle contributes into p nodes. We optimize the particle form in such a way that under movement of every solitary particle in the mesh domain the density fluctuations would be minimum.

The contribution into the density in the node x_α of the mesh from the particle with number j equals

$$\rho_\alpha = m_j \bar{R}(x_\alpha - x_j). \quad (40)$$

In order to determine the density in any point of the phase space we need to use some interpolating function $S(x)$ (compare with Eq. (32)).

$$\rho(x) = \sum_{(\alpha)} \rho_\alpha S(x - x_\alpha). \quad (41)$$

Let the function $S(x)$ satisfy the condition

$$\sum_{(\alpha)} S(x - x_\alpha) = 1$$

for all values of x . In order that the mesh function ρ_α symmetrical relatively some node could give symmetrical restored function $\rho(x)$ it is necessary that $S(x)$ should be an even function: $S(x) = S(-x)$. Uniting the formulas (40) and (41) we receive the density in any point of the space which is given by the particle with number j

$$\rho(x) = m_j \sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) S(x - x_\alpha). \quad (42)$$

The limits of summation in Eq. (42) mean that the particle contributes only into p of the nearest nodes with numbers i_1, \dots, i_p , where $\bar{R}(x_\alpha - x_j) \neq 0$. The exact density in point x (see Eq. (33)) generated by j -particle depends only on the distance $x - x_j$, whereas the approximate density depends both on $x - x_j$, and on $(x_\alpha - x_j)$, i.e., it depends on the location of j -particle relatively to the mesh nodes. Let us choose the particle dimension and density distribution inside the particle in such a way that to reduce the dependence of the density restored by the formula (42) on the location relatively to the mesh nodes.

Assuming that the function $S(x)$ under consideration is infinitely differentiated, let us substitute into Eq. (42) its expansion in the Taylor series in the point $(x - x_j)$ in powers of increment $\Delta = x_j - x_\alpha$

$$\begin{aligned} \rho(x) = m_j \sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) S(x - x_j) + \\ + m_j \sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) \sum_{n=1}^{\infty} \frac{\Delta^n}{n!} \frac{d^n}{dx^n} S(x - x_\alpha) \Big|_{x=x_j}. \end{aligned} \quad (43)$$

The property following from the conservation law gives the first restriction on the function \bar{R}

$$\sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) = \frac{1}{h}. \quad (44)$$

If the particle contributes into the density in one node only ($p = 1$), then the relation (44) fully determines the mesh kernel which coincides with the NGP-model Eq. (37). If $p > 1$, then in Eq. (43) additional possibilities appear for weakening of the function $\rho(x)$ dependence on a mesh. For this should be required that terms of higher powers of Δ in the expansion (43) didn't successively depend on the coordinates of the mesh nodes. The latter requirement will be realized if

$$\sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) (x_\alpha - x_j)^n = \text{const}, \quad (45)$$

for constantly increasing values n as the number of mesh nodes p participating in the distribution of density increases. As far as $S(x)$ is an even function and $\bar{R}(x)$ is nonnegative, it is necessary that additional restrictions should be introduced into Eqs. (45). Since \bar{R} is a nonnegative function, then the conditions replacing Eq. (45) have the form

$$\sum_{\alpha=i_1}^{i_p} \bar{R}(x_\alpha - x_j) (x_\alpha - x_j)^n = \begin{cases} 0, & n - \text{odd}, \\ \text{const}, & n - \text{even}. \end{cases} \quad (46)$$

For $p = 2$ it can satisfy only the condition Eq. (44) and condition Eq. (46) for $n = 1$. In this case, the higher remainder term in the formula (27) has the order $O(|x_\alpha - x_j|^2)$. The conditions look as follows

$$\begin{cases} \bar{R}_1 + \bar{R}_2 = \frac{1}{h}, \\ \bar{R}_1 x_{i_1} + \bar{R}_2 x_{i_2} = x_j (\bar{R}_1 + \bar{R}_2). \end{cases} \quad (47)$$

Here $\bar{R}_1 = \bar{R}(x_{i_1} - x_j)$, $\bar{R}_2 = \bar{R}(x_{i_2} - x_j)$, $x_{i_1} \leq x_j \leq x_{i_2}$, and x_{i_1} and x_{i_2} are the neighbouring nodes of the mesh, $x_{i_2} - x_{i_1} = h$. Solving the system Eq. (47), we get

$$\bar{R}(x_{i_1} - x_j) = \frac{1}{h} \left(1 - \frac{x_j - x_{i_1}}{h} \right),$$

$$\bar{R}(x_{i_2} - x_j) = \frac{1}{h} \left(1 - \frac{x_{i_2} - x_j}{h} \right).$$

The union of these formulas gives us

$$\bar{R}(x_\alpha - x_j) = \begin{cases} \frac{1}{h} \left(1 - \frac{|x_\alpha - x_j|}{h} \right), & |x_\alpha - x_j| < h, \\ 0, & |x_\alpha - x_j| \geq h, \end{cases}$$

which coincides with the formula (38).

Increasing p up to three we can satisfy the conditions Eq. (46) up to $n = 2$. We obtain a system in the form

$$\begin{cases} \bar{R}_1 + \bar{R}_2 + \bar{R}_3 = \frac{1}{h}, \\ \bar{R}_1 x_{\alpha-1} + \bar{R}_2 x_\alpha + \bar{R}_3 x_{\alpha+1} = \frac{x_j}{h}, \\ \bar{R}_1 x_{\alpha-1}^2 + \bar{R}_2 x_\alpha^2 + \bar{R}_3 x_{\alpha+1}^2 = \frac{x_j^2}{h} + C, \end{cases} \quad (48)$$

where $\bar{R}_1 = \bar{R}(x_{\alpha-1} - x_j)$, $\bar{R}_2 = \bar{R}(x_\alpha - x_j)$, $\bar{R}_3 = \bar{R}(x_{\alpha+1} - x_j)$, $x_\alpha - h/2 \leq x_j \leq x_\alpha + h/2$, C is some constant. Solving of this system of Eqs. (48) gives us

$$\begin{cases} \bar{R}_1 = \frac{1}{2h^3} (x_\alpha - x_j)(x_{\alpha+1} - x_j) + \frac{C}{2h^2}, \\ \bar{R}_2 = \frac{1}{h^3} (x_{\alpha+1} - x_j)(x_j - x_{\alpha-1}) - \frac{C}{h^2}, \\ \bar{R}_3 = \frac{1}{2h^3} (x_j - x_\alpha)(x_j - x_{\alpha-1}) + \frac{C}{2h^2}. \end{cases}$$

Rewriting these formulas in the form of the mesh kernel we deduce

$$\bar{R}(x) = \begin{cases} \frac{1}{2h^3} (h+x)(2h+x) + \frac{C}{2h^2}, & -\frac{3}{2}h \leq x \leq -\frac{h}{2}, \\ \frac{1}{h} - \frac{1}{h^3} (x+hC), & -\frac{h}{2} \leq x \leq \frac{h}{2}, \\ \frac{1}{2h^3} (h-x)(2h-x) + \frac{C}{2h^2}, & \frac{h}{2} \leq x \leq \frac{3}{2}h. \end{cases}$$

Demanding that the function $\bar{R}(x)$ in the whole definition domain should be nonnegative we get the restriction $C \geq h/4$. The additional condition of the continuity of the function $\bar{R}(x)$ gives us the final value $C = h/4$. In this case, the function $\bar{R}(x)$ is not only continuous but also it has the continuous first derivative. As a result $\bar{R}(x)$ has the form:

$$\bar{R}(x) = \begin{cases} \frac{1}{h} \left(\frac{3}{4} - \frac{x^2}{h^2} \right), & |x| \leq \frac{h}{2}, \\ \frac{1}{2h} \left(\frac{3}{2} - \frac{|x|}{h} \right)^2, & \frac{h}{2} \leq |x| \leq \frac{3}{2}h, \\ 0, & |x| > \frac{3}{2}h. \end{cases} \quad (49)$$

This mesh kernel could have been found from the formula (24) by integration of the expression (35). It is seen that under increasing p the optimization process allows to derive more and more smooth mesh kernels. But at the same time the complexity of them increases.

1.4. Errors of the particle-in-cell schemes

The sources of errors in such complicated algorithms as particle-in-cell methods are rather various. In separate applications of these methods, for example, to the problems of plasma physics (see Chapter 5) the great attention is paid to the considering of errors. However, up to now there are almost no universal results here. Therefore in practice we should often restrict ourselves to the estimation of the total error according to the results of test calculations.

The fluctuations is one of the sources of errors in the particle method. Fluctuations of the feature density carried by particles are connected with comparatively small number of model particles used in calculations. If the particle distribution modeled depends on several variables, it additionally increases fluctuations, and the only way to reduce them is to increase the number of particles [7, 22, 38].

The other source of error appearing is the presence of the space mesh in which in the Euler step the values of generalized force fields are calculated. This group includes errors of the discrete approximation of equations for the "fields", errors of the interpolation of "forces" into the location of particles and some others.

1.4.1. The self-force effect

The specific error of the particle method connected with the mesh is the existence of self-force. This is a "force" that is created on a mesh by particle itself and acts on particle in turn. In accordance with the papers [39, 40] let us consider the reasons for the appearing of this effect on the example of electrostatic problem for solitary particle.

Let a particle with electric charge q is located between two nodes of the space mesh with numbers 1 and 2 situated along the axis x with step h and $r < h/2$ is the distance from the particle to node 1. Here the force is the electric field \mathbf{E} the potential of which is determined by the Poisson equation

$$\Delta\varphi = -4\pi\rho,$$

where ρ is the density of the spatial charge.

If we use the well-known NGP- and PIC-model with the mesh kernels (37) and (38) respectively, then the charge q of the particle is attributed to the cell with number 1 in the NGP-model and is distributed between cells 1 and 2 inversely proportionally to the distance to the nodes in the PIC-model.

Thus, the charge in the points 1 and 2 equals

$$q_1 = \begin{cases} q, & (\text{NGP}) \\ q \frac{h-r}{h}, & (\text{PIC}) \end{cases} \quad q_2 = \begin{cases} 0, & (\text{NGP}) \\ q \frac{r}{h}. & (\text{PIC}) \end{cases}$$

The potential on the line between points 1 and 2 where the particle locates in the case of NGP-model can be written in the form

$$\varphi = -\frac{Cq}{r},$$

where C is some constant. In the case of PIC-model the potential equals

$$\varphi = -\frac{C\rho_1}{r} + \frac{C\rho_2}{h-r}.$$

The force acting on the particle equals $F = qE = -q \frac{\partial \varphi}{\partial r}$. In the first case it equals

$$F = C \frac{q^2}{r^2},$$

and in the second case it correspondingly equals

$$F = \frac{Cq^2}{h} \left(\frac{h-r}{r^2} - \frac{r}{(h-r)^2} \right).$$

Let us emphasize the fact that exact electrostatic formulas are used here for the values of the potential and of the force acting on the particle.

It follows from these formulas that in the NGP-model the force acting on the particle always exists, and this force repels it from the nearest node. When the particle is in the middle of the cell, this force is minimum, in this case it changes its sign with a jump while intersecting the middle of the cell.

In the PIC-model such a force exists also. The force equals zero in the middle of the cell, but with bringing nearer the node it increases.

Thus, we have shown that despite of the Poisson equation is solved exactly, and electric field is also calculated exactly some non-physical force is obtained which repels the particle from the mesh nodes. This force connected with presence of the mesh is called "self-force". In Fig. 5 curve 2 represents the trajectory of a solitary particle moving in space where there are no other charges. Self-action becomes apparent in the twist of the trajectory which must be rectilinear. From the formulas it is seen that the value of the force decreases quadratically with particle charge decreasing. Under increasing of the number of model particles N the influence of this self-force diminishes because of the charge conservation law $q \sim \frac{1}{N}$ and principal action on particle is produced by other particles.

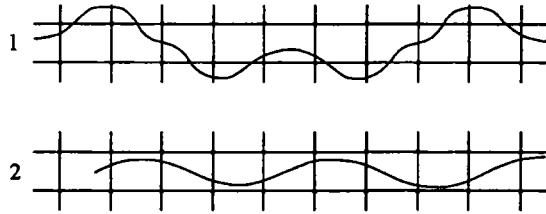


Fig. 5. The distortion of the trajectory of model particles under the influence of self-force.

1 — gravitating mass; 2 — electric charge.

Now let us consider the appearing of self-force when fields are being calculated on the basis of finite-difference schemes. Let us return to the one-dimensional electrostatic problem. Let the uniform mesh with step h be introduced into the definition domain $[0, L]$. I is the number of the mesh cells, i.e., $L = h \cdot I$. There are particles with identical charge in this domain. These particles after restoring of charge density in the mesh nodes according to one or another interpolation model give the mesh density function $\{\rho_\alpha\}_{\alpha=1}^{I-1}$. For the determining of the potential in the mesh nodes it is necessary to solve the difference Poisson equation

$$\frac{\varphi_{\alpha+1} - 2\varphi_\alpha + \varphi_{\alpha-1}}{h^2} = \rho_\alpha.$$

Let on the left boundary $\varphi_0 = A$ and on the right boundary $\varphi_I = B$. Then the difference solution can be written analytically in the form

$$\begin{aligned} \varphi_\alpha &= \frac{I-\alpha}{I}A + \frac{\alpha}{I}B - \\ &- \frac{h^2}{I} \left[(I-\alpha) \sum_{k=1}^{\alpha} k \rho_k + \alpha \sum_{k=\alpha+1}^{I-1} (I-k) \rho_k \right]. \end{aligned} \quad (50)$$

Let us consider solitary particle with coordinate x located between nodes $x_{\alpha-1}$ and x_α . Then $\delta = (x - x_{\alpha-1})/h$ is a relative distance between the particle and the node $x_{\alpha-1}$. In the PIC-model the particle under the calculation of the density in the mesh nodes contributes into the node $x_{\alpha-1}$ a part of the charge that equals $(1 - \delta)q$ and into the node x_α corresponding part δq (the inverse linear interpolation). By division on the mesh width h we obtain the charge density in the nodes:

$$\rho_{\alpha-1} = (1 - \delta) \frac{q}{h}, \quad \rho_\alpha = \delta \frac{q}{h}.$$

In other nodes the density equals zero. Substituting these densities into Eq. (50), we find the potential value

$$\begin{aligned}\varphi_{\alpha-1} &= \frac{I-\alpha+1}{I}A + \frac{\alpha-1}{I}B - \\ &- \frac{h^2}{I}(\alpha-1)[(I-\alpha+1)\rho_{\alpha-1} + (I-\alpha)\rho_\alpha]. \\ \varphi_\alpha &= \frac{I-\alpha}{I}A + \frac{\alpha}{I}B - \frac{h^2}{I}(I-\alpha)[(\alpha-1)\rho_{\alpha-1} + \alpha\rho_\alpha].\end{aligned}$$

The strength of electric field in the segment $[x_{\alpha-1}, x_\alpha]$ one can determine in a piecewise-constant manner according to the formula

$$E_h = -\frac{\varphi_\alpha - \varphi_{\alpha-1}}{h}.$$

It equals

$$\begin{aligned}E_h &= \frac{A-B}{hI} + \frac{h}{I}[(I-\alpha)\rho_\alpha - (\alpha-1)\rho_{\alpha-1}] = \\ &= \frac{A-B}{L} + \frac{q}{I}[\alpha-1-\delta(I-1)].\end{aligned}$$

The first term in the right side gives the external electric field determined by the boundary conditions. The second term shows us that the force acting upon the particle depends on the particle location in the solution domain and on the particle location relatively to the nearest nodes. Let us remove the influence of the boundary conditions setting $A = B = 0$. In order to exclude the influence of the particle location in the domain, let us consider the odd number of the cells I and place the particle into the central cell $\alpha = (I+1)/2$. Then we get

$$F_h = qE_h = -\frac{q^2}{2}\left(1 - \frac{1}{I}\right)(1-2\delta). \quad (51)$$

This is the self-force. At $\delta = 1/2$ when the particle is in the middle of the cell, it equals zero, and with approaching towards the mesh nodes ($\delta \rightarrow 0$ or $\delta \rightarrow 1$) the force strives to its maximum value that equals approximately $q^2/2$. From the formula one can see that it is the repulsing force.

It can be noted that self-action is rather a universal effect for particle-in-cell methods. For example, the same effect appears in gravitational problems. Really changing in the above reasonings the potential of the electric field by the potential of the gravitational field and the electric charge by a gravitating mass, we can derive the same formulas excepting the fact that the self-force will be the attraction. The corresponding distortion of the calculated trajectory of the gravitating particle is demonstrated by the curve 1 in Fig. 5.

The self-action can take place also in problems of vortex dynamics (see Chapter 3) and in close to them problems of the dynamics of the magnetized plasma in the "guiding center" approximation solved by particle methods.

The analysis carried out above allows to suggest a new model of particles [39, 40], which essentially decreases the self-force. Let us turn to electrostatic problem again. The one-dimensional initial kernel of this model can be described by the formula

$$R(x) = \begin{cases} 0, & |x| < \frac{h}{2}, \\ \frac{1}{2h}, & \frac{h}{2} \leq |x| \leq \frac{3}{2}h, \\ 0, & |x| > \frac{3}{2}h. \end{cases} \quad (52)$$

The corresponding mesh kernel is

$$\bar{R}(x) = \begin{cases} \frac{|x|}{2h^2}, & |x| \leq h, \\ \frac{1}{h} - \frac{|x|}{2h^2}, & h \leq |x| \leq 2h, \\ 0, & |x| \geq 2h. \end{cases} \quad (53)$$

The shape of a kernel of a new particle and the scheme of interpolation of the charge into the nodes of the Euler mesh determining the self-force are shown in Fig. 6a,b.

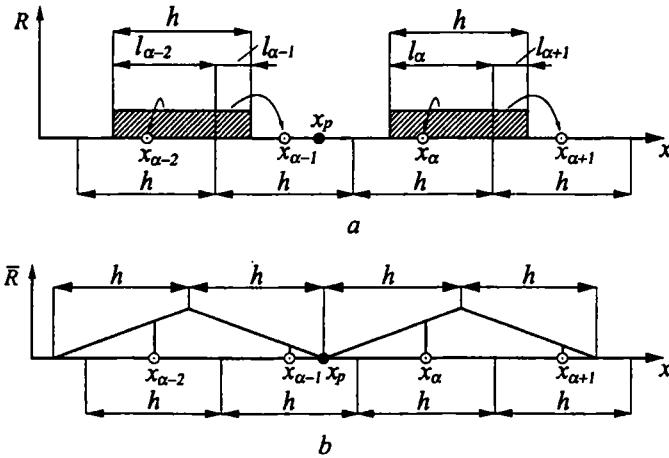


Fig. 6. The model of the hollow particle with reduced self-force. The scheme of interpolation into the nodes of the Euler mesh.
a — initial kernel; b — mesh kernel.

Let us consider the properties of a new particle which is a pair of two particles of the PIC-model divided by the cavity with the width h (see Fig. 6a). Because that the particle with the center x_p located between the nodes $x_{\alpha-1}$ and x_α gives contributions into four nodes:

$$\rho_{\alpha-2} = \rho_\alpha = (1 - \delta) \frac{q}{2h},$$

$$\rho_{\alpha-1} = \rho_{\alpha+1} = \delta \frac{q}{2h}.$$

For this particle the electric field equals

$$\begin{aligned} E_x &= \frac{A - B}{L} - \frac{h}{I}(2\alpha - I - 1)(\rho_\alpha + \rho_{\alpha-1}) + \frac{h}{I}(\rho_\alpha - \rho_{\alpha-1}) = \\ &= \frac{A - B}{L} - \frac{q}{2I}(2\alpha - I - 1) + \frac{q}{2I}(1 - 2\delta). \end{aligned}$$

By such a way the self-forces of particle interaction with each of the nodes are partly compensated. As above neglecting by boundary conditions and placing the particle into central cell ($\alpha = (I + 1)/2$) we obtain

$$F_h = qE_h = \frac{q^2}{2I}(1 - 2\delta). \quad (54)$$

Let us compare the formulas (51) and (54). We see that the value of self-force in the last case is I times less than it is in the first one. Besides, the self-force has an opposite sign. It allows to create a kernel for which a self-force equals zero. This kernel looks as follows

$$\bar{R}(x) = \frac{1}{I}R_1(x) + \left(1 - \frac{1}{I}\right)\bar{R}_2(x),$$

where $\bar{R}_1(x)$ is the mesh kernel of the PIC-method, $\bar{R}_2(x)$ is the kernel Eq. (53).

In spite of the comparative complexity of the new kernel the number of operations under calculating of charge density increases negligibly because the calculations are fulfilled in two steps. First we calculate the intermediate density of the charge $\tilde{\rho}_\alpha$ according to the usual PIC-model with the mesh kernel Eq. (38). Then we find a final value

$$\rho_\alpha = \frac{1}{2}(\tilde{\rho}_{\alpha+1} + \tilde{\rho}_{\alpha-1}).$$

It should be noted that such an analysis can be taken only in one-dimensional case where the solution of the Poisson equation is easily constructed in explicit form. In 2D- and 3D-cases the dependence of self-force on the kernel and on the mesh are more complicated. And the mesh itself can be non-rectangular and the weights of kernels R_1 and R_2 can differ for different nodes of the mesh. However, in this case also the value of self-force for a new kernel is less than for the known kernel of a PIC-model.

1.4.2. The influence of the mesh periodicity

The regular mesh in subspace of generalized coordinates Ω_r is usually a necessary attribute of particles-in-cells methods. However its regularity brings a periodicity on space in the problem which is unnatural for the simulated phenomenon. This parasitic periodicity is also a source of computational errors. For the first time this kind of errors was observed in applications of the particle-in-cell method to the modeling of plasma dynamics. However, one can be convinced that under certain conditions such an error has the universal character.

Following [38] let us use the Fourier method for analysis and again restrict ourselves to the one-dimensional case. The direct and inverse Fourier transform are determined by formulas:

$$\begin{aligned} g(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} dk \exp(ikx)\hat{g}(k), \\ \hat{g}(k) &= \int_{-\infty}^{\infty} dx \exp(-ikx)g(x), \end{aligned} \quad (55)$$

where i is an imaginary unit. We use formulas of Fourier transform for representing mesh functions determined in the segment $[-\pi, \pi]$ on the regular mesh of nodes $\{x_\alpha\}$, $\alpha = 0, \pm 1, \pm 2, \dots$

$$\begin{aligned} g_\alpha &= \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} dk \hat{g}(k) \exp(ikx_\alpha), \\ \hat{g}(k) &= h \sum_{(\alpha)} g_\alpha \exp(-ikx_\alpha), \end{aligned} \quad (56)$$

Here g_α is a mesh function, $\hat{g}(k)$ is its Fourier transformation.

Let us consider the mesh function of the density of the feature q carried by particles. It can be an electric charge, inert or gravitating mass or some other. This function is determined through the mesh kernel by formula (36)

$$\rho_\alpha = \sum_{j=1}^N q_j \bar{R}(x_\alpha - x_j).$$

The formula realizes the interpolation of the carried feature from the Lagrangian mesh of particles $j = 1, \dots, N$ into nodes x_α of the Euler mesh.

Let us represent ρ_α with the help of formula (56). The kernel $\bar{R}(x_\alpha - x_j)$ we express through the Fourier transform (55). Then we have

$$\rho_\alpha = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} dk \exp(ikx_\alpha) \rho(\hat{k}) = \sum_{j=1}^N q_j \frac{1}{2\pi} \int_{-\infty}^{\infty} dk \hat{R}(k) \exp[ik(x_\alpha - x_j)]. \quad (57)$$

The improper integral in Eq. (57) one can transform into the follow form

$$\int_{-\infty}^{\infty} dk \dots = \sum_{p=-\infty}^{\infty} \int_{\frac{\pi}{h}(2p-1)}^{\frac{\pi}{h}(2p+1)} dk \dots$$

Let make the substitution of the variables

$$k_p = k - \frac{2\pi}{h} p, \quad p = \pm 1, \pm 2 \dots$$

and take into account that in nodes $x_\alpha = \alpha h$ of the Euler mesh

$$\exp(ik_p x_\alpha) = \exp(ik x_\alpha).$$

After this, having changed places the summing up and integration we turn from Eq. (57) to the equality

$$\rho_\alpha = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} dk \left\{ \sum_{j=1}^N q_j \sum_{p=-\infty}^{\infty} \exp(-ik_p x_j) \hat{R}(k_p) \right\} \exp(ik x_\alpha). \quad (58)$$

From it taking into consideration the formula of Eq. (56), it follows that

$$\rho(k) = \sum_{j=1}^N q_j \sum_{p=-\infty}^{\infty} \exp(-ik_p x_j) \hat{R}(k_p). \quad (59)$$

In such a way it appears that additional harmonics $k_p = k - \frac{2\pi}{h} p$, $p = \pm 1, \pm 2, \dots$, give a contribution into the spectrum of the feature for every wave number k . These harmonics are known as aliases. As it follows directly from the derivation of formula (58), their appearance is connected only with the presence of a regular mesh. Evidently the harmonics of the same kind will appear in the spectra of the values connected with the mesh function $\{\rho_\alpha\}$ by means of difference equations on the Euler mesh. For example, in the 2D-problem of the vorticity evolution (see Chapter 4) where $\{\rho_\alpha\}$ is a component of vortex, the error appears in the flow velocity of fluid. In the mentioned above electrostatic and gravitation problems where ρ_α is the charge density or the gravitating mass density, the mesh harmonics give non-physical contribution into potentials of the electric and gravitation fields.

As one can see from the formula (59) for the reducing of these errors it is necessary to suppress as possible the contributions of mesh harmonics k_p far from the wave number k . It can be done by the choice of function $\hat{R}(x)$. In fact, for the simplest kernel (37) (NGP-model) its Fourier transformation is follows

$$\begin{aligned} \hat{R}_N(k) &= \int_{-\infty}^{\infty} \bar{R}_N(x) \exp(-ikx) dx = \frac{1}{h} \int_{-h/2}^{h/2} \exp(-ikx) dx = \\ &= \sin\left(\frac{kh}{2}\right) / \left(\frac{kh}{2}\right) \sim \frac{1}{k}. \end{aligned}$$

Respectively for kernel (38) of PIC-model we have

$$\begin{aligned}\hat{\bar{R}}_P(k) &= \int_{-\infty}^{\infty} \bar{R}_P(x) \exp(-ikx) dx = \frac{1}{h} \int_{|x|<h} (1 - \frac{|x|}{h}) \exp(-ikx) dx = \\ &= \frac{2}{kh} \sin kh - \frac{2}{h^2} \int_0^h x \cos kx dx = \left[\sin\left(\frac{kh}{2}\right) / \left(\frac{kh}{2}\right) \right]^2 \sim \frac{1}{k^2}.\end{aligned}$$

It is seen, that functions $\hat{\bar{R}}_N(k)$ and $\hat{\bar{R}}_P(k)$ in the points $2\pi p/h$ have respectively the 1st and 2nd multiplicities of a zero and they are small in the vicinity of these points. In this case for the kernel PIC the order of smallness is higher and the contribution of mesh harmonics is less. It can be shown that the more smoothness of interpolation is given by the kernel $\bar{R}(x)$ the less contribution of mesh harmonics takes place.

1.4.3. The errors of interpolation

One more typical error appearing in the method of particles is the error of interpolation of features determined on the Lagrangian mesh of particles into the nodes of the Euler mesh. We restrict ourselves to consideration of the error which appears under the projecting of distribution (7) determined by the set of model particles into the space of mesh functions which is given by formulas (30) [41].

Again let us turn to the one-dimensional case with the density defined by Eq. (33) and the projecting operation in the form Eq. (36). Additionally we assume that all the particles have the equal value of the carried feature $m_\alpha = q$, which we call in this point a "charge" though in the following calculations nothing change if we use, for instance, the term "mass".

Let us put the model particles on $[0, L]$ in such a way that their number in the neighborhood of any point should be proportional to the "charge" density in this point. The distribution of the particles will satisfy this condition, if the coordinates of the particles satisfy the equality

$$\int_{x_{j-1}}^{x_j} \rho(x) dx = q, \quad j = 1, 2, \dots, N,$$

where $x_0 = 0$, $x_N = L$. The full charge of the particles in the domain equals $Q = Nq$.

It should be remind that $\bar{R}(x)$ is a symmetrical mesh kernel with the properties

$$\sum_{(\alpha)} \bar{R}(x_\alpha - x) = \frac{1}{h}, \quad \int_0^L \bar{R}(x) dx = 1.$$

Let us estimate the deviation δ_α of the mesh function ρ_α from the initial density $\rho(x)$ in the nodes x_α of mesh:

$$\delta_\alpha = |\rho_\alpha - \rho(x_\alpha)|.$$

From Eq. (36) ρ_α may be represented as follows

$$\begin{aligned} \rho_\alpha &= q \sum_{j=1}^N \frac{\bar{R}(x_\alpha - x_{j-1}) + \bar{R}(x_\alpha - x_j)}{2} + \\ &\quad + \frac{q}{2} [\bar{R}(x_\alpha - x_N) - \bar{R}(x_\alpha - x_0)]. \end{aligned} \quad (60)$$

Let us suppose that $\bar{R}(x_\alpha - x_N) - \bar{R}(x_\alpha - x_0) = 0$. It is true when the density $\rho(x)$ and correspondingly the distribution of the particles are the periodic functions with period L . In other cases, if we are not interested in boundary effects, we can neglect this difference.

Let us assume that $\bar{R}(x) \neq 0$ only at $|x| \leq h/2 + \Delta \equiv \Delta_1$ where Δ is a width of the initial kernel. After this one can write the following equality for arbitrary x_α :

$$\begin{aligned} A &\equiv \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \bar{R}(x_\alpha - x') \rho(x') dx' = \\ &= \int_0^L \bar{R}(x_\alpha - x') \rho(x') dx' = \int_{x_\alpha - \Delta_1}^{x_\alpha + \Delta_1} \bar{R}(x_\alpha - x') \rho(x') dx'. \end{aligned} \quad (61)$$

Adding and subtracting A to the expression under the modulus in the formula for δ_α with using the equalities Eq. (60) and (61), we obtain

$$\delta_\alpha \leq \delta_\alpha^1 + \delta_\alpha^2,$$

where

$$\begin{aligned} \delta_\alpha^1 &= \left| \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \left[\frac{\bar{R}(x_\alpha - x_{j-1}) + \bar{R}(x_\alpha - x_j)}{2} - \right. \right. \\ &\quad \left. \left. - \bar{R}(x_\alpha - x') \right] \rho(x') dx' \right|, \\ \delta_\alpha^2 &= \left| \int_{x_\alpha - \Delta_1}^{x_\alpha + \Delta_1} \bar{R}(x_\alpha - x') \rho(x') dx' - \rho(x_\alpha) \right|. \end{aligned}$$

Let us estimate δ_α^1 and δ_α^2 separately. As $\rho(x) \geq 0$, then according to the mean value theorem

$$\begin{aligned} \int_{x_{j-1}}^{x_j} \left[\frac{\bar{R}(x_\alpha - x_{j-1}) + \bar{R}(x_\alpha - x_j)}{2} - \bar{R}(x_\alpha - x') \right] \rho(x') dx' &\leq \\ &\leq P_1 \int_{x_{j-1}}^{x_j} \rho(x') dx' = P_1 q, \end{aligned}$$

where

$$P_1 = \max_{x_{j-1} < x' < x_j} \left| \frac{\bar{R}(x_\alpha - x_{j-1}) + \bar{R}(x_\alpha - x_j)}{2} - \bar{R}(x_\alpha - x') \right|.$$

If $\bar{R}(x)$ is a piecewise differentiable function, and the interval (x_{j-1}, x_j) contains not more than one discontinuity point of the function $\bar{R}(x_\alpha - x)$, then

$$P_1 \leq \frac{1}{2} \left\{ |\bar{R}^+(x^*) - \bar{R}^-(x^*)| + (x_j - x_{j-1}) \left| \frac{d\bar{R}}{dx} \right|_{max} \right\},$$

where x^* is the discontinuity point.

For estimating δ_α^2 let us expand $\rho(x)$ in the vicinity of node x_α into the truncated Taylor series and integrate the obtained expression term-by-term

$$\begin{aligned} \delta_\alpha^2 &= \left| \int_{x_\alpha - \Delta_1}^{x_\alpha + \Delta_1} \bar{R}(x_\alpha - x') \rho(x') dx' - \rho(x_\alpha) \right| = \\ &= \left| \int_{-\Delta_1}^{\Delta_1} \bar{R}(x) \rho(x_\alpha + x) dx - \rho(x_\alpha) \right| = \\ &= \left| \int_{-\Delta_1}^{\Delta_1} \bar{R}(x) \left[\rho(x_\alpha) + x \frac{\partial \rho}{\partial x} \Big|_{x_\alpha} + \frac{x^2}{2} \frac{\partial^2 \rho}{\partial x^2} \Big|_X \right] dx - \rho(x_\alpha) \right|, \end{aligned}$$

where $X \in [x_\alpha - \Delta_1, x_\alpha + \Delta_1]$. Under the integration the first term gives the density $\rho(x_\alpha)$ and the second one vanishes in accordance with the symmetry $R(x)$. Estimating the reminder term of the expansion according to the mean value theorem we get

$$\delta_\alpha^2 \leq \frac{1}{2} \max_{x_\alpha - \Delta_1 < x < x_\alpha + \Delta_1} \left| \frac{\partial^2 \rho}{\partial x^2} \right| \int_{-\Delta_1}^{\Delta_1} x^2 \bar{R}(x) dx.$$

Uniting the derived estimations, we finally obtain

$$\delta_\alpha \leq \frac{q}{2} \left\{ \sum [\bar{R}] + L \left| \frac{d\bar{R}}{dx} \right|_{max} \right\} + C \Delta_1^2 \max_x \left| \frac{\partial^2 \rho}{\partial x^2} \right|. \quad (62)$$

Here $\sum [\bar{R}]$ is a sum of the moduluses of all jumps of the function \bar{R} . As far as $q = Q/N$, then the first term of the sum is inversely proportional to the number of the particles which are used in the calculation. The value Δ_1 is proportional to mesh step h . Because of that the second term is proportional to h^2 .

For specific mesh kernels the obtained estimation may be refined. Let us make it for two models most widely used in calculation practice.

For NGP-model (37) we have

$$\sum [\bar{R}] = \frac{2}{h}, \quad \left| \frac{d\bar{R}}{dx} \right|_{max} = 0, \quad \Delta_1 = \frac{h}{2},$$

$$\int_{-\Delta_1}^{\Delta_1} x^2 \bar{R}(x) dx = \frac{1}{h} \int_{-h/2}^{h/2} x^2 dx = \frac{h^2}{12}.$$

With using the formula (62) one can deduce

$$\delta_\alpha \leq \frac{q}{h} + \frac{h^2}{24} \max_x \left| \frac{\partial^2 \rho}{\partial x^2} \right| = \frac{\rho_{av}}{\bar{n}} + \frac{h^2}{24} \max_x \left| \frac{\partial^2 \rho}{\partial x^2} \right|,$$

where $\rho_{av} = Q/L$ is a mean density, $\bar{n} = N/K$ is a mean number particles in a cell.

In such a way a mesh function ρ_α calculating in according to NGP-model approximates the initial density $\rho(x)$ with order $O(h^2)$. The derivation δ_α is inversely proportionally to a mean number of particles in a cell.

Now let us consider PIC-model with mesh kernel (38). The function $\bar{R}(x)$ is continuous, but its derivative has the discontinuities in three points. Because of this it is necessary to estimate the integral

$$I_j = \int_{x_{j-1}}^{x_j} \left[\frac{\bar{R}(x_\alpha - x_{j-1}) + \bar{R}(x_\alpha - x_j)}{2} - \bar{R}(x_\alpha - x') \right] \rho(x') dx'$$

separately in each interval containing these points. According to the formula for P_1 we have

$$P_1 = \frac{1}{2h^2} (x_j - x_{j-1}).$$

Because of that

$$|I_j| \leq \frac{q}{2h^2} (x_j - x_{j-1}).$$

In the intervals which don't contain discontinuity points of derivative of kernel $\bar{R}(x)$ the integral I_j equals zero if $|x_j - x_\alpha| \geq h$ and $|x_{j-1} - x_\alpha| \geq h$. In any other cases integral I_j equals

$$I_j = \frac{1}{h^2} \int_{x_{j-1}}^{x_j} \left[-\frac{1}{2}(|x_\alpha - x_j| + |x_\alpha - x_{j-1}|) + |x_\alpha - x'| \right] \rho(x') dx'.$$

Let, for example, $x_{j-1}, x_j < x_\alpha$. Then

$$I_j = \frac{1}{h^2} \int_{x_{j-1}}^{x_j} \left(\frac{x_j + x_{j-1}}{2} - x' \right) \rho(x') dx'. \quad (63)$$

Let us designate $(x_j + x_{j-1})/2 = z_j$ and expand $\rho(x')$ in the vicinity of point z_j into the truncated Taylor series

$$\rho(x') = \rho(z_j) + (x' - z_j) \frac{\partial \rho(X)}{\partial x},$$

where X is some point between z_j and x' . Substituting it into Eq. (63), we derive

$$\begin{aligned}
I_j &= -\frac{1}{h^2} \int_{x_{j-1}}^{x_j} (x' - z_j) \left[\rho(z_j) + (x' - z_j) \frac{\partial \rho(X)}{\partial x} \right] dx' = \\
&= -\frac{1}{h^2} \left(\frac{\partial \rho}{\partial x} \right)_{j \text{ av}} \int_{x_{j-1}}^{x_j} (x' - z_j)^2 dx' = \\
&= -\frac{(x_j - x_{j-1})^3}{12h^2} \left(\frac{\partial \rho}{\partial x} \right)_{j \text{ av}}.
\end{aligned}$$

Difference $x_j - x_{j-1}$ can be estimated with using mean density in the interval $[x_{j-1}, x_j]$

$$(x_j - x_{j-1}) \rho_{j \text{ av}} = q,$$

The number of particles N_1 in the interval $[x_\alpha - h, x_\alpha + h]$ contains in the following limits

$$\frac{2h\rho_{min}}{q} \leq N_1 \leq \frac{2h\rho_{max}}{q}.$$

From this one can deduce that

$$\begin{aligned}
\delta_\alpha^1 &= \left| \sum_j I_j \right| \leq \\
&\leq \frac{3q}{2h^2} (x_j - x_{j-1}) + N_{1 \text{ max}} \frac{(x_j - x_{j-1})^3}{12h^2} \left| \frac{\partial \rho}{\partial x} \right|_{\text{max}} = \\
&= \frac{3\rho_{av}^2}{2\bar{n}^2 \rho_{min}} + \frac{h\rho_{av}^2 \rho_{max}}{6\bar{n}^2 \rho_{min}^3} \left| \frac{\partial \rho}{\partial x} \right|_{\text{max}} = \\
&= \left(\frac{3\rho_{av}^2}{2\rho_{min}} + h \frac{\rho_{av}^2 \rho_{max}}{6\rho_{min}^3} \left| \frac{\partial \rho}{\partial x} \right|_{\text{max}} \right) \frac{1}{\bar{n}^2}.
\end{aligned}$$

Estimation δ_α^2 gives us the following result

$$\begin{aligned}
\delta_\alpha^2 &= \left| \int_{x_\alpha-h}^{x_\alpha+h} \bar{R}(x_\alpha - x') \rho(x') dx' - \rho(x_\alpha) \right| = \\
&= \left| \int_{x_\alpha-h}^{x_\alpha+h} \bar{R}(x_\alpha - x') \left[\rho(x_\alpha) + (x' - x_\alpha) \frac{\partial \rho}{\partial x} \Big|_{x_\alpha} + \right. \right. \\
&\quad \left. \left. + \frac{(x' - x_\alpha)^2}{2!} \frac{\partial^2 \rho(X)}{\partial x^2} \right] dx' - \rho(x_\alpha) \right|,
\end{aligned}$$

where X is located between x and x_α . Computing the integrals, we derive

$$\delta_\alpha^2 \leq \frac{h^2}{12} \left| \frac{\partial^2 \rho}{\partial x^2} \right|_{\text{max}}.$$

Consequently, in the case of PIC-model the following estimation takes place for deviation

$$\delta_\alpha \leq \left(\frac{3\rho_{av}^2}{2\rho_{min}} + h \frac{\rho_{av}^2 \rho_{max}}{6\rho_{min}^3} \left| \frac{\partial \rho}{\partial x} \right|_{max} \right) \frac{1}{n^2} + \frac{h^2}{12} \left| \frac{\partial^2 \rho}{\partial x^2} \right|_{max}.$$

Thus, in the considered PIC- and NGP-models the order of approximation with respect to the space variable is $O(h^2)$. In contrast to NGP-model the error of fulfillment of the feature (charge) density function by the interpolation on the particles for PIC-model is inversely proportionally to the square of the mean number of particles in the cell.

The presented estimations show the dependence of computation accuracy of particle method on the form of the mesh kernel.

1.5. The continuity equation in the particle method

In all realizations of the particle-in-cell method on the Lagrangian step some system of differential conservation laws in the form of Eq. (4) is modeled. However, in a lot of cases divergent equations for the features carried by the particles already present in the initial non-splitting problem. In this case, the problem of realization of difference analog of the corresponding conservation law appears, which is determined on the Euler mesh. One of the frequently encountered example is the continuity equation of some scalar feature such as charge, mass, vorticity at the absence of its sources (sinks) in the coordinate space. This equation which is encountered in the plasma physics, fluid mechanics, gas dynamics and in other applications has the following form

$$\frac{\partial \rho}{\partial t} + \operatorname{div} j = 0. \quad (64)$$

where ρ is the density of the carried feature, j is the flow density of this feature. For plasma dynamics this may be, for example, an electric charge and current densities, for gas dynamic correspondingly the volume density and the density of mass flow, etc.

The difference equation on the Euler mesh corresponding to Eq. (64) we write down as follows

$$\frac{\rho_\alpha^{m+1} - \rho_\alpha^m}{\tau} + \operatorname{div}_h j_\alpha^m = 0, \quad (65)$$

where τ is a time step, div_h is an operator of difference divergence.

Let the mesh density after the recurrent moving of particles in the Lagrangian step be calculated according to formulas in the form (21)

$$\rho_\alpha^{m+1} = \sum_{(p)} q_p \bar{R}(r_\alpha - r_p^{m+1}) \quad (66)$$

It seems quite natural if we use the same formula for determination of current density:

$$\mathbf{j}_\alpha^m = \sum_{(p)} q_p \mathbf{v}_p^m \bar{R}(\mathbf{r}_\alpha - \mathbf{r}_p^m), \quad (67)$$

or similar other one depending on concrete form of Eq. (65). However, it appears that the simultaneous using of formulas (66), (67) doesn't allow to satisfy the difference continuity equation by any model of particles with the finite kernel. As the interpolation of the feature carried by particles into the mesh nodes according to the formula (66) is the obligatory element connecting both steps of the splitting scheme, then for realization of conservativeness on the Euler mesh the special algorithm is necessary.

Let us show the possibility of construction of such an algorithm on the one-dimensional example. Let us consider the divergent difference scheme for Eq. (64) in the following form

$$\frac{\rho_{\alpha-1/2}^{m+1} - \rho_{\alpha-1/2}^m}{\tau} + \frac{j_\alpha^{m+1/2} - j_{\alpha-1}^{m+1/2}}{h} = 0. \quad (68)$$

Here the mesh density function $\{\rho_\alpha\}$ is determined in the nodes that are centers of cells. Flows are calculated on the boundaries of cells.

Such a choice gives the second order of approximation on h and allows to realize the mesh analog of formula of Ostrogradskii-Hauss (divergence theorem). That is to provide for conservativeness. In the scheme Eq. (68) shifting time meshes are used and that allows to receive the second order of approximation on τ . The similar scheme is frequently encountered in the problems of plasma physics (see Chapter 5) when for equation of particle dynamics the leapfrog scheme is used. In this case, for particle coordinates the following scheme is implied

$$\frac{x_p^{m+1} - x_p^m}{\tau} = v_p^{m+1/2}. \quad (69)$$

Scheme (68) is linear with respect to the mesh functions contained in it. Therefore it is enough to realize the difference continuity equation for every individual particle. Then its correctness for full densities and flows received by summing up on all particles will be guaranteed.

The flow density is determined by the formula

$$j_{\alpha+1/2}^{m+1/2} = \sum_{(p)} q_p v_p^{m+1/2} \bar{R}(x_{\alpha+1/2} - x_p^{m+1/2}). \quad (70)$$

Below for definiteness we use PIC-model of particles with mesh kernel (38)

$$\bar{R}(x_\alpha - x_p) = \begin{cases} \frac{1}{h} \left(1 - \frac{|x_\alpha - x_p|}{h} \right), & |x_\alpha - x_p| \leq h \\ 0, & |x_\alpha - x_p| > h \end{cases} \quad (71)$$

Let us consider the individual particle which at the time moments $t_m = m\tau$, $t_{m+1} = (m+1)\tau$ has the coordinates x^m and x^{m+1} correspondingly.

The following variants of particle movements at time step τ are possible.

- a. $x_{\alpha-1} < x^m, x^{m+1} < x_\alpha$ i.e., the particle moves between the centers of two neighbouring cells. Then according to the chosen model (71) and formula (66) the particle in both locations x^m and x^{m+1} contributes a feature only into the nodes $x_{\alpha-1}$ and x_α :

$$\rho_{\alpha-1}^m = q\bar{R}(x_{\alpha-1} - x^m) = \frac{q}{h} \left(1 - \frac{x^m - x_{\alpha-1}}{h} \right), \quad (72)$$

$$\rho_\alpha^m = q\bar{R}(x_\alpha - x^m) = \frac{q}{h} \left(1 - \frac{x_\alpha - x^m}{h} \right), \quad (73)$$

Expressions for contributions at the time moment τ_{m+1} are derived from Eqs. (72), (73) by the simple substitution of the time index. In this case, the density change connected with the particle displacements is expressed as follows:

$$\begin{aligned} \rho_{\alpha-1}^{m+1} - \rho_{\alpha-1}^m &= q\bar{R}(x_{\alpha-1} - x^{m+1}) - q\bar{R}(x_{\alpha-1} - x^m) = \\ &- \frac{q}{h^2} (x^{m+1} - x^m), \end{aligned} \quad (74)$$

$$\rho_\alpha^{m+1} - \rho_\alpha^m = q\bar{R}(x_\alpha - x^{m+1}) - q\bar{R}(x_\alpha - x^m) = \frac{q}{h^2} (x^{m+1} - x^m), \quad (75)$$

Let us note that the total increment of the density equals zero, and the conservation law of the carried feature is realized.

From difference scheme (68) these density increments can be expressed in terms of the flows of the feature across the cell boundaries in the following way

$$\rho_{\alpha-1}^{m+1} - \rho_{\alpha-1}^m = -\frac{\tau}{h} (j_{\alpha-1/2}^{m+1/2} - j_{\alpha-3/2}^{m+1/2}), \quad (76)$$

$$\rho_\alpha^{m+1} - \rho_\alpha^m = -\frac{\tau}{h} (j_{\alpha+1/2}^{m+1/2} - j_{\alpha-1/2}^{m+1/2}), \quad (77)$$

To all the other nodes of the mesh $x_{\alpha\pm(1+k)}$, $k = 1, 2, \dots$ considered particle gives zero contributions. Then, if there are no flows from the outside, sequentially considering the difference Eq. (68) in all nodes one can obtain that $j_{\alpha-3/2}^{m+1/2} = j_{\alpha+1/2}^{m+1/2}$ equals zero. In this case, from formulas (76), (77) the following expressions succeed

$$j_{\alpha-1/2}^{m+1/2} = -h \frac{\rho_{\alpha-1}^{m+1} - \rho_{\alpha-1}^m}{\tau}, \quad j_{\alpha-1/2}^{m+1/2} = h \frac{\rho_\alpha^{m+1} - \rho_\alpha^m}{\tau}.$$

Substituting here the corresponding increments of the density (74), (75) we receive that from both formulas the flow $j_{\alpha-1/2}^{m+1/2}$ is determined identically in the form

$$j_{\alpha-1/2}^{m+1/2} = \frac{q}{h} \frac{x^{m+1} - x^m}{\tau} = \frac{q}{h} v^{m+1/2}, \quad (78)$$

where in the last equality Eq. (69) is used.

Under this approach the difference continuity equation (68) appears to be satisfied and the particle with PIC-kernel (71) contributes into the flow only in one boundary node $x_{\alpha-1}$, separating two cells between the centers of which the particle is located.

Now let us calculate the flows on the formula (70). For definiteness let us assume that in the whole interval of time (t_m, t_{m+1}) the particle has the velocity $v^{m+1/2}$. Then its location at the time moment $t_{m+1/2}$ can be expressed as follows

$$x^{m+1/2} = x^m + \frac{\tau}{2} v^{m+1/2} = \frac{x^{m+1} + x^m}{2}.$$

Let $x_{\alpha-1/2} > x^{m+1}$. In this case, in accordance with kernel (71) the particle contributes into the flows in two boundary nodes $x_{\alpha-1/2}, x_{\alpha-3/2}$. This already contradicts to the of conservation law. For the flows we obtain the following expressions

$$\begin{aligned} j_{\alpha-3/2}^{m+1/2} &= q \frac{v^{m+1/2}}{h^2} (x_{\alpha-1/2} - x^{m+1/2}), \\ j_{\alpha-1/2}^{m+1/2} &= q \frac{v^{m+1/2}}{h^2} (x^{m+1/2} - x_{\alpha-3/2}). \end{aligned} \quad (79)$$

The last ones, it is clear, don't satisfy the difference scheme (68) in which increments (76), (77) are used.

Let us consider another variant:

b. At $x_{\alpha-1} < x^m < x_\alpha < x^{m+1} < x_{\alpha+1}$ the particle during step τ moves across mesh node x_α from left to right. At the moment t_m the particle, as in the previous case, gives the contributions in the form of Eqs. (72), (73) into nodes $x_\alpha, x_{\alpha+1}$:

$$\rho_\alpha^{m+1} = \frac{q}{h} \left(1 - \frac{x^{m+1} - x_\alpha}{h} \right), \quad \rho_{\alpha+1}^{m+1} = \frac{q}{h} \left(1 - \frac{x_{\alpha+1} - x^{m+1}}{h} \right), \quad (80)$$

Thus, in this case the density changing takes place in three mesh nodes $x_{\alpha-1}, x_\alpha, x_{\alpha+1}$. From the difference scheme the density increment in nodes $x_{\alpha-1}, x_\alpha$ is still expressed by formulas (76), (77). The increment in node $x_{\alpha+1}$ is written similarly:

$$\rho_{\alpha+1}^{m+1} - \rho_{\alpha+1}^m = -\frac{\tau}{h} (j_{\alpha+3/2}^{m+1/2} - j_{\alpha+1/2}^{m+1/2}). \quad (81)$$

In this case

$$\rho_{\alpha+1}^m = \rho_{\alpha-1}^m = 0. \quad (82)$$

Under requirement that the difference conservation law (68) should be satisfied in all the other nodes of the mesh $x_{\alpha \pm (k+1)}, k = 1, 2, \dots$, where density doesn't change, we derive $j_{\alpha-3/2}^{m+1/2} = j_{\alpha+3/2}^{m+1/2} = 0$. Then from Eq. (76) taking into account (80), (82), we deduce the expression for the flow in node $x_{\alpha-1/2}$:

$$j_{\alpha-1/2}^{m+1/2} = \frac{h}{\tau} \rho_{\alpha-1}^m = \frac{q}{h} \frac{x_\alpha - x^m}{\tau}, \quad (83)$$

and from Eq. (81)

$$j_{\alpha+1/2}^{m+1/2} = \frac{h}{\tau} \rho_{\alpha+1}^{m+1} = \frac{q}{h} \frac{x_\alpha - x^m}{\tau}. \quad (84)$$

Substitution of expressions (83), (84) into difference Eq. (77) gives us

$$\rho_\alpha^{m+1} - \rho_\alpha^m = -\frac{q}{h^2} [(x^{m+1} - x_\alpha) + (x^m - x_\alpha)].$$

The direct calculation of the density increment in nodes x_α from interpolation formulas (73), (80) leads to the same result. Thus, for the derived formulas for flow density (83), (84) the difference continuity equation is satisfied. At the same time the using of formula (70) for computation of flows violates the difference conservation law (68).

c. At last, for $x_{\alpha-1} < x^{m+1} < x_\alpha < x^m < x_{\alpha+1}$ at the particle movement from right to left we similarly carry out the following relations for flow densities

$$j_{\alpha-1/2}^{m+1/2} = -\frac{h}{\tau} \rho_{\alpha-1}^{m+1} = \frac{q}{h} \frac{x^{m+1} - x_\alpha}{\tau}, \quad (85)$$

$$j_{\alpha+1/2}^{m+1/2} = -\frac{h}{\tau} \rho_{\alpha+1}^m = \frac{q}{h} \frac{x_\alpha - x^m}{\tau}. \quad (86)$$

Comparing formulas (73), (83)-(86), one can note the universal property which allows to construct easily the numerical algorithm. Namely, the trajectory of the particle in the time step is subdivided into separate parts by nodes x_α and within every part $[x_\alpha, x_{\alpha+1}]$ the corresponding flow density $j_{\alpha+1/2}^{m+1/2}$ is calculated according to one and the same formula

$$j_{\alpha+1/2}^{m+1/2} = \frac{q}{h} \frac{x_e - x_i}{\tau}, \quad (87)$$

where x_i and x_e are the initial and endpoint coordinates of the particle in this part.

Thus, the use of the obtained formulas for the flow density (73), (83)-(86) together with the procedure of density interpolation on the Euler mesh according to formula (66) allows to construct the conservative algorithm without violation of the difference analog of continuity equation (64).

We should note that the break of the divergent property is not connected with the use of any concrete manner of computation of velocity $v_p^{m+1/2}$ as well as with use of *leapfrog scheme* [42] because in all formulas for computing the flow densities are used only initial and finite particle coordinates.

Absolutely analogously one can analyze the simple scheme of the form

$$\frac{\rho_{\alpha}^{m+1} - \rho_{\alpha}^m}{\tau} + \frac{j_{\alpha+1/2}^m - j_{\alpha-1/2}^m}{\tau} = 0, \quad (88)$$

$$\frac{x^{m+1} - x^m}{\tau} = v^m.$$

In this case, the unique expressions for flows are also obtained which allow to satisfy the difference continuity equation (88).

Let us write down formulas for flow density in two-dimensional and three-dimensional cases. In the three-dimensional case the feature density and flow density must satisfy the difference scheme

$$\frac{\rho_{\alpha,\beta,\gamma}^{m+1} - \rho_{\alpha,\beta,\gamma}^m}{\tau} = \frac{j_{x,\alpha+1/2,\beta,\gamma}^{m+1/2} - j_{x,\alpha-1/2,\beta,\gamma}^{m+1/2}}{h_x} +$$

$$\frac{j_{y,\alpha,\beta+1/2,\gamma}^{m+1/2} - j_{y,\alpha,\beta-1/2,\gamma}^{m+1/2}}{h_y} + \frac{j_{z,\alpha,\beta,\gamma+1/2}^{m+1/2} - j_{z,\alpha,\beta,\gamma-1/2}^{m+1/2}}{h_z}, \quad (89)$$

where h_x , h_y , h_z are steps on space coordinates (x , y , z); j_x , j_y , j_z are components of the vector of flow density \mathbf{j} . The integer subscripts designate the cell centers, and the flows are determined in the middles of the cell sides. As in Eq. (68) densities and flows are computed on shifted time meshes. Thus, scheme (89) approximates continuity equation (64) with the second order $O(\tau^2, h_x^2, h_y^2, h_z^2)$. Together with *leapfrog scheme* for Lagrangian particles it is often used in the problems of plasma dynamics (see Chapter 5). To satisfy Eq. (89) the contribution of individual particles into flow density must be computed according to formulas similar to formula (87) derived for the one-dimensional case. As in one-dimensional case, the rectilinear trajectory connecting the initial \mathbf{r}^m and finite \mathbf{r}^{m+1} locations of the particle is subdivided into separate segments $(\mathbf{r}_i, \mathbf{r}_e)$, every one of them is located in a parallelepiped II: $\{x_\alpha < x < x_{\alpha+1}; y_\beta < y < y_{\beta+1}; z_\gamma < z < z_{\gamma+1}\}$ and every segment contributes into flow density in mesh nodes corresponding to this parallelepiped:

$$j_{x,\alpha+1/2,\beta,\gamma}^{m+1/2} = q \frac{\Delta x}{\tau} \left[(1 - \delta_y)(1 - \delta_z) + \frac{\Delta y \cdot \Delta z}{12h_y h_z} \right],$$

$$j_{x,\alpha+1/2,\beta,\gamma+1}^{m+1/2} = q \frac{\Delta x}{\tau} \left[(1 - \delta_y)\delta_z - \frac{\Delta y \cdot \Delta z}{12h_y h_z} \right],$$

$$j_{x,\alpha+1/2,\beta+1,\gamma}^{m+1/2} = q \frac{\Delta x}{\tau} \left[\delta_y(1 - \delta_z) - \frac{\Delta y \cdot \Delta z}{12h_y h_z} \right],$$

$$j_{x,\alpha+1/2,\beta+1,\gamma+1}^{m+1/2} = q \frac{\Delta x}{\tau} \left[\delta_y \delta_z + \frac{\Delta y \cdot \Delta z}{12h_y h_z} \right],$$

$$j_{y,\alpha,\beta+1/2,\gamma}^{m+1/2} = q \frac{\Delta y}{\tau} \left[(1 - \delta_x)(1 - \delta_z) + \frac{\Delta x \cdot \Delta z}{12h_x h_z} \right].$$

$$\begin{aligned}
j_{y,\alpha,\beta+1/2,\gamma+1}^{m+1/2} &= q \frac{\Delta y}{\tau} \left[(1 - \delta_x) \delta_z - \frac{\Delta x \cdot \Delta z}{12h_x h_z} \right], \\
j_{y,\alpha+1,\beta+1/2,\gamma}^{m+1/2} &= q \frac{\Delta y}{\tau} \left[\delta_x (1 - \delta_z) - \frac{\Delta x \cdot \Delta z}{12h_x h_z} \right], \\
j_{y,\alpha+1,\beta+1/2,\gamma+1}^{m+1/2} &= q \frac{\Delta y}{\tau} \left[\delta_x \delta_z + \frac{\Delta x \cdot \Delta z}{12h_x h_z} \right], \\
j_{z,\alpha,\beta,\gamma+1/2}^{m+1/2} &= q \frac{\Delta z}{\tau} \left[(1 - \delta_x)(1 - \delta_y) + \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right], \\
j_{z,\alpha,\beta+1,\gamma+1/2}^{m+1/2} &= q \frac{\Delta z}{\tau} \left[(1 - \delta_x) \delta_y - \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right], \\
j_{z,\alpha+1,\beta,\gamma+1/2}^{m+1/2} &= q \frac{\Delta z}{\tau} \left[\delta_x (1 - \delta_y) - \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right], \\
j_{z,\alpha+1,\beta+1,\gamma+1/2}^{m+1/2} &= q \frac{\Delta z}{\tau} \left[\delta_x \delta_y + \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right],
\end{aligned}$$

where $\Delta x = x_e - x_i$, $\Delta y = y_e - y_i$, $\Delta z = z_e - z_i$ are increments of coordinates in the parallelepiped Π ;

$$\begin{aligned}
\delta_x &= \frac{1}{h_x} \left(\frac{x_i + x_e}{2} - x_{\alpha+1/2} \right), \\
\delta_y &= \frac{1}{h_y} \left(\frac{y_i + y_e}{2} - y_{\beta+1/2} \right), \\
\delta_z &= \frac{1}{h_z} \left(\frac{z_i + z_e}{2} - z_{\gamma+1/2} \right).
\end{aligned}$$

In the two-dimensional case in plane (x, y) the formulas for contributions into flow density from particles have the following form:

$$\begin{aligned}
j_{x,\alpha+1/2,\beta}^{m+1/2} &= q \frac{\Delta x}{\tau} (1 - \delta_y), \\
j_{x,\alpha+1/2,\beta+1}^{m+1/2} &= q \frac{\Delta x}{\tau} \delta_y, \\
j_{y,\alpha,\beta+1/2}^{m+1/2} &= q \frac{\Delta y}{\tau} (1 - \delta_x), \\
j_{y,\alpha+1,\beta+1/2}^{m+1/2} &= q \frac{\Delta y}{\tau} \delta_x, \\
j_{z,\alpha,\beta}^{m+1/2} &= q v_z^{m+1/2} \left[(1 - \delta_x)(1 - \delta_y) + \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right], \\
j_{z,\alpha,\beta+1}^{m+1/2} &= q v_z^{m+1/2} \left[(1 - \delta_x) \delta_y - \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right],
\end{aligned}$$

$$j_{z,\alpha+1,\beta}^{m+1/2} = qv_z^{m+1/2} \left[\delta_x(1 - \delta_y) - \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right],$$

$$j_{y,\alpha+1,\beta+1}^{m+1/2} = qv_z^{m+1/2} \left[\delta_x \delta_y + \frac{\Delta x \cdot \Delta y}{12h_x h_y} \right],$$

where $v_z^{m+1/2}$ is z-component of particle velocity. The other designations are the same as in the three-dimensional case.

Chapter 2

Particle-in-cell methods on unstructured meshes

2.1. Introduction

The methods of particles on regular meshes described in the Introduction and in the previous chapter widely used in various applications. However, they can be used for numerical modeling in domains of relatively simple configuration into which orthogonal Euler meshes (Cartesian, cylindrical or spherical) can be introduced. On these meshes simple and economical algorithms were constructed for typical computational procedures such as the determination of current coordinates of particles, and direct and inverse interpolation between meshes and particles. Besides, fast solvers were created for the computation of generalized fields on the Euler meshes.

Moreover, the needs of fundamental science and applied researches lead to the necessity of using the particle method for modeling in domains with complicated geometry of boundaries or with strong local gradients of generalized fields where the lack of exactness can lead to global distortions of the modeling process as a whole. This situation can appear in the investigation of some non-stationary plasma processes, for example, in ion diodes, i.e. setups for producing powerful ion beams [43], or in problems of computational aerodynamics that arise in the modeling of flows past of complicated constructions, when the boundaries of the computational domain can not be approximated by coordinate surfaces.

That is why during the last decade various variants of the particle method have been developed for the numerical modeling of processes in domains of complicated shape [44, 45, 46, 47, 48, 49, 50]. They base on meshes that adequately describe the boundaries of the computational domain. In the existing literature such meshes are named boundary-fitted coordinates, which are subdivided into structured inhomogeneous meshes and unstructured ones.

Up to now in the literature on numerical methods there have been no generally accepted definitions of structured and unstructured meshes. The most general representation of an arbitrary computational mesh corresponds to the concept of a dimensionally homogeneous connected geometric (topological) complex [51]. In this case a mesh is considered as a complex of interconnected and interdependent geometric elements,

such as nodes, faces, edges, cells, etc. As structured meshes we usually mean meshes for which the topological structure of the corresponding complex is identically characterized by the set of indices of node points.

An unstructured mesh can be defined as a mesh that is not structured. By a more constructive definition, an unstructured mesh [52] is a totality of polygons or polyhedrons of arbitrary dimensions called cells or elements that may have various number of neighbors but correctly affiliate to each other. The vertices of these polygons (polyhedrons) are called nodes, and their sides are called edges or faces.

As to irregular meshes, structured and unstructured, it is necessary to work out new algorithms of particle localization in the Euler meshes, and also of "mesh-particles" and "particles-mesh" interpolation. For structured meshes these problems are often solved by mapping of the initial mesh domain in the coordinate space onto a *logical* rectangular mesh. If the main steps of the particle method are realized on a logical mesh [46], then the location of particles is simply determined, and the procedures described in the first chapter can be used for interpolation. However, because of errors due to numerical mapping of a mesh onto a logical mesh, there appear errors in the trajectories of particles. In order to avoid them sometimes only the calculation of generalized fields (the Euler step) is carried out on the logical mesh, and the particle motion is realized on the initial non-uniform mesh in the physical domain [47]. But in this case it is necessary to solve again the problem of localization of particles on an inhomogeneous mesh, and under some conditions to solve also the problem of interpolation.

Thus, the use of auxiliary logical meshes for structured meshes leads to specific errors in the particle trajectories or to the necessity of constructing new algorithms of particle localization (and interpolation) on the initial inhomogeneous meshes. In this case there is no point in mapping onto a logical mesh. Besides, this approach is not suitable for unstructured meshes, which can not be immersed into a regular logical mesh by any mapping [53].

To solve the problem of particle localization there are algorithms that are realized both on structured and unstructured meshes. Specifically, to trace the particle trajectories a detailed auxiliary Cartesian mesh can be used for the coordinate space, or a sequence of such meshes with decreasing mesh-sizes can be used [54]. This approach is easily realized and suitable for parallelization, but if in the initial mesh the cell sizes vary over wide limits, the efficiency and accuracy of the method can decrease significantly. This is due to the difficulty of finding a satisfactory compromise between the size of auxiliary Cartesian mesh and the average size of elements in the initial mesh.

An alternative strategy for the construction of a particle localization algorithm is a successive search for their location on the next time step in the nearest neighbor cells with the use of local bases of linear functions taken from the finite-element method of [33]. As it is shown in

papers [49, 53, 55], such an approach appeared to be efficient for plane domains with triangulation as well as for three-dimensional domains with tetrahedrization.

From this short survey of possible ways of realization of the particle-in-cell method by using boundary-fitted coordinates, it follows that, from the point of view of characteristic computational procedures, the approach based on unstructured meshes is the most universal one. In fact, there are no restrictions on this approach, in terms of geometry of the boundaries and connectivity of the computational domain. The generation of such meshes is easily automatized, and this allows one to adapt them dynamically according to such local peculiarities of the solution as zones of discontinuities and topology changing, free and contact boundaries, extreme gradients, etc. Up to now a vast experience is accumulated concerning the construction and use of unstructured meshes in finite-element methods and finite-volume ones, which are widely used in solving problems of mechanical strength of constructions, outer and inner aerodynamics and in other applications. Search algorithms of particle localization and "particles-mesh" and "mesh-particles" interpolation algorithms worked out for unstructured meshes can be used also for inhomogeneous structured meshes. All this allows us to concentrate on the peculiarities of realization of the particle-in-cell method on unstructured meshes.

Nowadays there are several papers concerning problems of algorithms of the particle method on unstructured meshes (see, for example, [49, 50]). The presentation of the material in this chapter follows papers [56, 57, 58]. First the necessary information about local bases of finite elements are given. For the Lagrangian step we consider some search algorithms of particle localization initially suggested in [49, 53]. Some interpolation procedures for point particles and particles of finite dimensions are described. In order to make the presentation closed, we give a numerical scheme for the Euler step based on the finite-volume method. The construction of a computational algorithm of generalized fields is shown by the example of the Maxwell equations.

2.2. Finite-elements bases

Before passing to the description of particle methods on unstructured meshes let us introduce the necessary notions of finite element bases, which play an important role in the realization of the method. Here we follow paper [33]. It is customary that mesh cells are called elements according to the terminology of the finite-element method. In the plane case we consider unstructured meshes consisting of triangular elements derived by some triangulation of the computational domain. In the three-dimensional case we use the meshes generated by the elements — tetrahedrons. In this case, the nodes of the mesh are the

vertices of the triangles or tetrahedrons.

Every finite element is determined by the set of basis functions (shape-functions) the number of which equals the number of nodes (vertices) of this element. In general polynomial basis functions are used in the finite-element method. The exponents of them determine the orders of the finite elements. In our case we restrict ourselves to linear basis functions. That means that we deal with linear finite elements.

Simultaneously with basis functions "natural" coordinates are introduced on every element. Their main purpose is describing the location of arbitrary point r_0 inside (outside) of the element. Their peculiarity is in the fact that the number of coordinates coincides with the number of element vertices. Moreover every of "natural" coordinates equals unit at one of the vertices and becomes zero at other vertices (nodes) of this element. Local coordinates are connected with global Cartesian ones by linear relations.

Two-dimensional case. On every finite element C_i with number i and local indexing of triangle vertices $k = 1, 2, 3$ three local basis functions N_k^i , are determined. These functions are linear on both Cartesian coordinates (x, y) and equal unit at the vertex with the coordinate (x_k^i, y_k^i) and equal zero at the other vertices:

$$N_k^i(x, y) = \alpha_k^i + \beta_k^i x + \gamma_k^i y, \quad N_k^i(x_m^i, y_m^i) = \delta_{km}. \quad (1)$$

Here δ_{km} is the Kronecker symbol.

On the given triangulation for determining the coefficients of base functions we have three systems of linear algebraic equations

$$\begin{pmatrix} 1 & x_1^i & y_1^i \\ 1 & x_2^i & y_2^i \\ 1 & x_3^i & y_3^i \end{pmatrix} \begin{pmatrix} \alpha_k^i \\ \beta_k^i \\ \gamma_k^i \end{pmatrix} = \begin{pmatrix} \delta_{1,k} \\ \delta_{2,k} \\ \delta_{3,k} \end{pmatrix}, \quad k = 1, 2, 3. \quad (2)$$

For elements with the finite area the systems Eqs. (2) are identically solvable. Really, by derive the equivalent transformation of the determinant Eqs. (2) one can receive

$$\Delta_i = \begin{vmatrix} 1 & x_1^i & y_1^i \\ 1 & x_2^i & y_2^i \\ 1 & x_3^i & y_3^i \end{vmatrix} = \begin{vmatrix} 1 & x_1^i & y_1^i \\ 0 & x_2^i - x_1^i & y_2^i - y_1^i \\ 0 & x_3^i - x_1^i & y_3^i - y_1^i \end{vmatrix} = \quad (3)$$

$$(x_2^i - x_1^i)(y_3^i - y_1^i) - (x_3^i - x_1^i)(y_2^i - y_1^i).$$

On the other hand, from Fig. 7 it is seen that $|\Delta_i| = |(\mathbf{r}_2^i - \mathbf{r}_1^i) \times (\mathbf{r}_3^i - \mathbf{r}_1^i)|$, that means that the modulus of the system determinant (2) equals the doubled area i -triangular element.

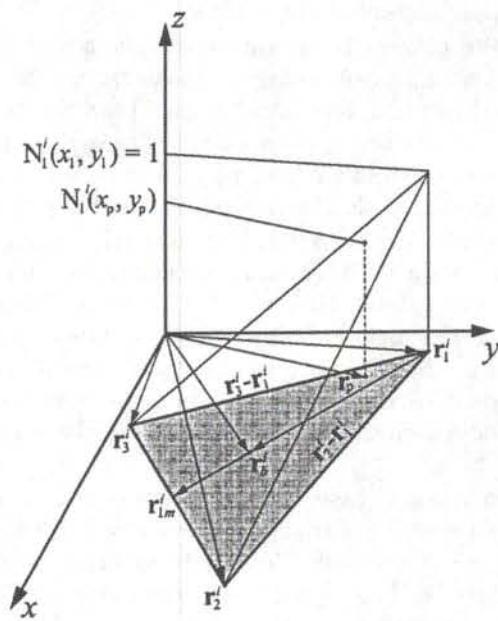


Fig. 7. The characteristics of the triangular element in the global coordinate system.

Therefore the solutions of systems Eqs. (2) can be found by Cramer formulas. In particular, for coefficients of the local basis function N_1^i we have

$$\alpha_1^i = \frac{x_2^i y_3^i - x_3^i y_2^i}{\Delta_i}, \quad \beta_1^i = \frac{y_2^i - y_3^i}{\Delta_i}, \quad \gamma_1^i = \frac{x_3^i - x_2^i}{\Delta_i}. \quad (4)$$

Expressions for coefficients of local basis functions N_2^i , N_3^i can be obtained from Eq. (4) by the cyclic permutation of indices.

Local coordinates L_1^i, L_2^i, L_3^i for arbitrary triangular element with number i shown in Fig. 7 are determined by relations

$$\begin{aligned} L_1^i x_1^i + L_2^i x_2^i + L_3^i x_3^i &= x, \\ L_1^i y_1^i + L_2^i y_2^i + L_3^i y_3^i &= y, \\ L_1^i + L_2^i + L_3^i &= 1. \end{aligned} \quad (5)$$

As in Eq. (1) subscripts designate local numbers of the nodes of this element. As far as the location of the point on the plane is identically characterized by the assigning of its two coordinates, then local coordinates L_1, L_2, L_3 are not linearly independent. The latter condition in Eqs. (5) reflects the linear dependence of local coordinates. Under this introduction of local coordinates the determinant of the system Eqs. (5) coincides with the determinant Eq. (3)

$$\begin{vmatrix} x_1^i & x_2^i & x_3^i \\ y_1^i & y_2^i & y_3^i \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x_1^i & y_1^i & 1 \\ x_2^i & y_2^i & 1 \\ x_3^i & y_3^i & 1 \end{vmatrix} = \Delta_i. \quad (6)$$

and, consequently, it differs from zero on any triangular element of the finite area. Thus, the systems Eqs. (5) are identically solvable at any values of Cartesian coordinates (x, y) . In particular, directly from Eqs. (5) it is seen that at $x = x_k^i$, $y = y_k^i$ $L_m^i = \delta_{km}$. In general case, for coordinate L_1^i from systems Eqs. (5) it is obtained the following expression in terms of current Cartesian coordinates (x, y) :

$$L_1^i(x, y) = \frac{x_2^i y_3^i - y_2^i x_3^i}{\Delta_i} + \frac{y_2^i - y_3^i}{\Delta_i} x + \frac{x_3^i - x_2^i}{\Delta_i} y = \alpha_1^i + \beta_1^i x + \gamma_1^i y, \quad (7)$$

where formulas (4) are used. Expressions for $L_2^i(x, y)$, $L_3^i(x, y)$ are received from (7) by the cyclic permutation of indices. Comparison of formulas (1), (4) and (7) and of analogous ones for $k = 2, 3$ allows to conclude that "natural" coordinates on the triangular element coincide with its linear basis functions (shape-functions)

$$L_k^i(x, y) = N_k^i(x, y), \quad k = 1, 2, 3. \quad (8)$$

Three-dimensional case. In the case of three space variables for arbitrary tetrahedron finite element (see Fig. 8) with the local indexing of the nodes (vertices) $k = 1, 2, 3, 4$ and the volume V_i the linear basis functions are determined by relations

$$N_k^i(\mathbf{r}) = \alpha_k^i + \beta_k^i x + \gamma_k^i y + \delta_k^i z, \quad N_k^i(\mathbf{r}_m^i) = \delta_{km}. \quad (9)$$

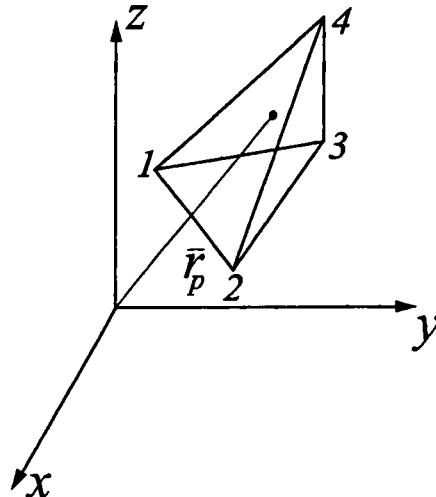


Fig. 8. Tetrahedron element in the global system of coordinates.

As above, coefficients in Eq. (9) are calculated as solutions of the system of linear algebraic equations, the determinant of which has the following form:

$$\Delta_i = \begin{vmatrix} 1 & x_1^i & y_1^i & z_1^i \\ 1 & x_2^i & y_2^i & z_2^i \\ 1 & x_3^i & y_3^i & z_3^i \\ 1 & x_4^i & y_4^i & z_4^i \end{vmatrix}. \quad (10)$$

In this case $|\Delta_i| = 6V_i$ and for non-degenerate finite elements of nonzero volume basis functions are determined identically. For example, for $N_1^i(\mathbf{r})$ coefficients are determined as

$$\alpha_1^i = \frac{\begin{vmatrix} x_2^i & y_2^i & z_2^i \\ x_3^i & y_3^i & z_3^i \\ x_4^i & y_4^i & z_4^i \end{vmatrix}}{\Delta_i}, \quad \beta_1^i = -\frac{\begin{vmatrix} 1 & y_2^i & z_2^i \\ 1 & y_3^i & z_3^i \\ 1 & y_4^i & z_4^i \end{vmatrix}}{\Delta_i},$$

$$\gamma_1^i = -\frac{\begin{vmatrix} x_2^i & 1 & z_2^i \\ x_3^i & 1 & z_3^i \\ x_4^i & 1 & z_4^i \end{vmatrix}}{\Delta_i}, \quad \delta_1^i = -\frac{\begin{vmatrix} x_2^i & y_2^i & 1 \\ x_3^i & y_3^i & 1 \\ x_4^i & y_4^i & 1 \end{vmatrix}}{\Delta_i}. \quad (11)$$

Local "natural" coordinates $L_1^i, L_2^i, L_3^i, L_4^i$ on the i -element are expressed in terms of current Cartesian coordinates by the system of linear equations analogous to Eqs. (5)

$$\begin{aligned} L_1^i x_1^i + L_2^i x_2^i + L_3^i x_3^i + L_4^i x_4^i &= x, \\ L_1^i y_1^i + L_2^i y_2^i + L_3^i y_3^i + L_4^i y_4^i &= y, \\ L_1^i z_1^i + L_2^i z_2^i + L_3^i z_3^i + L_4^i z_4^i &= z, \\ L_1^i + L_2^i + L_3^i + L_4^i &= 1. \end{aligned} \quad (12)$$

As in the plane case, "natural" coordinates introduced in such a way for the tetrahedron finite element coincide with the determined on it basis of linear functions

$$L_k^i(x, y, z) = N_k^i(x, y, z), \quad k = 1, 2, 3, 4. \quad (13)$$

The marked equivalence of "natural" coordinates and linear bases on the finite elements allows to use linear shape-functions not only in the procedures of interpolation what is quite natural, but also in search algorithms of the particle location on the mesh where they play the role of local coordinates. The fulfilled consideration of them makes such an unification quite obvious.

2.3. The Lagrangian step on unstructured meshes

When passing to unstructured meshes, the general scheme of the particle method remains unchanged, but the main procedures differ significantly from represented above for orthogonal meshes.

Let us suppose that the initial problem in the form Eq. (1.1) on the step τ of the evolutional variable t is factorized approximately in the form

$$\mathbf{q}(t_p + \tau, \mathbf{u}) \cong (I + \tau E)(I + \tau L)\mathbf{q}(t_p, \mathbf{u}),$$

where both the operator L of the Lagrangian step and operator E of the Euler step have the divergent form. The latter is necessary if for computation of generalized fields on the unstructured meshes the finite-volume method is used. Let dynamic variables \mathbf{u} be subdivided into generalized coordinates $\mathbf{r} = \{r_1, \dots, r_s\}$ and corresponding to them generalized momentums $\mathbf{p} = \{p_1, \dots, p_s\}$. Then discretization is fulfilled in the form Eq. (1.7)

$$\mathbf{q}(t, \mathbf{r}, \mathbf{p}) = \sum_{j=1}^N \mathbf{Q}_j R(\mathbf{r}, \mathbf{r}_j(t)) \delta(\mathbf{p} - \mathbf{p}_j(t)). \quad (14)$$

We should remember that here the function $R(\mathbf{r}, \mathbf{r}_j(t))$ is the kernel of the model particle which gives its form and the distribution of features in it that are the components of vector \mathbf{q} ; $\delta(\mathbf{p})$ is s -dimensional Dirac distribution. In the case of point particles $R(\mathbf{r}, \mathbf{r}_j(t)) = \delta(\mathbf{r} - \mathbf{r}_j)$. In accordance with the general scheme of the "particle-in-cell" algorithm described in Chapter 1, the moving of model particles in phase space is realized at the Lagrangian step as well as the procedures of interpolation "mesh-particles" and "particles-mesh" are carried out. Let the moving of particles be realized along the trajectories of the dynamic system

$$\frac{d\mathbf{r}_j}{dt} = \mathbf{p}_j, \quad \frac{d\mathbf{p}_j}{dt} = \mathbf{F}_j(\mathbf{r}_j, \mathbf{p}_j), \quad j = 1, \dots, N. \quad (15)$$

calculated on the base of some finite-difference schemes. For fulfillment of the interpolation procedures it's necessary to establish addresses (multi-indices) of the Euler cells into which the "centers" of model particles locate.

2.3.1. Particle localization algorithms

Effective procedures for particle localization on the unstructured meshes consisting of triangular elements in the case of two space dimensions and tetrahedrons in the three-dimensional one were constructed in [49, 53, 55]. The idea of successive search for particle location in the nearest neighboring cells (successive neighbor searching) after their displacement at the step τ is realized in them. Local basis functions Eqs. (1), (9) playing the role of local coordinates are used for this purpose. It is supposed that for every element the local bases N_k^i are computed in terms of Cartesian coordinates nodes (vertices) by formulas Eqs. (4), (11) for given partition of the computational domain in coordinate space Ω_r .

Let the particle center at the time moment t^n be in point $\mathbf{r}^n \in C_i$. Under given above definition of linear basis functions for any $\mathbf{r} \in C_i$

and for all $k = 1, \dots, d+1$ we have $0 \leq N_k^i(\mathbf{r}) \leq 1$. Therefore $\mathbf{r}^{n+1} \in C_i$ at the time moment $t^{n+1} = t^n + \tau$ if and only if $\min N_k^i(\mathbf{r}^{n+1}) \geq 0$ and $\max N_k^i(\mathbf{r}^{n+1}) \leq 1$ for $\forall k, k = 1, \dots, d+1$.

If $\min N_k^i(\mathbf{r}^{n+1}) < 0$, at least for one of the nodes (vertices), then $\mathbf{r}^{n+1} \notin C_i$.

In the plane case if $\mathbf{r}^{n+1} \notin C_i$ the search should be continued in the neighboring element lying opposite to k -vertex of the element C_i for which $N_k^i(\mathbf{r}^{n+1}) = \min_l N_l^i(\mathbf{r}^{n+1}) < 0$. (see Fig. 9). Subroutine on the Fortran-77 which realizes the search in the two-dimensional case is given in Supplement B.

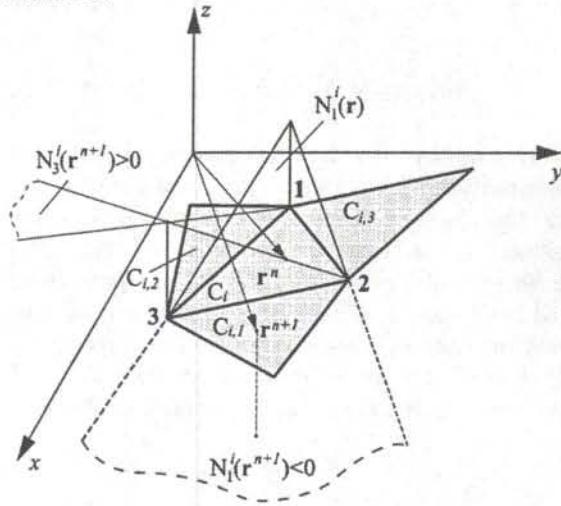


Fig. 9. Searching for particle location in triangulated domain.

In the three-dimensional case the search algorithm of the particle location was suggested in [49]. Let us assume that the local right-oriented indexing of the nodes is introduced in every tetrahedron. Denote by \mathbf{B}_i vectors lying on the edges of the tetrahedron and connecting the vertex with number $k \neq i$ with the rest ones. Then so that the tetrahedron should be right-oriented it will be enough that vectors $(\mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4)$ in Fig. 10a should form the right triple. In order words it means that the triple scalar product $D_1 = (\mathbf{B}_2 \times \mathbf{B}_3, \mathbf{B}_4)$ must be nonnegative. In this case the right triples can be also formed from vectors \mathbf{B}_i , outgoing from other vertices $k = 2, 3, 4$ (see Fig. 10). The following triple scalar products $D_2 = (\mathbf{B}_1 \times \mathbf{B}_4, \mathbf{B}_3)$, $D_3 = (\mathbf{B}_1 \times \mathbf{B}_2, \mathbf{B}_4)$, $D_4 = (\mathbf{B}_1 \times \mathbf{B}_3, \mathbf{B}_2)$ will be positive for them. Let us remind that D_k are calculated as determinants, the lines of which are formed by Cartesian coordinates of vectors \mathbf{B}_i .

Let us consider the right-oriented element C_i with vertices M_k^i , having Cartesian coordinates $\mathbf{r}_k^i = (x_k^i, y_k^i, z_k^i)$, $k = 1, \dots, 4$ (see Fig. 11). Denote by P_k the planes in which the faces of tetrahedron C_i , opposite

to the tops M_k^i , lie. Let introduce vectors $\mathbf{A}_k = \mathbf{r}_k^i - \mathbf{r}^{n+1}$, directed from the point of particle location towards the vertices M_k^i . We put together the mixed products from them

$$\begin{aligned} D_1^i(\mathbf{r}^{n+1}) &= (\mathbf{A}_2 \times \mathbf{A}_3, \mathbf{A}_4), & D_2^i(\mathbf{r}^{n+1}) &= (\mathbf{A}_1 \times \mathbf{A}_4, \mathbf{A}_3), \\ D_3^i(\mathbf{r}^{n+1}) &= (\mathbf{A}_1 \times \mathbf{A}_2, \mathbf{A}_4), & D_4^i(\mathbf{r}^{n+1}) &= (\mathbf{A}_1 \times \mathbf{A}_3, \mathbf{A}_2), \end{aligned} \quad (16)$$

which will be used for searching the particle location.

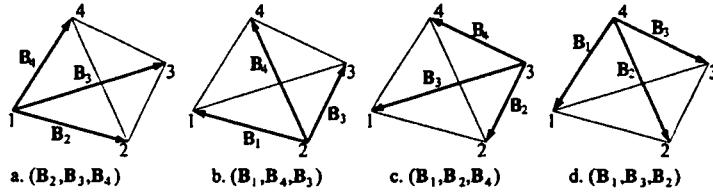


Fig. 10. The right triples of the vectors on the right-oriented tetrahedron.

If at the time moment t^{n+1} the particle still remains inside the element C_i , that means $\mathbf{r}^{n+1} \in C_i$, then by the continuous movement of the particle center within the element into every of vertices M_k^i one can superpose the vectors \mathbf{A}_i , $i \neq k$, onto vectors \mathbf{B}_i (see the corresponding variant in Fig. 10). While doing this it's obviously that $D_k^i(\mathbf{r}^{n+1}) = D_k > 0$. From the continuity of the transformation $D_k^i(\mathbf{r}^{n+1}) \rightarrow D_k$ it follows that at $\mathbf{r}^{n+1} \in C_i$ all determinants $D_k^i(\mathbf{r}^{n+1}) > 0$.

Now let $\mathbf{r}^{n+1} \notin C_i$. Then for every determinant $D_k^i(\mathbf{r}^{n+1})$ two cases are possible.

a. The vertex M_k^i and the particle center \mathbf{r}^{n+1} lie in one half-space with respect to the plane P_k . Then one can move the particle center into the vertex M_k^i by continuous transformations of dilatation and translation in this half-space. These transformations don't change the orientation of the vector triple in the determinant $D_k^i(\mathbf{r}^{n+1})$; then $D_k^i = D_k > 0$. As it is above, it allows to conclude that in this case $D_k^i(\mathbf{r}^{n+1}) > 0$.

b. The vertex M_k^i and point \mathbf{r}^{n+1} are in different half-spaces with respect to the plane P_k . Then the mapping $D_k^i(\mathbf{r}^{n+1}) \rightarrow D_k$ additionally to continuous transformations of dilatation and translation requires also mirror image relatively to the plane P_k . Under mirror image the orientation of the vector triple entering D_k^i is changed into the inverse one. Thus, in this case $D_k^i(\mathbf{r}^{n+1}) < 0$.

This allows to offer the following scheme of the search algorithm [49]. The following alternatives are possible after the computation of determinants $D_k^i(\mathbf{r}^{n+1})$ Eqs. (16).

1. If $\forall D_k^i(\mathbf{r}^{n+1}) > 0$, then the particle remains inside the element C_i .

2. If only one of the determinants $D_k^i(\mathbf{r}^{n+1}) < 0$, then the search of the particle should be continued in the element $C_{i,k}$, lying opposite to the vertex M_k^i (see Fig. 11).

3. Let two determinants, for definiteness, be $D_1^i(\mathbf{r}^{n+1}) < 0$ and $D_2^i(\mathbf{r}^{n+1}) < 0$. In order to continue the search in this case it's necessary to determine the face across which the particle trajectory went while outgoing from the element C_i .

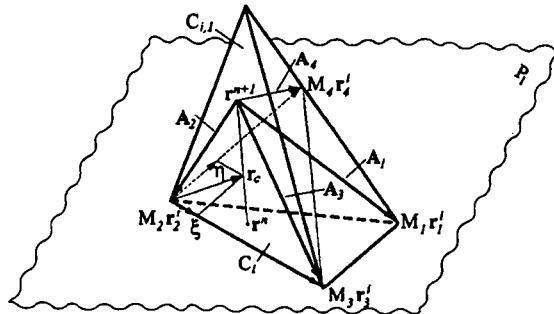


Fig. 11. The search of the particle location in the tetrahedron elements.

The equation of the particle trajectory at the time interval $[t_n, t_{n+1}]$ has the form

$$\mathbf{r}_p = \mathbf{r}^n + \theta \mathbf{p}^{n+1/2}, \quad (17)$$

where $\mathbf{p}^{n+1/2}$ is the generalized particle momentum. Let us consider the face which is opposite to the vertex M_1^i . Let us introduce the local basis of vectors lying on the edges $(\mathbf{r}_3^i - \mathbf{r}_2^i, \mathbf{r}_4^i - \mathbf{r}_2^i)$ as it is shown in Fig. 11. Let \mathbf{r}_c be the radius-vector of the intersection point of the particle trajectory with the plane P_1 containing this face. One can expand the vector $\mathbf{r}_c - \mathbf{r}_2^i$ in the local basis

$$\mathbf{r}_c - \mathbf{r}_2^i = \xi(\mathbf{r}_3^i - \mathbf{r}_2^i) + \eta(\mathbf{r}_4^i - \mathbf{r}_2^i), \quad (18)$$

where (ξ, η) are local coordinates of the intersection point of the particle with the plane P_1 . The obvious alternative takes place. If $(\xi, \eta) \in [0, 1]$ and $\xi + \eta \leq 1$, then the intersection point lies in the limits of this face, and the particle search should be continued in the element $C_{i,1}$. Otherwise, when $(\xi, \eta) \notin [0, 1]$, the trajectory intersects the other face, and the search should be continued in the element $C_{i,2}$.

In fact, for the computation of local coordinates (ξ, η) it's not necessary to find properly the intersection point \mathbf{r}_c . Really, the unknown vector can be represented as

$$\mathbf{r}_c - \mathbf{r}_2^i = (\mathbf{r}^{n+1} - \mathbf{r}_2^i) - (\mathbf{r}^{n+1} - \mathbf{r}_c), \quad (19)$$

where vector $\mathbf{r}^{n+1} - \mathbf{r}_c$ is collinear to the vector of the generalized momentum $\mathbf{p}^{n+1/2}$. Besides, $\mathbf{r}^{n+1} = \mathbf{r}^n + \tau \mathbf{p}^{n+1/2}$. Substituting Eq. (19)

into Eq. (18) and multiplying vectorially both parts of the equality by $\mathbf{p}^{n+1/2}$, we obtain the system for determining (ξ, η) :

$$(\mathbf{r}^{n+1} - \mathbf{r}_2^i) \times \mathbf{p}^{n+1/2} = \xi(\mathbf{r}_3^i - \mathbf{r}_2^i) \times \mathbf{p}^{n+1/2} + \eta(\mathbf{r}_4^i - \mathbf{r}_2^i) \times \mathbf{p}^{n+1/2}. \quad (20)$$

4. If three determinants, for example, $D_1^i(\mathbf{r}^{n+1})$, $D_2^i(\mathbf{r}^{n+1})$, $D_3^i(\mathbf{r}^{n+1})$ become negative, then according to the described algorithm we successively check the two faces located opposite to vertices M_1^i and M_2^i on intersection by the particle trajectory. If the trajectory intersects one of them, then the search should be continued in elements $C_{i,1}$ or $C_{i,2}$. Otherwise the search should be fulfilled in element $C_{i,3}$. It should be noted that the systems of the form (20) for determining the local coordinates (ξ, η) appear to be overdetermined. But in computations this difficulty is overcome by Gauss algorithm using.

Subroutine of particle localization on the three-dimensional unstructured mesh consisting of tetrahedron elements is brought in Supplement.

2.3.2. The interpolation procedures

Let us further consider the interpolation procedures "particles-mesh" and "mesh-particles". While fulfilling the interpolation "particles-mesh" it's necessary to project the solution \mathbf{q} Eq. (14) into the space of mesh functions $\{\tilde{\mathbf{q}}\}$. Let us consider the model particle with phase coordinates $(\mathbf{r}_j, \mathbf{p}_j)$, which is inside the element C_i of the unstructured mesh determined by localization procedure. The simplest interpolation algorithm corresponds to the model "nearest-grid-point" (see Eq. (1.22)) when the point particle is considered. The projection operation onto the Euler mesh is determined by the Steklov projector (see Eq. (1.25)):

$$\tilde{\mathbf{q}}_i = P_\omega \tilde{\mathbf{q}} = \sum_{j=1}^N \tilde{Q}_j \frac{1}{|\omega_i|} \int_{\omega_i} d\mathbf{r} \delta(\mathbf{r} - \mathbf{r}_j(t)) = \sum_{j=1}^N \tilde{Q}_j \frac{\xi_i(\mathbf{r}_j)}{|\omega_i|} = \sum_{\{j\} \in \omega_i} \tilde{Q}_j. \quad (21)$$

Here $|\omega_i|$ is the Lebesgue measure of the element C_i , $\xi(\mathbf{r})$ is the characteristic function (identifier) C_i . In the sum from the right side the summation is carried out over all particles the centers of which \mathbf{r}_j are located in the element C_i . The received in such a way density of the vector feature is assigned to barycentre \mathbf{r}_i^i of the considered element. As it was already shown in Chapter 1, the conservation law of the total feature $\mathbf{Q} = \sum_{j=1}^N \mathbf{Q}_j$ appears to be satisfied. In the other variant of this model the value of the feature Q_j relates to the node m of the element C_i for which $N_m^i(\mathbf{r}_j) = \max_k N_k^i(\mathbf{r}_j)$, $k = 1, \dots, d+1$. However, as it was noted in p. 2 of the first chapter, in the case of restricted number of particles this procedure leads to the high fluctuations of mesh functions $\tilde{\mathbf{q}}_i$.

In traditional particle methods on Cartesian meshes for reducing such fluctuations the transition to mesh kernels is realized that correspond to the PIC-, CIC- and others models [7, 22, 38]. It's quite

obvious that if we don't use auxiliary Cartesian meshes, then because of the chaotic character of dimensions and orientation of unstructured mesh cells the direct transference of this approach on the unstructured meshes is impossible. One of the variants for solution of this problem already approved in the calculations [49, 56, 57] consists in coincidence of the smoothing (see Chapter 1) with the projection operation on the mesh. For this purpose one can use local basis functions $N_k^i(\mathbf{r})$ determined on every element C_i of the unstructured mesh in the projection operator. Let the discretization be realized according to formula Eq. (14) by the particles with the singular kernel $\delta(\mathbf{r} - \mathbf{r}_j(t))$. We determine the mesh density function of the vector feature \mathbf{q} by the formula

$$\tilde{\mathbf{q}}_k = \frac{1}{V_k} \sum_{\{a\}} \int_{\Omega} d\mathbf{r} d\mathbf{p} \mathbf{q}(t, \mathbf{r}, \mathbf{p}) N_k^a(\mathbf{r}) = \frac{1}{V_k} \sum_a \sum_j \mathbf{Q}_j N_k^a(\mathbf{r}_j), \quad (22)$$

where V_k is the $(d+1)$ -dimensional hypervolume associated with the k -node; by the index a we denote the elements surrounding k -node. The summation is carried out over all elements having the common node k and over all particles with centers \mathbf{r}_j located in these elements. For the twodimensional case the associated volume consisting of straight tetrahedrons is shown in Fig. 12. The upper faces of these tetrahedrons are prescribed by the values of basis functions $N_k^a(\mathbf{r})$ that gives us the following expression

$$V_k = \sum_a \int_{\Omega_r} d\mathbf{r} N_k^a(\mathbf{r}). \quad (23)$$

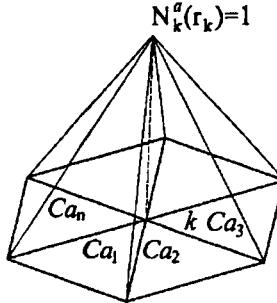


Fig. 12. The hypervolume V_k associated with k -node in the 2D-case.

Directly from Fig. 12 it can be easily deduce

$$V_k = \sum_a \frac{1}{3} \cdot 1 \cdot \frac{\Delta_a}{2}, \quad (24)$$

where Δ_a is the doubled area of the triangular element C_a .

The common expression for V_k has the form [49]

$$V_k = \sum_a \frac{|\omega_a|}{d+1}. \quad (25)$$

Let the integral conservation law takes place

$$\mathbf{Q} = \int_{\Omega} dr d\mathbf{p} \mathbf{q}(t, \mathbf{r}, \mathbf{p}) = \sum_{j=1}^N Q_j = const. \quad (26)$$

It can be seen that the discrete analog Eq. (26) is fulfilled for the mesh function Eq. (22). The following equality is valid

$$\sum_{(k)} \sum_{(a)} N_k^a(\mathbf{r}) = \sum_{(i)} \sum_{m=1}^{d+1} N_m^i(\mathbf{r}) = \sum_{(i)} \xi_i(\mathbf{r}). \quad (27)$$

The summation in the double sum at the left-hand-side is carried out on the set (k) of all nodes of the mesh domain. For every node the basis functions of all elements C_a associated with the given node are summed up. Regrouping leads to the summation on all elements C_i of the given partition of domain Ω_r ; for each of them all basis functions belonging to it are summed up. Then taking into account all the conditions Eq. (5) or Eq. (12) it's easy to obtain

$$\sum_{k=1}^{d+1} N_k^i(\mathbf{r}) = \xi_i(\mathbf{r}) = \begin{cases} 1, & \mathbf{r} \in C_i \\ 0, & \mathbf{r} \notin C_i \end{cases}$$

whence the relation Eq. (27) follows. Now let us sum up the mesh function $\{\tilde{q}_k\}$ on all nodes of the mesh domain. Under relation Eq. (27) we have

$$\sum_{(k)} \tilde{q}_k V_k = \sum_{(k)} \sum_{(a)} \int_{\Omega} dr d\mathbf{p} \mathbf{q}(t, \mathbf{r}, \mathbf{p}) N_k^a = \sum_i \int_{C_i} dr \int d\mathbf{p} \mathbf{q}(t, \mathbf{r}, \mathbf{p}) = \sum_{j=1}^N Q_j.$$

On the other hand, analogously for the right part of the equality Eq. (22) it is derived

$$\sum_{(k)} \tilde{q}_k V_k = \sum_{(k)} \sum_{(a)} \sum_{(j)} Q_j N_k^a(\mathbf{r}_j) = \sum_{j=1}^N Q_j.$$

Thus, the interpolation procedure "particles-mesh" determined by the formula Eq. (22) allows to satisfy the discrete conservation law of the vector feature carried by the particles .

Let us show that the smoothing given by formulas of Eq. (22) corresponds to the PIC-model widely used in the particle method on the Cartesian meshes (see p.p. 1,2 of Chapter 1). For the sake of simplicity we restrict ourselves to the one-dimensional case. While doing this, it's obvious that the unstructured mesh turns into a irregular mesh with variable step on the number line R^1 . The mesh steps we denote as $h_\alpha = x_\alpha - x_{\alpha-1}$. Linear basis functions of the element between the nodes $x_\alpha, x_{\alpha-1}$ are given by the relations

$$N_{\alpha-1}(x) = a_\alpha + b_\alpha x, \quad N_{\alpha-1}(x_\beta) = \delta_{\alpha-1,\beta}$$

$$N_\alpha(x) = a_\alpha + b_\alpha x, \quad N_\alpha(x_\beta) = \delta_{\alpha,\beta}.$$

From here having obtained the expressions for the coefficients, we get

$$N_{\alpha-1}(x) = \frac{x_\alpha - x}{x_\alpha - x_{\alpha-1}}, \quad N_\alpha(x) = \frac{x - x_{\alpha-1}}{x_\alpha - x_{\alpha-1}}. \quad (28)$$

"Associated" with these nodes hypervolumes in accordance with formulas Eqs. (23), (25) equal

$$V_{\alpha-1} = \frac{1}{2} (h_{\alpha-1} + h_\alpha) = \bar{h}_{\alpha-1}, \quad V_\alpha = \frac{1}{2} (h_\alpha + h_{\alpha+1}) = \bar{h}_\alpha. \quad (29)$$

Let the center of particle with mass m_p has the coordinate $x_{\alpha-1} < x_p < x_\alpha$, as it is shown in Fig. 1. Then from Eqs. (22), (28), (29) it follows that contributions into the mesh function of the density is expressed as follows

$$\begin{aligned} \rho_{\alpha-1} &= \frac{1}{V_{\alpha-1}} m_p N_{\alpha-1}(x_p) = \frac{m_p}{\bar{h}_{\alpha-1}} \frac{x_\alpha - x_p}{h_\alpha}, \\ \rho_\alpha &= \frac{1}{V_\alpha} m_p N_\alpha(x_p) = \frac{m_p}{\bar{h}_\alpha} \frac{x_p - x_{\alpha-1}}{h_\alpha}. \end{aligned} \quad (30)$$

Comparing the relations with expressions Eqs. (1.32) for the one-dimensional PIC-model, we conclude that up to notations of variable steps h_α, \bar{h}_α they coincide. In both cases the contributions of the particles are distributed by inverse linear interpolation into the two nearest nodes between which it is located.

In the two-dimensional case the similar correspondence to the PIC-model can be received if instead of triangulation we use the partition into quadrangle elements with linear bases. In this case, the contribution of the particle would be distributed among four nodes of the element by inverse bilinear interpolation. Subroutines realizing the interpolation algorithm Eqs. (22), (25) are presented in Supplement B.

Under interpolation "mesh-particles" it's necessary to interpolate the values of generalized force fields $\tilde{\mathbf{F}}_k$, computed on the Euler mesh into the particle locations. In the simplest case one can determine the force acting onto the particle with phase coordinates $(\mathbf{r}_j, \mathbf{p}_j)$ by formula:

$$\mathbf{F}(\mathbf{r}_j, \mathbf{p}_j) = \mathbf{F}(\mathbf{r}'_k), \quad (31)$$

where \mathbf{r}'_k are the coordinates of the node of the unstructured mesh nearest to this particle.

A different way of the generalized force interpolation consists in using of linear basis functions according to their direct assignment

$$\mathbf{F}(\mathbf{r}_j, \mathbf{p}_j) = \sum_{k=1}^{d+1} N_k^i(\mathbf{r}_j) \mathbf{F}(\mathbf{r}_k^i). \quad (32)$$

Here $\mathbf{F}(\mathbf{r}_k^i)$ are the values of generalized force in the vertices of the element C_i inside which this particle locates. The interpolation algorithm of generalized fields is realized in the form of a subroutine presented in Supplement B.

Sometimes in the process of calculation it becomes necessary to interpolate mesh functions derived at the Euler mesh nodes into the barycentres of the elements. Here the connection between global Cartesian coordinates of the vertices and the barycentres is necessary.

In the two-dimensional case for the arbitrary triangular element shown in Fig. 7 it's easy to derive a relation for the radius-vector of the barycentre:

$$\mathbf{r}_b^i = \mathbf{r}_k^i + \frac{2}{3} \mathbf{r}_{km}^i, \quad k = 1, 2, 3, \quad (33)$$

where \mathbf{r}_{km}^i is a radius-vector of the median drawn from the k -vertex. For example,

$$\mathbf{r}_{1m}^i = \frac{1}{2} (\mathbf{r}_2^i + \mathbf{r}_3^i) - \mathbf{r}_1^i.$$

The symmetrization of the expression Eq. (33) gives us

$$\mathbf{r}_b^i = \frac{1}{3} \sum_{k=1}^3 \left(\mathbf{r}_k^i + \frac{2}{3} \mathbf{r}_{km}^i \right) = \frac{1}{3} \left\{ \sum_{k=1}^3 \mathbf{r}_k^i + \frac{2}{3} \left[2 \sum_{k=1}^3 \frac{1}{2} \mathbf{r}_k^i - \sum_{k=1}^3 \mathbf{r}_k^i \right] \right\} = \frac{1}{3} \sum_{k=1}^3 \mathbf{r}_k^i \quad (34)$$

Analogously we can obtain the expression for coordinates of the barycentre of the tetrahedron element. The universal expression is given by the formula

$$\mathbf{r}_b^i = \frac{1}{d+1} \sum_{k=1}^{d+1} \mathbf{r}_k^i. \quad (35)$$

For interpolation from the nodes of the element C_i into its barycentre it's quite natural to use formulas of the type Eq. (32):

$$\tilde{\mathbf{q}}_b^i = \sum_{k=1}^{d+1} N_k^i(\mathbf{r}_b^i) \tilde{\mathbf{q}}_k^i.$$

Though basis functions are not homogeneous, but it's directly checked that the substitution Eq. (35) gives us

$$N_k^i(\mathbf{r}_b^i) = N_k^i \left(\frac{1}{d+1} \sum_{l=1}^{d+1} \mathbf{r}_l^i \right) = \frac{1}{d+1} \sum_{l=1}^{d+1} N_k^i(\mathbf{r}_l^i) = \frac{1}{d+1} \sum_{l=1}^{d+1} \delta_{kl} = \frac{1}{d+1}.$$

As a result, a simple interpolation formula is obtained

$$\tilde{\mathbf{q}}_b^i = \frac{1}{d+1} \sum_{k=1}^{d+1} \tilde{\mathbf{q}}_k^i, \quad (36)$$

That means that the value in the barycentre equals arithmetic mean on the nodes.

2.3.3. Numerical examples

To illustrate the efficiency described in points 2.1, 2.2 of the algorithms and the programs realizing them and given in Supplement, let us consider the results of computation of some test problems. In the plane case the domain in which test calculations were made represented a unit square. The examples of unstructured meshes introduced into this domain are shown in Fig. 13. The mesh in Fig. 13a contains 130 nodes and 222 triangles and the mesh in Fig. 13b contains 701 nodes and 1320 triangles.

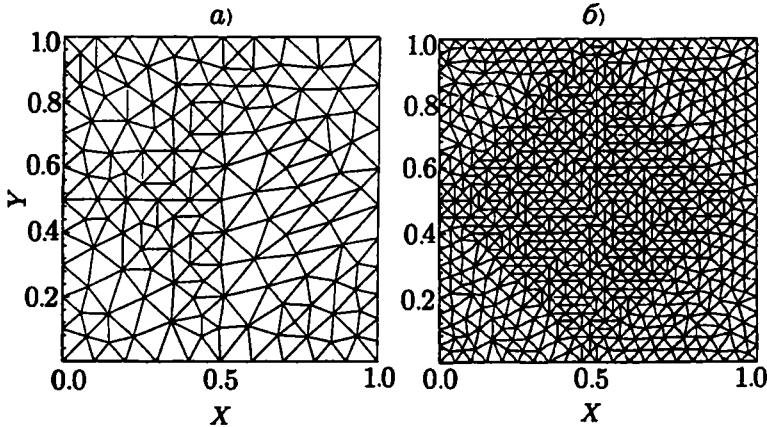


Fig. 13. Examples of unstructured meshes used in test calculations:
a. 130 nodes, 222 elements; b. 701 nodes, 1320 elements.

For checking the accuracy of the interpolation procedure "particles-mesh" the following test has been considered. The model domain was divided into N Cartesian cells coinciding with the number of the particles $N \in [400 \div 409600]$ which we placed into domain. One model particle was placed into the center of every cell. The same scalar feature to every model particle was assigned:

$$Q_j = Q = \tilde{q}_0/N, \quad j = 1, 2, \dots, N.$$

Here \tilde{q}_0 is the value of the scalar feature, for example, the electric charge or the mass which was supposed to be constant in Ω_r .

Further in accordance with the procedure of particle localization described in p. 2.1 (the corresponding subroutine is brought in Supplement B) the indexes of mesh elements C_i containing every particle were determined.

The example of the search trajectory of the particle location (searching for host element) is shown in Fig. 14. With taking into account the found particle locations the values $\{\tilde{q}_k^i\}$ at the nodes of the unstructured mesh were calculated by formula Eq. (22). The recalculation of the found $\{\tilde{q}_k^i\}$ into the barycentres of the elements was realized by Eq.

(36). Then the comparison of mesh densities $\{q_k^i\}$ $\{q_b^i\}$ was fulfilled with the given exact value q_0 .

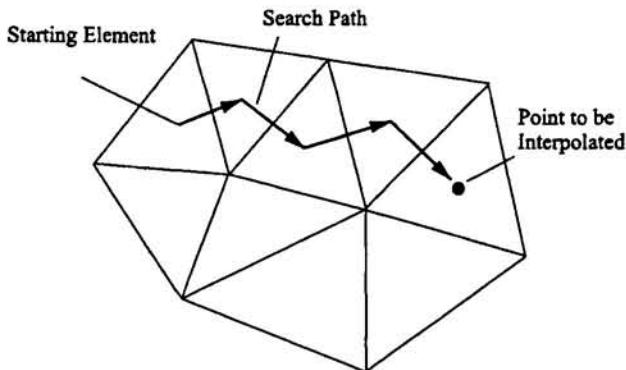


Fig. 14. The search for the particle location on the two-dimensional unstructured mesh.

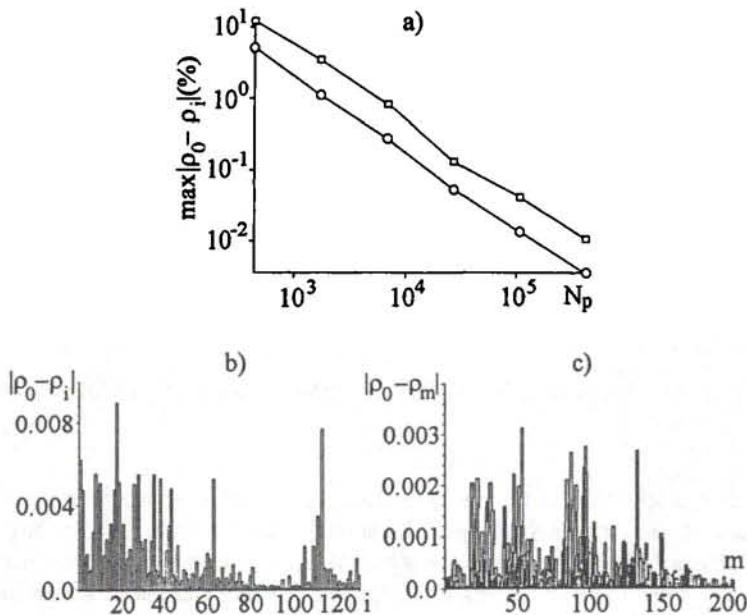


Fig. 15. The results of test calculations on restoration of the uniform density in two-dimensional case (130 nodes, 222 elements).

The results of the numerical tests are presented in Fig. 15. In Fig. 15 the maximum absolute values $|\rho_0 - \rho_i|$ are shown. Its are the density feature deviations at the nodes (squares) and at the barycentres (circles)

of the unstructured mesh depending on the number of model particles. It is seen that these deviations for the given mesh is less than 1% if the number of particles in the domain $N \geq 6400$. Histograms of the density feature deviations from the theoretical value for $N = 6400$ are shown in Fig. 15b,c. Analogous histograms for the case of three dimensions and for an unstructured mesh containing 200 nodes and 744 elements are shown in Fig. 16.

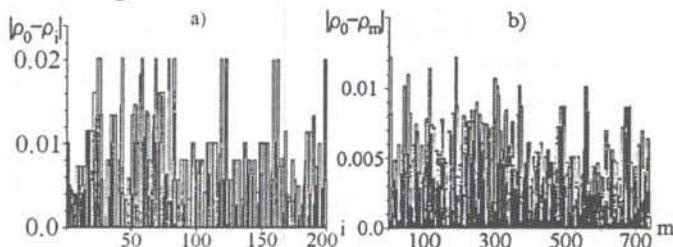


Fig. 16. The results of test calculations on restoration of the uniform density in the three-dimensional case (200 nodes, 744 elements).

The significant numerical example which can serve for approving of the main procedures of the Lagrangian step is the plane Kepler problem on moving of solitary particle in the centrally symmetric field [59].

The force field of is given by relations

$$F_x = \frac{-0.1r_x}{(\sqrt{r_x^2 + r_y^2})^3}, \quad F_y = \frac{-0.1r_y}{(\sqrt{r_x^2 + r_y^2})^3}, \quad (37)$$

where $r_x = x - 0.5$, $r_y = y - 0.5$. At the time moment $t = 0$ the initial conditions were posed

$$x(0) = 0.2, \quad y(0) = 0.5, \quad v_x(0) = 0, \quad v_y(0) = \frac{1}{\sqrt{3}}.$$

In this case the exact solution for the particle trajectory in this domain is the circle

$$x^2 + y^2 = 0.3^2.$$

For calculation of the particle trajectory at the beginning of every time step the current location of the particle was determined, that means the index of the element C_i , in which it is located at the time moment $t = t^n$. Then the force acting upon the particle was calculated according to formulas Eq. (37), and the integration of the equations of the particle motion was carried out. The *leapfrog* scheme of the second order approximation

$$\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\tau} = \mathbf{F}(\mathbf{r}^n), \quad \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} = \mathbf{v}^{n+1/2}. \quad (38)$$

was used that widespread in the applications of the particle method to the plasma physics problems.

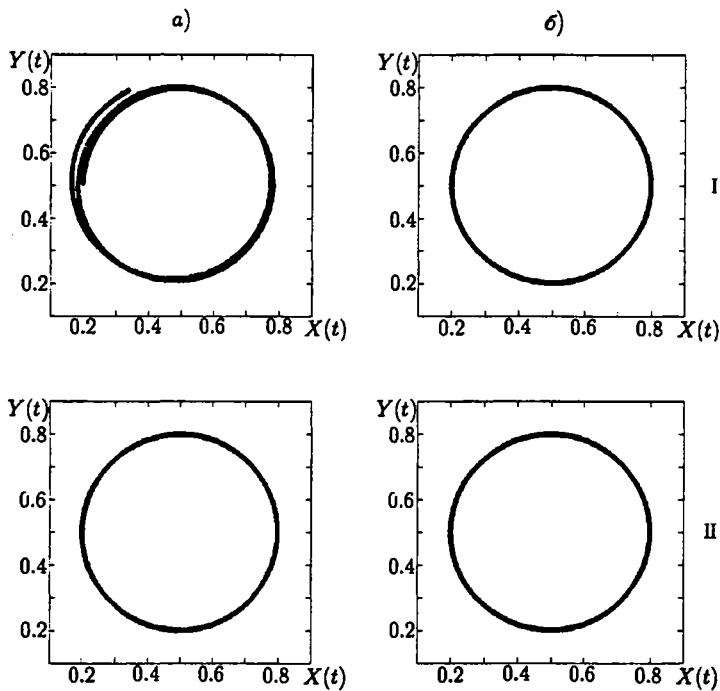


Fig. 17. The moving of "the test particle" in the centrical force problem:

I — 130 nodes, 222 elements. a) — calculation, b) — analytical solution;

II — 701 nodes, 1320 elements. a) — calculation, b) — analytical solution.

The comparison of the results of these numerical calculations with the exact analytical solution for unstructured meshes plotted in Fig. 13 is given in Fig. 17. The graphs demonstrate the improvement of the numerical solution convergence to the analytical one under transition from "the rough" to "the detailed" computational mesh.

2.3.4. The Lagrangian step with auxiliary Cartesian mesh

For improvement the procedures of interpolation "particles-mesh" and "mesh-particles" on the unstructured meshes we offer the modification in which model particles of the fixed finite dimensions are used. For the sake of simplicity we consider the two-dimensional case and assume that the distribution of vector density q_j of the features is uniform

over the particle area, that means $\mathbf{Q}_j = \mathbf{q}_j \cdot S_j$, where S_j is the area of the model particle with number j . The modeling domain is covered by the auxiliary Cartesian mesh with steps h_x , h_y (see, for example, [54]). In the suggested algorithm one can give the particle a form of any convex polygon. However, the structure of the data is the simplest for the particle of rectangular form with sides h_x , h_y which during the whole time of movement remain parallel to Cartesian coordinates.

In this case, the cell index in which the particle center with the coordinate $\mathbf{r}_j = (x_j, y_j)$ locates is determined in the following way:

$$I = \text{INT}[(x_j - x_0)/h_x] + 1, \quad J = \text{INT}[(y_j - y_0)/h_y] + 1. \quad (39)$$

Here operation INT means taking of the integer part of the expression confined in square brackets and x_0 , y_0 are the coordinates of the left lower node of the computational mesh.

The main procedure of the algorithm is the determination of the intersection area "particle-convex polygon" with elements of the unstructured mesh. The sequence of the steps realized in this procedure is shown in Fig. 18. Figuratively speaking, we take "scissors" and sequentially cut with their help along every side of the "element-triangle" and throw away that part of the "particle-rectangle" which remains outside the line of cutting. As the result (after three cuts) we receive a convex polygon which is an intersection of the model particle with the element of the unstructured mesh.

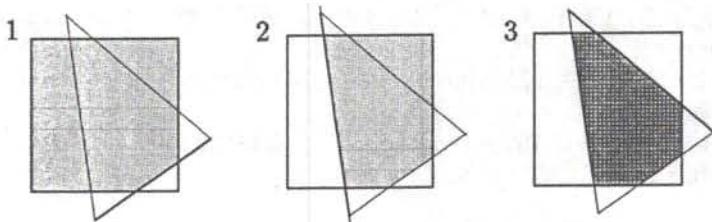


Fig. 18. The sequence of the steps at determination the intersection area of the rectangular particle with the triangular element of the unstructured mesh.

As the nodes of the mesh domain we consider the barycentres of elements. It is supposed that the contribution of the feature density of the particle \mathbf{q}_j into barycentre of the element C_i is proportional to the intersection area of the particle and this element

$$\tilde{\mathbf{q}}_{i,j} = \mathbf{q}_j \frac{S_{i,j}}{S_i}, \quad (40)$$

Here $S_{i,j}$ is the intersection area of the particle and element, S_i is the element area.

Summing up Eq. (40) on all particles we get

$$\tilde{\mathbf{q}}_i = \sum_{j=1}^N \tilde{\mathbf{q}}_{i,j} \frac{S_{i,j}}{S_i}. \quad (41)$$

Under interpolation "mesh-particles" the generalized force acting on the particle with number j is determined in such a way:

$$\mathbf{F}(\mathbf{r}_j, \mathbf{p}_j) = \frac{\sum_i S_{i,j} \mathbf{F}(\mathbf{r}_b^i)}{S_j}, \quad (42)$$

where \mathbf{r}_b^i are the coordinates of the barycentre C_i , and the sum is taken on all elements i with nonzero intersection area with j -particle.

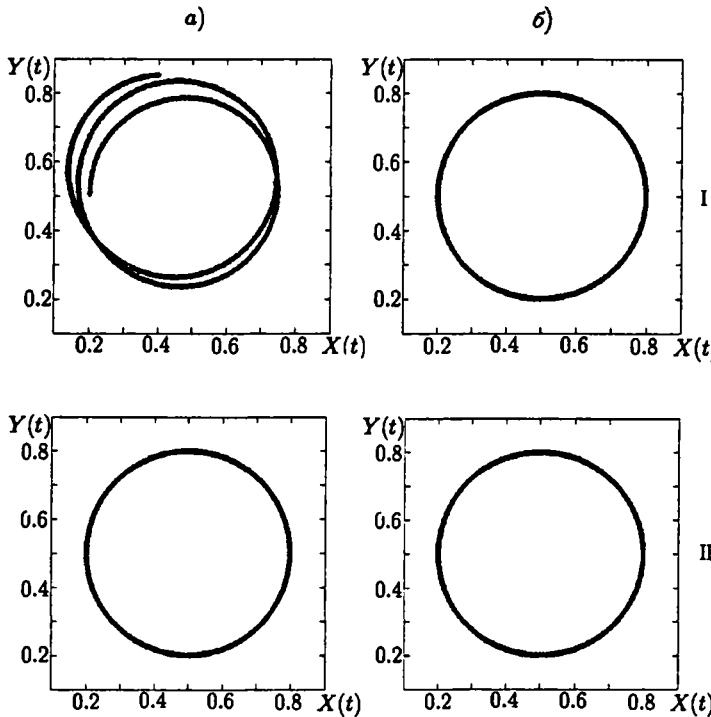


Fig. 19. The motion of the test particle of the finite dimension in the centrally symmetric domain:

I — 130 nodes, 222 elements. a) — calculation, b) — analytical solution.

II — 701 nodes, 1320 elements. a) — calculation, b) — analytical solution.

At the chosen dimensions the particle can intersect simultaneously with not more than four cells of the auxiliary Cartesian mesh analogously to PIC-model considered in Chapter 1. Having determined by Eq. (39) these cells, we then for every of them determine the indices of elements of the unstructured mesh which this particle intersects. After

that we calculate the intersection area of the particle with every of such elements. This algorithm allows to reduce the number of the necessary operations on localization approximately in $N_d/4$ times, where N_d is the number of cells of the Cartesian mesh. In this estimation it is supposed that the number of the cells of the unstructured mesh finding in each cell of auxiliary rectangular mesh differs in a small way from the corresponding mean value.

For testing the localization procedure of the particles and interpolation of the generalized forces into their location in the modified algorithm and comparison with the results of p. 2.3. the Kepler problem Eq. (37) was considered. The particle of the finite dimensions had the square form with the side $h = 0.05$. The solution of the Kepler problem for the meshes shown in Fig. 13 is represented in Fig. 19 (130 nodes, 222 elements; 701 nodes, 1320 elements correspondingly).

In the case of the rough mesh the particle dimension is compared with the typical dimension of a triangle element. This means that in the process of its moving the particle has nonzero intersection area (as a rule) with only one element of the mesh. As the force acting upon the particle is determined by the value of the force in the barycentre of this element, here we have the analogy with NGP-model for a Cartesian mesh. Therefore in this case the force interpolation procedure using linear basis functions (32) achieves better results in comparison with this method.

Under calculation on the more detailed mesh the force interpolation into the particle location is realized from several barycentres of the mesh analogously to CIC-model on the regular orthogonal mesh. In this case the results of numerical calculations is well agreed with the analytical solution.

2.4. The Euler step. The finite-volume method

For solving the equations of the Euler step for generalized fields on the unstructured meshes the finite-volume method or the finite-element method are used. Up to now the significant experience is accumulated and there is an extensive literature concerning the use of these methods on unstructured meshes for solving the systems of hydrodynamic equations of Euler and Navier-Stocks. The corresponding references can be found, for example, in the survey [51].

Let us briefly consider the peculiarities of the use of finite-volume method on the unstructured mesh on the example of the Maxwell electromagnetics equations. It should be noted that up to nowadays the series of algorithms is already described in literature based both on finite-element method [50, 60, 61] and on the finite-volume method of [62, 63]. The presented here algorithm was worked out in the work [56].

Following [56] let us write out the full system of vacuum Maxwell equations in the SI-system:

$$\frac{\partial \mathbf{E}}{\partial t} - c^2 \operatorname{rot} \mathbf{B} = -\frac{1}{\epsilon_0} \mathbf{j}, \quad (43)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \operatorname{rot} \mathbf{E} = 0, \quad (44)$$

$$\operatorname{div} \mathbf{E} = \frac{\rho}{\epsilon_0}, \quad (45)$$

$$\operatorname{div} \mathbf{B} = 0. \quad (46)$$

Under realization of finite-volume algorithm it's convenient to rewrite the initial system of equations (43)-(46) in the conservative form [56]:

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^3 \frac{\partial f_i(\mathbf{u})}{\partial x_i} = \mathbf{q}. \quad (47)$$

Here $\mathbf{x} = (x_1, x_2, x_3) = (x, y, z)$, $\mathbf{u}(\mathbf{x}, t) = (\mathbf{E}, \mathbf{B}) = (E_1, E_2, E_3, B_1, B_2, B_3)^T$. Physical fluxes $f_i(\mathbf{u})$ are defined by the formulas $f_i(\mathbf{u}) = K_i \mathbf{u}$, $i = 1, 2, 3$ where the block-structured matrices K_i have the dimension 6×6 and are determined according to formulas

$$K_i = \begin{pmatrix} 0 & -c^2 M_i \\ M_i & 0 \end{pmatrix}, \quad (48)$$

and

$$M_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, M_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, M_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (49)$$

The source term \mathbf{q} in the right-hand-side part has the form

$$\mathbf{q} = -\frac{1}{\epsilon_0} (j_1, j_2, j_3, 0, 0, 0)^T, \quad (50)$$

where $\mathbf{j} = (j_1, j_2, j_3)^T$ is a vector of current density.

Let us consider the calculation scheme of the finite-volume method supposing that the modeling domain consists of N non-overlapping elements C_i . Let us denote by h_i^n the average value over the cell C_i of any integrated function $h(\mathbf{x}, t)$ at the time moment $t = t^n = n\tau$. Then

$$h_i^n = \frac{1}{V_i} \int_{C_i} h(\mathbf{x}, t) dV, \quad (51)$$

where V_i is the volume of element C_i .

For solution of the system Eqs. (47) we use the scheme of splitting on physical processes. The approximated factorization is reduced to the successive integrating the following equations. First we solve the equation

$$\frac{\partial \mathbf{u}^{(1)}}{\partial t} = \mathbf{q}, \quad (52)$$

on one-half of the time step with the initial condition $\mathbf{u}^{(1)}(t^n) = \mathbf{u}^n$.

After this we solve the uniform conservation equation

$$\frac{\partial \mathbf{u}^{(2)}}{\partial t} + \sum_{i=1}^3 \frac{\partial f_i(\mathbf{u}^{(2)})}{\partial x_i} = 0, \quad (53)$$

on the full time step τ assuming that $\mathbf{u}^{(2)}(t^n) = \mathbf{u}^{(1)}(t^n + \frac{\tau}{2})$.

At the last step we integrate anew the ordinary differential equation

$$\frac{\partial \mathbf{u}^{(3)}}{\partial t} = \mathbf{q}, \quad (54)$$

on one-half of the step τ with the initial condition $\mathbf{u}^{(3)}(t^n + \frac{\tau}{2}) = \mathbf{u}^{(2)}(t^{n+1})$. Finally we receive

$$\mathbf{u}^{n+1} = \mathbf{u}^{(3)}(t^{n+1}) + O(\tau^2). \quad (55)$$

Integration of Eq. (52) over space-time volume $C_i \times [t^n, t^n + \frac{\tau}{2}]$ gives us the following equation

$$V_i \left[\mathbf{u}_i^{(1)} \left(t^n + \frac{\tau}{2} \right) - \mathbf{u}_i^{(1)}(t^n) \right] = \int_{t^n}^{t^n + \frac{\tau}{2}} \int_{C_i} \mathbf{q} dV dt,$$

which it can be approximated in the following way

$$\mathbf{u}_i^{(2)}(t^n) = \mathbf{u}_i^{(1)} \left(t^n + \frac{\Delta t}{2} \right) = \mathbf{u}_i^n + \frac{\tau}{2V_i} \int_{C_i} \mathbf{q} \left(\mathbf{x}, t^n + \frac{\tau}{2} \right) dV. \quad (56)$$

In the same way by integrating Eq. (54) we derive

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^{(3)}(t^{n+1}) = \mathbf{u}_i^{(2)}(t^{n+1}) + \frac{\tau}{2V_i} \int_{C_i} \mathbf{q} \left(\mathbf{x}, t^n + \frac{\tau}{2} \right) dV. \quad (57)$$

Applying the Gauss theorem to the Eq. (53), we carry out

$$V_i \left[(\mathbf{u}^{(2)})_i^{n+1} - (\mathbf{u}^{(2)})_i^n \right] = - \sum_{\alpha=1}^{\sigma_i} \int_{t^n}^{t^{n+1}} \int_{S_{i,\alpha}} \left(\sum_{j=1}^3 (n_j)_{i,\alpha} f_j(\mathbf{u}^{(2)}) \right) dS dt, \quad (58)$$

where $S_{i,\alpha}$ is the face with number α of the element C_i and σ_i is a total number of the faces of this element. Besides, $(n_j)_{i,\alpha}$ is j -component of the outward unit normal to the surface $S_{i,\alpha}$. Relation Eq. (58) we write down in the following form

$$(\mathbf{u}^{(2)})_i^{n+1} = (\mathbf{u}^{(2)})_i^n - \frac{\Delta t}{V_i} \sum_{\alpha=1}^{\sigma_i} G_{i,\alpha}^{n+1/2}, \quad (59)$$

where the flow $G_{i,\alpha}^{n+1/2}$ is approximated in the following way

$$G_{i,\alpha}^{n+1/2} \approx \frac{1}{\tau} \int_{t^n}^{t^{n+1}} \int_{S_{i,\alpha}} A_{i,\alpha} \mathbf{u}^{(2)} dS dt. \quad (60)$$

Here

$$A_{i,\alpha} = \sum_{j=1}^3 (n_j)_{i,\alpha} K_j. \quad (61)$$

Approximating the integral into Eq. (60) by formula of middle rectangles one can deduce

$$\mathbf{G}_{i,\alpha}^{n+1/2} \approx L_{i,\alpha} A_{i,\alpha} \mathbf{u}(M_{i,\alpha}, t^{n+1/2}), \quad (62)$$

where $M_{i,\alpha}$ and $L_{i,\alpha}$ correspondingly are the midpoint and the area of the face $S_{i,\alpha}$ (see Fig. 20).

For computing the value $\mathbf{u}(M_{i,\alpha}, t^{n+1/2})$ we suppose that at the time moment $t = t^n$ the value of k -component of gradient \mathbf{u} in the barycentre

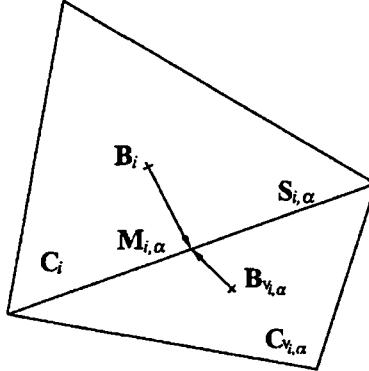


Fig. 20. Two neighboring computational cells C_i , $C_{\nu_i, \alpha}$ with barycentres B_i , $B_{\nu_i, \alpha}$ and the common side $S_{\nu_i, \alpha}$ with the midpoint $M_{i, \alpha}$.

B_i of the element C_i is known with the first-order of approximation.

$$(\mathbf{s}_i^n)_{kj} \approx \frac{\partial \mathbf{u}_k}{\partial x_j}(B_i, t^n). \quad (63)$$

Expanding $\mathbf{u}(M_{i,\alpha}, t^{n+1/2})$ into the truncated Taylor series with respect to t yields

$$\mathbf{u}_k(M_{i,\alpha}, t^{n+1/2}) = \mathbf{u}_k(M_{i,\alpha}, t^n) + \frac{\tau}{2} \frac{\partial \mathbf{u}_k}{\partial t}(M_{i,\alpha}, t^n) + O(\tau^2). \quad (64)$$

From Eqs. (48), (53) it follows that

$$\mathbf{u}_k(M_{i,\alpha}, t^n) = - \sum_{j=3}^3 \left[K_j \frac{\partial \mathbf{u}}{\partial x_j}(M_{i,\alpha}, t^n) \right]_k. \quad (65)$$

As far as values \mathbf{u}_k are determined in the element barycentres then expanding into the truncated Taylor series with respect to \mathbf{x} , one can obtain

$$\mathbf{u}_k(M_{i,\alpha}, t^n) = \mathbf{u}_k(B_i, t^n) + \sum_{j=3}^3 (\mathbf{B}_i \mathbf{M}_{i,\alpha})_j \frac{\partial \mathbf{u}_k}{\partial x_j}(B_i, t^n) + O(\Delta x^2), \quad (66)$$

where $(\mathbf{B}_i \mathbf{M}_{i,\alpha})_j$ is j -component of vector $(\mathbf{B}_i \mathbf{M}_{i,\alpha})$ (see Fig. 20). Taking into consideration the relations [56]

$$[\mathbf{u}_k]_i^n = \mathbf{u}_k(B_i, t^n) + O(\Delta x^2), \quad \frac{\partial \mathbf{u}_k}{\partial x_j}(M_{i,\alpha}, t^n) = (s_i^n)_{kj} + O(\Delta x), \quad (67)$$

we get an approximation of the k -component $\mathbf{u}(M_{i,\alpha}, t^{n+1/2})$ (68)

$$[\mathbf{u}_k]_{i,\alpha}^{n+} = [\mathbf{u}_k]_i^n + \sum_{j=3}^3 (\mathbf{B}_i \mathbf{M}_{i,\alpha})_j (s_i^n)_{kj} - \frac{\tau}{2} \sum_{j=3}^3 [K_j(s_i^n)_j]_k. \quad (68)$$

Analogously we deduce that

$$[\mathbf{u}_k]_{i,\alpha}^{n-} = [\mathbf{u}_k]_{\nu_i,\alpha}^n + \sum_{j=3}^3 (\mathbf{B}_{\nu_i,\alpha} \mathbf{M}_{i,\alpha})_j (s_{\nu_i,\alpha}^n)_{kj} - \frac{\tau}{2} \sum_{j=3}^3 [K_j(s_{\nu_i,\alpha}^n)_j]_k. \quad (69)$$

(69) The further step under construction of the numerical scheme for equations of the Euler step (here Maxwell equations) is the use of approximations (68), (69) for solving the Riemann problem. The local Riemann problem consists in the solution of the equation

$$\frac{\partial \mathbf{u}}{\partial t} + A \frac{\partial \mathbf{u}}{\partial \xi} = 0, \quad (70)$$

with the initial conditions

$$\mathbf{u}^0(\xi) = \mathbf{u}(\xi, 0) = \begin{cases} \mathbf{u}^+, & \xi < 0 \\ \mathbf{u}^-, & \xi > 0 \end{cases}, \quad (71)$$

where coordinate ξ is connected with the orientation of the normal vector at the face $S_{i,\alpha}$ (see Fig. 21).

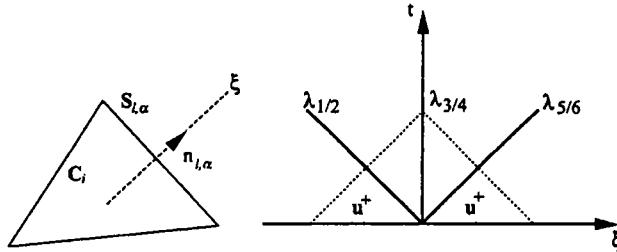


Fig. 21. Concerning the solution of the Riemann problem on the face $S_{i,\alpha}$ of the cell C_i in the (ξ, t) plane.

Here we drop the superscript n and the subscripts i, α of the solution \mathbf{u} and the matrix A (see, for instance, formulas (60), (61), (68), (69)). This problem can be solved exactly by the method of characteristics. Let us introduce the characteristic variable

$$\mathbf{v}(\xi, t) = R^{-1}\mathbf{u}(\xi, t), \quad (72)$$

where

$$R^{-1} = \begin{pmatrix} -\frac{n_3}{2c} & 0 & \frac{n_1}{2c} & -\frac{n_1 n_2}{2} & \frac{n_1^2 + n_3^2}{2} & -\frac{n_2 n_3}{2} \\ \frac{n_2^2 + n_3^2}{2} & -\frac{n_1 n_2}{2} & -\frac{n_1 n_3}{2} & 0 & -\frac{n_3 c}{2} & \frac{n_2 c}{2} \\ 0 & 0 & 0 & n_1 n_3 & n_2 n_3 & n_3^2 \\ n_1 n_2 & n_2^2 & n_2 n_3 & 0 & 0 & 0 \\ \frac{n_2^2 + n_3^2}{2} & -\frac{n_1 n_2}{2} & -\frac{n_1 n_3}{2} & 0 & \frac{n_3 c}{2} & -\frac{n_2 c}{2} \\ \frac{n_3}{2c} & 0 & -\frac{n_1}{2c} & -\frac{n_1 n_2}{2} & \frac{n_1^2 + n_3^2}{2} & -\frac{n_2 n_3}{2} \end{pmatrix} \quad (73)$$

is the matrix of left eigenvectors of the initial matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & n_3 c^2 & -n_2 c^2 \\ 0 & 0 & 0 & -n_3 c^2 & 0 & n_1 c^2 \\ 0 & 0 & 0 & n_2 c^2 & -n_1 c^2 & 0 \\ 0 & -n_3 & n_2 & 0 & 0 & 0 \\ n_3 & 0 & -n_1 & 0 & 0 & 0 \\ -n_2 & n_1 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (74)$$

Then it's possible to rewrite Eq. (70) in the form

$$\frac{\partial \mathbf{v}}{\partial t} + \Lambda \frac{\partial \mathbf{u}}{\partial \xi} = 0, \quad (75)$$

where the diagonal matrix

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6) = \text{diag}(-c, -c, 0, 0, c, c), \quad (76)$$

$\lambda_1, \dots, \lambda_6$ are eigenvalues of matrix A . Thus, Eqs. (75) split into six independent linear transport equations. The solutions of them are

$$\mathbf{v}_k(\xi, t) = v_k^0(\xi - \lambda_k t); \quad k = 1, \dots, 6, \quad (77)$$

with initial magnitudes $\mathbf{v}^0(\xi) = R^{-1}\mathbf{u}^0(\xi)$. At $\xi = 0$ the solution of Riemann problem is written in characteristic variables as follows (see Fig. 21)

$$\mathbf{v}(0, t) = (v_1^-(u^-), v_2^-(u^-), \bar{v}_3(\bar{u}), \bar{v}_4(\bar{u}), v_5^+(u^+), v_6^+(u^+))^T, \quad (78)$$

where $\mathbf{v}^\pm = R^{-1}\mathbf{u}^\pm$. As far as two eigenvalues of matrix A equal zero, $\lambda_3 = \lambda_4 \equiv 0$, then the solution depends only on the initial value $\bar{u} = u(0, 0)$. Then multiplying equation (78) by the matrix of the right eigenvectors

$$R = \begin{pmatrix} 0 & 1 & 0 & \frac{n_1}{n_2} & 1 & 0 \\ -\frac{n_3c}{n_1n_2} & -\frac{n_1^2 + n_3^2}{n_1n_2} & 0 & 1 & -\frac{n_1^2 + n_3^2}{n_1n_2} & \frac{n_3c}{n_1n_2} \\ \frac{c}{n_1} & \frac{n_3}{n_1} & 0 & \frac{n_3}{n_2} & \frac{n_3}{n_1} & -\frac{c}{n_1} \\ -\frac{n_2^2 + n_3^2}{n_1n_2} & -\frac{n_3}{n_1n_2c} & \frac{n_1}{n_3} & 0 & \frac{n_3}{n_1n_2c} & -\frac{n_2^2 + n_3^2}{n_1n_2} \\ 1 & 0 & \frac{n_2}{n_3} & 0 & 0 & 1 \\ \frac{n_3}{n_2} & \frac{1}{n_2c} & 1 & 0 & -\frac{1}{n_2c} & \frac{n_3}{n_2} \end{pmatrix}, \quad (79)$$

we obtain the solution of the Riemann problem at $\xi = 0$

$$\mathbf{u}(0, t) = R\mathbf{v}(0, t). \quad (80)$$

The explicit expression of this solution is written out in the paper [56]. Multiplying by the matrix A the last equation, we can show [56] that

$$A\mathbf{u}(0, t) = A^+\mathbf{u}^+ + A^-\mathbf{u}^-. \quad (81)$$

Here

$$A^\pm = \frac{1}{2} (A \pm |A|), \quad (82)$$

where

$$|A| = R|\Lambda|R^{-1} = \begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix}.$$

Here

$$D = \begin{pmatrix} (n_2^2 + n_3^2)c & -n_1n_2c & -n_1n_3c \\ -n_1n_2c & (n_1^2 + n_3^2)c & -n_2n_3c \\ -n_1n_3c & -n_2n_3c & (n_1^2 + n_2^2)c \end{pmatrix}.$$

Then we deduce that

$$A^\pm = \frac{1}{2} \begin{pmatrix} D^\pm & c^2\varepsilon \\ -\varepsilon & D^\pm \end{pmatrix},$$

where $D^\pm = \pm D$ and

$$\varepsilon = \begin{pmatrix} 0 & n_3 & -n_2 \\ -n_3 & 0 & n_1 \\ n_2 & -n_1 & 0 \end{pmatrix}.$$

By substituting Eq. (81) into Eq. (62) the compact flow-splitting form is carried out

$$\mathbf{G}_{i,\alpha}^{n+1/2} = L_{i,\alpha} (A_{i,\alpha}^+ \mathbf{u}_{i,\alpha}^{n+} + A_{i,\alpha}^- \mathbf{u}_{i,\alpha}^{n-}). \quad (83)$$

Formula (83) corresponds to the fact that the total flow $\mathbf{G}_{i,\alpha}^{n+1/2}$ combines the flow from the "right" with the positive eigenvalues and the flow from the "left" with the negative eigenvalues connected with matrices $A_{i,\alpha}^+$ and $A_{i,\alpha}^-$ correspondingly.

Chapter 3

The particle methods in gas dynamics

3.1. Introduction

Historically the method of particle-in-cell arose in the modeling of gas dynamics problems. The method was worked out in the middle of the 50s by a group lead by F. H. Harlow [9, 10] that worked in Los Alamos in the USA.

The success of the new method was stipulated by three factors. One was the necessity of finding an alternative to finite-difference methods to solve problems of great importance. These problems appeared in the modeling of the effects of powerful (nuclear) explosions on various media. Their characteristic peculiarities were large deformations, shock waves and fast intermixings of heterogeneous substances with changes in their topology (breakdown of continuity), which were reproduced poorly by finite-difference schemes.

Another important circumstance was the fact that the new method simply and effectively combined the positive qualities of pure Lagrangian and pure Euler algorithms.

A third factor was the possibility of using the powerful computers of the Los Alamos laboratory for the realization of the method.

Further, the method was developed by applied mathematicians of many schools in many countries. Thus, at the time some original programs realizing the Harlow method were worked out in the Russian nuclear centers, though known publications concerning them appeared much later [11].

As a result, at the beginning of the 60s authentic and descriptive computational data were obtained by the particle method for rather complicated dynamical problems of multicomponent continuous media.

At the same time, along with the attractive for numerical modeling peculiarities of the method, its characteristic defects were also revealed. In particular, it was the appearance of non-physical oscillations of the flow functions in subdomains with relatively little number of particles. Elimination of this effect was possible only by increasing of the total number of model particles. A specific for the particle-in-cell method dual Euler-Lagrangian mesh could be effectively realized only with a big storage capacity. All these factors required computational powers that exceeded the possibilities of most computers at the time. That is

why the mathematicians working in computational aerodynamics had a strong desire to modify the particle method in order to keep its positive peculiarities, but to reduce the requirements on the computer parameters. The first step in this direction was made also in the group of F.H.Harlow [64]. This modification was developed in papers [65, 66]. In the West this scheme is known under the abbreviation FLIC (Fluid-in-Cell). In Russia this direction is developed by the school of O.M. Belotserkovsky, and it was named coarse (large) particle method [13]. In this modification the characteristic features of the Harlow method, in particular, a splitting scheme for physical processes are kept. However, in fact a Lagrangian particle mesh is absent in the algorithm, and at the Lagrangian step continuous fluxes are calculated across the faces of Euler cells. Thus, the economical numerical method created allowed one to solve a lot of important for practice aerodynamical problems on available computers with frequency up to 1 MHz and memory storage of the order of 1 Mb.

Finally, for the solution of a certain class of problems, for example, of collision of solid bodies with the destruction of surfaces, when the domains of mutual penetration of heterogeneous materials are not too large, a method was proposed [67] that combined properly the particle-in-cell method and the mentioned modification of the FLIC-type.

In the formation of the modern concepts of the particle-in-cell methods an important role was played by of N.N. Yanenko and his followers [12]. In this paper for the first time a connection with scheme of splitting into physical processes was emphasized.

In p. 3.2 of the chapter the system of gas dynamics equations for two-dimensional flows as well as a method of splitting into physical processes in the form of approximated factorization is presented. In pp. 3.3 and 3.4. the Harlow scheme and the method of "coarse" particles (FLIC) are described and a combined method based on both schemes is also briefly considered. The last point contains an example of concrete realization of the particle-in-cell method that gives an idea of its requirements on computing resources, the computer time and the character of problems to be solved by the method.

3.2. Basic equations

3.2.1. Gas dynamics equations in divergent form

The system of differential equations of gas dynamics in divergent form consists of the continuity equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial y}(\rho v_y) = 0, \quad (1)$$

momentum equations

$$\frac{\partial}{\partial t}(\rho v_x) + \frac{\partial}{\partial x}(\rho v_x^2) + \frac{\partial}{\partial y}(\rho v_x v_y) = -\frac{\partial p}{\partial x}, \quad (2)$$

$$\frac{\partial}{\partial t}(\rho v_y) + \frac{\partial}{\partial x}(\rho v_x v_y) + \frac{\partial}{\partial y}(\rho v_y^2) = -\frac{\partial p}{\partial y} \quad (3)$$

and the energy equation

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x}[v_x(W + p)] + \frac{\partial}{\partial y}[v_y(W + p)] = 0. \quad (4)$$

Here t is the time, ρ is the gas density, v_x, v_y are the components of the gas velocity, p is the pressure, W is the density of full energy which can be expressed by the formula

$$W = \rho(\epsilon + \frac{v^2}{2}), \quad (5)$$

where ϵ is the specific internal gas energy. The system of Eqs. (1)-(4) represents the differential form of conservation laws of the mass, momentum and energy [1]. For closing the system it's necessary to add the thermodynamic relations connecting p and W . For example, for perfect gas they have the form [13]:

$$p = \rho RT, \quad \epsilon = c_v T,$$

where T is the gas temperature, R is the universal gas constant, c_v is the specific heat at the constant volume.

For the system of differential equations (1)-(4) in some domain Ω with the boundary S the initial-boundary problem is posed. At the initial time moment $t = 0$ in the solution domain the values of functions ρ, v_x, v_y, W are set. On the boundary S for $t \geq 0$ some boundary conditions are put.

3.2.2. The splitting scheme

The splitting scheme on physical processes is introduced here in the following way. Instead of the system Eqs. (1)-(4) let us consider two systems of differential equations

$$\left\{ \begin{array}{l} \frac{\partial \tilde{\rho}_1}{\partial t} = 0, \\ \frac{\partial}{\partial t}(\tilde{\rho}_1 \tilde{v}_{x,1}) = -\frac{\partial p}{\partial x}, \\ \frac{\partial}{\partial t}(\tilde{\rho}_1 \tilde{v}_{y,1}) = -\frac{\partial p}{\partial y}, \\ \frac{\partial \tilde{W}}{\partial t} = -\frac{\partial}{\partial x}(p \tilde{v}_{x,1}) - \frac{\partial}{\partial y}(p \tilde{v}_{y,1}). \end{array} \right. \quad (6)$$

$$\left\{ \begin{array}{l} \frac{\partial \rho_2}{\partial t} + \frac{\partial}{\partial x}(\rho_2 v_{x,2}) + \frac{\partial}{\partial y}(\rho_2 v_{y,2}) = 0, \\ \frac{\partial}{\partial t}(\rho_2 v_{x,2}) + \frac{\partial}{\partial x}(\rho_2 v_{x,2}^2) + \frac{\partial}{\partial y}(\rho_2 v_{x,2} v_{y,2}) = 0, \\ \frac{\partial}{\partial t}(\rho_2 v_{y,2}) + \frac{\partial}{\partial x}(\rho_2 v_{x,2} v_{y,2}) + \frac{\partial}{\partial y}(\rho_2 v_{y,2}^2) = 0, \\ \frac{\partial W_2}{\partial t} + \frac{\partial}{\partial x}(v_{x,2} W_2) + \frac{\partial}{\partial y}(v_{y,2} W_2) = 0. \end{array} \right. \quad (7)$$

The system Eqs. (6) is derived from Eqs. (1)-(4) if we omit in them the divergent terms of the mass density, the momentum components and energy fluxes. From the physical point of view Eqs. (6) describe process of changing gas parameters in the flow domain under boundary pressure forces. In turn, the system Eqs. (7) containing divergent terms corresponds to the process of the convective transport of gas dynamics parameters. Thus, the splitting selects in the gas dynamics two physical processes. In this case, the approximated factorization of the initial system Eqs. (1)-(4) is reached on every time step τ . This means that at the recurrent step τ the solving of system Eqs. (1)-(4) is obtained by two steps and is reduced to the successive solving of systems Eqs. (6) and Eqs. (7). It's obvious that system Eqs. (6) corresponds to the Euler step, and system Eqs. (7) corresponds to the Lagrangian step. The initial values for the Euler step are taken from the previous time step as follows

$$\tilde{f}_1(n\tau) = f(n\tau),$$

where f is any unknown function from system Eqs. (6), n is the number of the step.

From the results of the Euler step calculation the particle characteristics needed for the Lagrangian step are obtained by the interpolation procedure "mesh-particles".

3.3. The realization of the method

3.3.1. Harlow scheme

In the coordinate plane (x, y) we introduce the Euler mesh of nodes $x_i = ih_1, y_k = kh_2$ formed by the centers of Cartesian cells with sides h_1 and h_2 . The mesh flow domain we denote by Ω_h . In this case the numbers of the cells and indices of mesh functions are determined by the cell indices.

The Euler step.

Up to the beginning of the step on n -time layer, that means at the time moment t^n , for all cells (i, k) the values $(\rho, u, v, \epsilon, p)_{i,k}^n$ are known.

The step is realized by finite-difference schemes. In the papers [12, 13] different schemes are described for the first step of the particle

method. Let us consider one of these schemes.

It is conveniently to transform the system Eqs. (6) into the form:

$$\begin{aligned} \frac{\partial \tilde{\rho}_1}{\partial t} &= 0, \\ \tilde{\rho}_1 \frac{\partial \tilde{v}_{x,1}}{\partial t} + \frac{\partial p}{\partial x} &= 0, \\ \tilde{\rho}_1 \frac{\partial \tilde{v}_{y,1}}{\partial t} + \frac{\partial p}{\partial y} &= 0, \\ \tilde{\rho}_1 \frac{\partial \tilde{\epsilon}_1}{\partial t} + p \left(\frac{\partial \tilde{v}_{x,1}}{\partial x} + \frac{\partial \tilde{v}_{y,1}}{\partial y} \right) &= 0. \end{aligned} \quad (8)$$

For this system we write down the following scheme

$$\begin{aligned} \tilde{\rho}_{i,k} &= \rho_{i,k}^n, \\ \tilde{v}_{x,i,k} &= v_{x,i,k}^n - \frac{\tau}{h_1 \rho_{i,k}^n} (p_{i+1/2,k}^n - p_{i-1/2,k}^n), \\ \tilde{v}_{y,i,k} &= v_{y,i,k}^n - \frac{\tau}{h_2 \rho_{i,k}^n} (p_{i,k+1/2}^n - p_{i,k-1/2}^n), \\ \tilde{\epsilon}_{i,k} &= \epsilon_{i,k}^n - \frac{\tau p_{i,k}^n}{\rho_{i,k}^n} \left(\frac{\tilde{v}_{x,i+1/2,k} - \tilde{v}_{x,i-1/2,k}}{h_1} + \frac{\tilde{v}_{y,i,k+1/2} - \tilde{v}_{y,i,k-1/2}}{h_2} \right). \end{aligned} \quad (9)$$

It is seen that the gas density $\rho_{i,k}^n$ doesn't change at the Euler step. Variables \tilde{v}_x , \tilde{v}_y , p with fractional subscripts correspond to the values on the cell boundaries and are determined as a half-sum of values of corresponding functions in two neighbouring cells. For example,

$$p_{i+1/2,k} = (p_{i,k} + p_{i+1,k})/2.$$

It's easy to see that scheme Eqs. (2) approximates Eqs. (6) with the order $O(\tau, h_1^2, h_2^2)$. Here the second order on h_1, h_2 is got because of the use of central difference approximation for space derivatives.

The important property of this scheme (9) is its conservativeness. It is proved by the way of its reduction to the divergent form:

$$\begin{aligned} \frac{\tilde{\rho}_{i,k} \tilde{v}_{x,i,k} - \rho_{i,k}^n v_{x,i,k}^n}{\tau} + \frac{p_{i+1/2,k}^n - p_{i-1/2,k}^n}{h_1} &= 0, \\ \frac{\tilde{\rho}_{i,k} \tilde{v}_{y,i,k} - \rho_{i,k}^n v_{y,i,k}^n}{\tau} + \frac{p_{i,k+1/2}^n - p_{i,k-1/2}^n}{h_2} &= 0, \\ \frac{\tilde{W}_{i,k} - W_{i,k}^n}{\tau} + \frac{F_{i+1/2,k} - F_{i-1/2,k}}{h_1} + \frac{G_{i,k+1/2} - G_{i,k-1/2}}{h_2} &= 0, \end{aligned} \quad (10)$$

where

$$W_{i,k}^n = \rho_{i,k}^n \left[\epsilon_{i,k}^n + \frac{1}{2} ((v_{x,i,k}^n)^2 + (v_{y,i,k}^n)^2) \right].$$

$$\tilde{W}_{i,k} = \tilde{\rho}_{i,k} \left[\tilde{\varepsilon}_{i,k} + \frac{1}{2} (\tilde{v}_{x,i,k}^2 + \tilde{v}_{y,i,k}^2) \right],$$

$$F_{i-1/2,k} = (p_{i,k}^n \tilde{v}_{x,i-1,k} + p_{i-1,k}^n \tilde{v}_{x,i,k})/2,$$

$$G_{i,k-1/2} = (p_{i,k}^n \tilde{v}_{y,i,k-1} + p_{i,k-1}^n \tilde{v}_{y,i,k})/2.$$

The first two equations of the system Eqs. (10) are simply derived from the system Eqs. (9), and the third equation is deduced in the following way

$$\begin{aligned} \frac{\tilde{W}_{i,k} - W_{i,k}^n}{\tau} &= \frac{1}{\tau} \left(\tilde{\rho}_{i,k} \tilde{\varepsilon}_{i,k} - \rho_{i,k}^n \varepsilon_{i,k}^n \right) + \\ &+ \frac{1}{2\tau} \left[\tilde{\rho}_{i,k} \left(\tilde{v}_{x,i,k}^2 + \tilde{v}_{y,i,k}^2 \right) - \rho_{i,k}^n \left((v_{x,i,k}^n)^2 + (v_{y,i,k}^n)^2 \right) \right] = \end{aligned}$$

$$= \rho_{i,k}^n \frac{\tilde{\varepsilon}_{i,k} - \varepsilon_{i,k}^n}{\tau} + \frac{1}{2\tau} \rho_{i,k}^n \left(\tilde{v}_{x,i,k} + v_{x,i,k}^n \right) \left(\tilde{v}_{x,i,k} - v_{x,i,k}^n \right) +$$

$$+ \frac{1}{2\tau} \rho_{i,k}^n \left(\tilde{v}_{y,i,k} + v_{y,i,k}^n \right) \left(\tilde{v}_{y,i,k} - v_{y,i,k}^n \right) =$$

$$= \rho_{i,k}^n \left[- \frac{p_{i,k}^n}{\rho_{i,k}^n} \left(\frac{\bar{v}_{x,i+1/2,k} - \bar{v}_{x,i-1/2,k}}{h_1} + \frac{\bar{v}_{y,i,k+1/2} - \bar{v}_{y,i,k-1/2}}{h_2} \right) - \right.$$

$$\left. - \bar{v}_{x,i,k} \frac{1}{h_1 \rho_{i,k}^n} (p_{i+1/2,k}^n - p_{i-1/2,k}^n) - \bar{v}_{y,i,k} \frac{1}{h_2 \rho_{i,k}^n} (p_{i,k+1/2}^n - p_{i,k-1/2}^n) \right] =$$

$$= -p_{i,k}^n \left(\frac{\bar{v}_{x,i+1,k} - \bar{v}_{x,i-1,k}}{2h_1} + \frac{\bar{v}_{y,i,k+1} - \bar{v}_{y,i,k-1}}{2h_2} \right) -$$

$$- \bar{v}_{x,i,k} \frac{p_{i+1,k}^n - p_{i-1,k}^n}{2h_1} - \bar{v}_{y,i,k} \frac{p_{i,k+1} - p_{i,k-1}}{2h_2} =$$

$$= -\frac{1}{2h_1} \left[(p_{i,k}^n \bar{v}_{x,i+1,k} + p_{i+1,k}^n \bar{v}_{x,i,k}) - (p_{i-1,k}^n \bar{v}_{x,i,k} + p_{i,k}^n \bar{v}_{x,i-1,k}) \right] -$$

$$- \frac{1}{2h_2} \left[(p_{i,k}^n \bar{v}_{y,i,k+1} + p_{i,k+1}^n \bar{v}_{y,i,k}) - (p_{i,k-1}^n \bar{v}_{y,i,k} + p_{i,k}^n \bar{v}_{y,i,k-1}) \right].$$

The Lagrangian step.

The system of equations (7) of the second step can be presented in the vector form as follows

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial(\mathbf{q}\mathbf{U})}{\partial \mathbf{r}} = 0, \quad (11)$$

where $\mathbf{q} = (\rho_2, \rho\mathbf{U}_2, \rho_2\varepsilon_2)$ is a vector the components of which are the mass density ρ_2 , the momentum density $\rho_2\mathbf{U}_2$ and the density of internal energy $\rho_2\varepsilon_2$.

A solution of Eq. (11) is look for in the form

$$\mathbf{q}(\mathbf{r}, t) = \sum_{j=1}^N \mathbf{Q}_j R(\mathbf{r} - \mathbf{r}_j(t)), \quad (12)$$

where $R(\mathbf{r})$ is a particle kernel which characterizes the form of the particle, its dimension, and the density distribution of carried features in it; \mathbf{r} is the current space coordinate, \mathbf{r}_j is a radius-vector of j -particle center; the summation is fulfilled over all particles. Correspondingly

$$\mathbf{Q}_j = (M_j; M_j\mathbf{U}_j; M_j\varepsilon_j) \quad (13)$$

is a vector of features, carried by j -particle.

In accordance with (1.10) it is supposed that function $R(\mathbf{r} - \mathbf{r}_j)$ satisfies the norming condition

$$\int_{\Omega} d\mathbf{r} R(\mathbf{r} - \mathbf{r}_j) = 1. \quad (14)$$

It should be noted that

$$\frac{\partial}{\partial t} R(\mathbf{r} - \mathbf{r}_j(t)) = \frac{\partial R}{\partial \mathbf{r}_j} \frac{d\mathbf{r}_j}{dt} = -\frac{\partial R}{\partial \mathbf{r}} \frac{d\mathbf{r}_j}{dt}.$$

Let substitute Eq. (12) into system Eqs. (11) and integrate it on R^2 with the finite smooth weight function $\varphi(\mathbf{r})$. Using an integrating by parts one can derive

$$\begin{aligned} \int d\mathbf{r} \left(\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial(\mathbf{q}\mathbf{U})}{\partial \mathbf{r}} \right) \varphi(\mathbf{r}) &= \sum_{j=1}^N \mathbf{Q}_j \int d\mathbf{r} \left[-\frac{\partial R}{\partial \mathbf{r}} \frac{d\mathbf{r}_j}{dt} \varphi - R\mathbf{U} \frac{\partial \varphi}{\partial \mathbf{r}} \right] = \\ &= \sum_{j=1}^N \mathbf{Q}_j \int d\mathbf{r} R(\mathbf{r} - \mathbf{r}_j) \frac{\partial \varphi}{\partial \mathbf{r}} \left(-\frac{d\mathbf{r}_j}{dt} + \mathbf{U} \right) = 0. \end{aligned}$$

As $\varphi(\mathbf{r})$ is an arbitrary function, it means that this equality is satisfied identically if the particles move in accordance with the equations of motion

$$\frac{d\mathbf{r}_j(t)}{dt} = \mathbf{U}(\mathbf{r}_j). \quad (15)$$

In the Harlow method the features \mathbf{Q}_j carried by particles are determined by the values of mesh functions $\tilde{\mathbf{q}}_{ik} = (\tilde{u}_{ik}, \tilde{v}_{ik}, \tilde{\varepsilon}_{ik})$ calculated at the first step. For this it's necessary to introduce the interpolation procedure from the Euler mesh to the Lagrangian one. Individual masses of particles are usually assumed to be constant.

Let the recalculation from the Euler mesh to particles is define by the formula

$$\mathbf{Q}_j^{(n)} = h_1 h_2 \sum_{(i,k)} \tilde{\mathbf{q}}_{ik} S(\mathbf{r}_{ik} - \mathbf{r}_j^{(n)}), \quad j = 1, \dots, N, \quad (16)$$

where $S(\mathbf{r})$ is some interpolating function. It's quite obvious that the interpolation on particles must keep the total momentum and energy of fluid in the mesh flow domain Ω_h at the given time step. The total mass is kept automatically if the masses of individual particles don't change. For summing up the mesh functions it's convenient to use quadrature formulas of "middle" rectangles. Then the conservation laws under interpolation (16) are written down in the form

$$\sum_{j=1}^N \mathbf{Q}_j^{(n)} = h_1 h_2 \sum_{(i,k)} \tilde{\mathbf{q}}_{ik}. \quad (17)$$

It puts on the following condition on function S

$$\sum_{j=1}^N S(\mathbf{r}_{ik} - \mathbf{r}_j^{(n)}) = 1. \quad (18)$$

After calculating the arrays of particle characteristics Eq. (16) the new locations of the particles $\{\mathbf{r}_j^{(n+1)}\}_{j=1}^N$ on $(n+1)$ -time step are calculated. For solution of motion Eqs. (15) the simplest explicit scheme

$$\mathbf{r}_j^{n+1} = \mathbf{r}_j^n + \tau \mathbf{U}_j$$

is most often used, or in terms of components

$$x_j^{n+1} = x_j^n + \tau u_j,$$

$$y_j^{n+1} = y_j^n + \tau v_j.$$

The cycle is finished by the calculation of all mesh functions in $(n+1)$ -time moment. It's obvious that at the motion of particles their individual characteristics \mathbf{Q}_j are kept

$$\mathbf{Q}_j^{(n)} = \mathbf{Q}_j^{(n+1)}, \quad j = 1, \dots, N.$$

It corresponds to the divergent character of the system Eqs. (11). In order to make the Lagrangian step fully conservative its necessary under the inverse interpolation $\mathbf{Q}_j^{(n+1)}$ to the Euler mesh to satisfy also the conservation laws

$$\sum_{j=1}^N \mathbf{Q}_j^{(n+1)} = h_1 h_2 \sum_{i,k} \mathbf{q}_{ik}^{(n+1)}, \quad (19)$$

where

$$\mathbf{q}_{ik}^{(n+1)} = (\rho_{ik}^{(n+1)}, \rho_{ik}^{(n+1)} \mathbf{U}_{ik}^{(n+1)}, \rho_{ik}^{(n+1)} \varepsilon_{ik}^{(n+1)})$$

In accordance with Eq. (12) (see also p. 2 of Chapter 1) the new mesh densities are calculated using the formulas:

$$\mathbf{q}_{ik}^{(n+1)} = \sum_{j=1}^N \mathbf{Q}_j^{(n+1)} \bar{R}(\mathbf{r}_{ik} - \mathbf{r}_j^{(n+1)}), \quad (20)$$

where mesh kernel $\bar{R}(\mathbf{r}_{ik} - \mathbf{r}_j)$ is determined (see p. 2) as

$$\bar{R}(\mathbf{r}_{ik} - \mathbf{r}_j) = \frac{1}{h_1 h_2} \int_{\omega_{ik}} d\mathbf{r}' R(\mathbf{r}' - (\mathbf{r}_{ik} - \mathbf{r}_j)), \quad (21)$$

ω_{ik} is an Euler cell (i, k) . In practice PIC-model (1.32) is usually used for particles. In this case the justification of conservation laws (19) is checked directly. Really summing up (20) on Ω_h we have

$$\begin{aligned} h_1 h_2 \sum_{i,k} \mathbf{q}_{ik}^{(n+1)} &= h_1 h_2 \sum_{j=1}^N \sum_{i,k} \frac{1}{h_1 h_2} \int_{\omega_{ik}} d\mathbf{r}' R(\mathbf{r}' - (\mathbf{r}_{ik} \\ &- \mathbf{r}_j^{(n+1)})) = \sum_{j=1}^N \int_{\Omega} d\mathbf{r}' R(\mathbf{r}' - (\mathbf{r}_{ik} - \mathbf{r}_j^{(n+1)})) = \sum_{j=1}^N \mathbf{Q}_j^{(n+1)} \end{aligned}$$

Here the transition to integration on Ω is made with the accuracy up to the error of a quadrature formula of "middle" rectangles. Under this construction the Lagrangian step appears to be fully conservative.

The interpolating function S in Eq. (16) one can be chosen, for example, in the form

$$S(\mathbf{r}_{ik} - \mathbf{r}_j) = \frac{1}{\rho_{ik}} M_j R(\mathbf{r}_{ik} - \mathbf{r}_j). \quad (22)$$

It is seen that with taking into account Eq. (20), the necessary condition (18) for S will be fulfilled. It should be noted that for interpolation from the Euler mesh on the particles in relations (16) and (21) as well for the inverse interpolation (20) it can be used different mesh kernels. It's only necessary that every kernel would satisfy relations (14), (18), (21). At this it's quite obvious that conservation laws (17), (19) will be satisfied.

It should be paid attention to the following circumstance. If without moving of particles we interpolate mesh functions on particle locations by formulas (16) and then derive the inverse interpolation according to formulas (20), then the deduced anew values of mesh functions \mathbf{q}_{ik} will be not coincide with initial values $\tilde{\mathbf{q}}_{ik}$. Really, after substitution from Eq. (16) into Eq. (20) we have

$$\mathbf{q}_{ik} = \sum_{j=1}^N h_1 h_2 \sum_{l,m} \tilde{\mathbf{q}}_{lm} S(\mathbf{r}_{lm} - \mathbf{r}_j) \bar{R}(\mathbf{r}_{ik} - \mathbf{r}_j).$$

It's easy to verify that if only the carriers of functions $S(\mathbf{r})$ and $\bar{R}(\mathbf{r})$ are the cells of the Euler mesh and the mesh kernel has the form

$$\bar{R}(\mathbf{r}) = \frac{1}{h_1 h_2} \xi_{ik}(\mathbf{r}),$$

where $\xi_{ik}(\mathbf{r})$ is a characteristic function of the Euler mesh, then the procedure of double interpolation allows to receive

$$\mathbf{q}_{ik} = \tilde{\mathbf{q}}_{ik}.$$

3.3.2. The "coarse" particle method and FLIC-method

The method of "coarse" (large) particles and FLIC-method [65, 13] are the modifications of the Harlow scheme which was considered in the previous point. The authors of the methods tried to keep the basic advantages of the initial scheme but at the same time to increase its economicity. It was achieved by rejection from the Lagrangian mesh of particles on the transport step. That's why in these methods the splitting of medium into individual particles in fact is absent. Formally in the studies of this authors [13] it's accepted to call a "coarse" particle the contents of Euler cells. However such particles keep their individuality only at the Euler step. What concerns the abbreviation FLIC, in our opinion, the more successful reading of it could be "fluxes-in-cells", that reflects more exactly the sense of this modification of the Lagrangian step.

Nevertheless the methods have much in common with the Harlow scheme. The same splitting scheme of system Eqs. (1)-(4) on physical processes described in p. 3.2 is used. At every time step τ the methods are also carried out by two steps called Euler and Lagrangian.

The original studies of the authors developing the variants of this scheme differ in concrete details of finite-difference approximations realizing both steps. As the bibliography shows, this modification received the most wide spreading among the Russian computational mathematicians. Partly it is explained by the lag in growth of average records of the computer park in Russia which took place in the 70th - 80th. In this chapter below we consider finite-difference schemes of the method of "coarse" particles derived and approved in the studies of the authors [13] and their followers.

The Euler step.

This step in the method of "coarse" particles fully coincides with the Euler step in the Harlow scheme. It is realized on the basis of the same finite-difference schemes (9) and consequently it has the conservation property. The result of this step is in the computation of the sets of intermediate mesh functions

$$\tilde{\mathbf{q}}_{ik} = \{\rho_{i,k}^n, \rho_{i,k}^n \bar{u}_{i,k}, \rho_{i,k}^n \bar{v}_{i,k}, \rho_{i,k}^n \varepsilon_{i,k}\}, \quad (23)$$

determined in the centers of the Euler cells. In the authors' terminology these sets determine the state of "coarse" particles before the beginning of the Lagrangian step.

The Lagrangian step.

In order to deduce the finite-difference scheme used at this step we complete all mesh functions from Eq. (23) up to piecewise constant ones in every Euler cell ω_{ik} . Let integrate the equations of the Lagrangian step in the form (11) on area ω_{ik} . We obtain

$$h_1 h_2 \frac{\partial \mathbf{q}_{i,k}}{\partial t} = -h_2 (\Delta \mathbf{q}_{i+1/2,k} - \Delta \mathbf{q}_{i-1/2,k}) \\ - h_1 (\Delta \mathbf{q}_{i,k+1/2} - \mathbf{q}_{i,k-1/2}). \quad (24)$$

Here the algebraic sum of the fluxes of gas dynamical parameters across the cell boundaries stands at the right side. It appears from the divergent term of the system Eqs. (11). Half-integer indices denote the values corresponding to the cell faces.

Let approximate Eq. (24) by the explicit on time finite-difference scheme. Taking into consideration the initial state of the cell (23) we get

$$\mathbf{q}_{i,k}^{p+1} = \tilde{\mathbf{q}}_{i,k} - \frac{\tau}{h_1} (\Delta \tilde{\mathbf{q}}_{i+1/2,k} - \Delta \tilde{\mathbf{q}}_{i-1/2,k}) - \\ \frac{\tau}{h_2} (\Delta \tilde{\mathbf{q}}_{i,k+1/2} - \Delta \tilde{\mathbf{q}}_{i,k-1/2}). \quad (25)$$

For correspondence to the physical gas flow the fluxes of gas dynamical values are calculated with taking into account the sign of velocity normal to the boundary of the cell

$$\Delta \tilde{\mathbf{q}}_{i+1/2,k} = \begin{cases} \tilde{\mathbf{q}}_{i,k} \tilde{u}_{i+1/2,k}, & \tilde{u}_{i+1/2,k} > 0, \\ \tilde{\mathbf{q}}_{i+1,k} \tilde{u}_{i+1/2,k}, & \tilde{u}_{i+1/2,k} < 0. \end{cases} \quad (26)$$

$$\Delta \tilde{\mathbf{q}}_{i,k+1/2} = \begin{cases} \tilde{\mathbf{q}}_{i,k} \tilde{v}_{i,k+1/2}, & \tilde{v}_{i,k+1/2} > 0, \\ \tilde{\mathbf{q}}_{i,k+1} \tilde{v}_{i,k+1/2}, & \tilde{v}_{i,k+1/2} < 0. \end{cases} \quad (27)$$

Normal velocity components on the cell boundary equal the average values of corresponding velocities in adjacent cells, that means

$$\tilde{u}_{i+1/2,k} = \frac{1}{2} (\tilde{u}_{i+1,k} + \tilde{u}_{i,k}), \quad \tilde{v}_{i,k+1/2} = \frac{1}{2} (\tilde{v}_{i,k+1} + \tilde{v}_{i,k}),$$

Relations (25)-(27) determine the new state of "coarse" particles (Euler cells) in $(n+1)$ -time moment.

$$\mathbf{q}_{i,k}^{n+1} = \{\rho_{i,k}^{n+1}, \rho_{i,k}^{n+1} u_{i,k}^{n+1}, \rho_{i,k}^{n+1} v_{i,k}^{n+1}, \rho_{i,k}^{n+1} \epsilon_{i,k}^{n+1}\}.$$

By this the calculation cycle on the $(n+1)$ -time step is finished.

Summing up Eq. (25) by the quadrature formula of "middle" rectangles over the whole mesh flow domain Ω_h , we carry out

$$\mathbf{Q}^{(n+1)} = \tilde{\mathbf{Q}} + \sum_{\Gamma_h} \Delta \mathbf{Q}.$$

Here \mathbf{Q} is a vector of total characteristics of gas in domain Ω_h — mass M , momentum MU and internal energy E . The sum at the right side means the algebraic sum of the full flows of these values across the outer boundary Γ_h of domain. It's obvious that the full sum of the fluxes across the internal boundaries of Euler cells equals zero. It means [68]

that the scheme (25) of the Lagrangian step is divergent (conservative). Thus, the method of "coarse" particles just as the Harlow scheme is fully conservative. It should be noted that both methods are realized by the schemes explicit on time. The scheme (25) has the order of approximation $O(\tau, h_1, h_2)$.

At the large number of particles the Lagrangian step in the Harlow scheme approximates the initial system (7) with the same order [11]. Boundary conditions in both methods are realized analogously, the only difference is in the fact that "coarse" particles don't leave the flow domain, and that simplifies the realization of boundary conditions. Thus, we can conclude that both considered numerical schemes have the same properties of approximation and stability.

The order of approximation on space variables in Eq. (25) can be increased up to the second one calculating the fluxes across the boundaries by formulas of the second order of accuracy. For this purpose in the expressions for $\Delta\tilde{q}$ the boundary values of the functions are determined by the expansion in the truncated Taylor series up to the second order in the centers of the cells

$$\tilde{u}_{i+1/2,k} \approx \tilde{u}_{i,k} + \frac{h_1}{2} \left(\frac{\partial \tilde{u}_{i,k}}{\partial x} \right) \approx \tilde{u}_{i,k} + \frac{\tilde{u}_{i+1,k} - \tilde{u}_{i-1,k}}{4}.$$

Then formula (26) takes the form:

$$\Delta q_{i+1/2,k} = \begin{cases} \left(\tilde{q}_{i,k} + \frac{\tilde{q}_{i+1,k} - \tilde{q}_{i-1,k}}{4} \right) \left(\tilde{u}_{i,k} + \frac{\tilde{u}_{i+1,k} - \tilde{u}_{i-1,k}}{4} \right), \\ \tilde{u}_{i+1/2,k} > 0; \\ \left(\tilde{q}_{i+1,k} - \frac{\tilde{q}_{i+2,k} - \tilde{q}_{i,k}}{4} \right) \left(\tilde{u}_{i+1,k} - \frac{\tilde{u}_{i+2,k} - \tilde{u}_{i,k}}{4} \right), \\ \tilde{u}_{i+1/2,k} < 0; \end{cases}$$

The similar formula takes place for $\Delta q_{i,k+1/2}$. The using of these expressions reduces the stability of the algorithm of the second step which doesn't compensate already the instability of the first step. That's why in the formulas of the first step the quantity r is added to the pressure that introduces the artificial viscosity. Usually it has the form [13]:

$$r = \begin{cases} -\mu_0 C h \frac{\partial u}{\partial s}, & \frac{\partial u}{\partial s} < 0; \\ 0, & \frac{\partial u}{\partial s} \geq 0; \end{cases}$$

Where C is the sound velocity, h is the cell dimension in the direction s and the value μ_0 is chosen in such a way that the width of the shock layer would equal $(4 \div 6)h$.

The main advantage of the method of "coarse" particles just as of the FLIC-method as compared with Harlow particle-in-cell method is their greater economy. In 60-years it allowed to use it for solving of many problems of computational aerodynamics very important for applications [13]. Of course, the rejection from particles at the Lagrangian step allows to use the method of "coarse" particles (FLIC) only for homogeneous media with moderate deformations. For example, in contrast to the Harlow method they are unfitted for numerical visualization of motion of the contact surfaces in multicomponent media.

3.4. The combined particle method

As it was stated above, the particle-in-cell method allows to model successfully the motion of multicomponent media with such complicated phenomena as collapse or separation of heterogeneous materials, the formation and permutation of contact and free boundaries, internal mixing. However, the method has also some defects which continue its advantages. The presentation of continuous medium by the set of discrete particles leads to non-physical oscillations of flow functions and insufficient accuracy of computation of local characteristics. These effects are reduced with increasing of the average number of particles in the cell, but it's impossible to eliminate them completely. Besides, for realization of the method the high power computers are necessary.

These deficiencies were overcome in the "coarse" particle (FLIC-) method. But the latter being in its essence the finite-difference scheme is considerably devoid of the advantages of the particle-in-cell method. In particular, the modeling of inhomogeneous intermixing flows of multicomponent media in the presence of mobile contact surfaces and free boundaries presents the significant difficulties for this modification.

Outcoming from the proximity of PIC-method and the method of "coarse particles" using the identical splitting schemes and the similar realization of the Euler step, in the paper [67] the combined algorithm was proposed which unites both methods in one approach. The basic idea of the combination is contained in the fact that the "coarse" particle method is used in subdomains of the computational domain with homogeneous substance (material), but in the vicinity of mobile free and contact boundaries PIC-method is used.

All Euler cells in the coordinate domain are subdivided into three types:

- empty cells (Z-type);
- cells of the *P*-type in which the inhomogeneous substance is represented by the set of (point) particles and total mass in the cell equals the sum of masses of particles located in it;
- cells of the *F*-type with the homogeneous substance.

At the initial time moment the cells of the *F*-type are located in sub-

domains with homogeneous medium. The cells intersected by contact surfaces and free boundaries have the *P*-type (see Fig. 22).

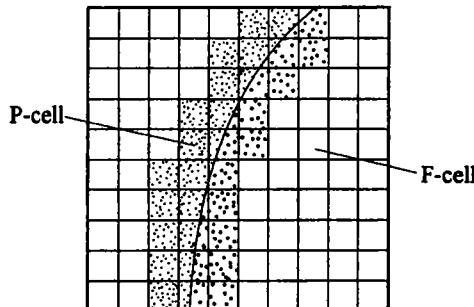


Fig. 22. The computational domain and the initial location of particles.

Just as in the schemes described above in this chapter, the initial system of gas dynamics equations is splitted on physical processes in two subsystems. On the Euler mesh the mesh functions of the velocity components (u, v) and pressure p are determined. Besides, for every substance contained in the cell its mass density ρ_α and specific internal energy ε_α are also assigned. The calculation is made cycle by cycle on time realized at two steps. At the first Euler step with initial data defined by values of all mesh functions at the time moment t^n the subsystem Eqs. (8) is solved by the finite-difference method (9) as it is described in p. 3.2.

At the second Lagrangian step cells of the *F*-type are processed by the scheme of the "coarse" particle method. Using formulas (26)-(27) the fluxes of mass, momentum and full energy are determined across the cell boundaries. In this case, if the substance passes into the cell of the *P*-type, then new particles are generated with total mass equaled to the mass of overflowed substance. At calculations of multicomponent media, if the substance passes into the cell of the *F*-type containing the substance of other species, then this cell is transformed into the cell of the *P*-type with particles of two species the total characteristics of which satisfy local conservation laws.

For treatment of *P*-type cells at the Lagrangian step the algorithm of the Harlow scheme is used. At intersection the boundaries of the *F*-type cells by the particles, carried by them fluxes of mass, momentum and full energy are calculated. If the particle enters the cell of the *F*-type containing the same substance, then its mass, momentum and full energy are summed up with the corresponding characteristics of this cell. In this case, the particle itself is excluded from the further calculation. If the substance in the *F*-type cell differs from the substance of the incoming particle, then the *F*-type cell is transformed into the *P*-type cell, and the substances in it are changed into the set of particles with

taking into account the satisfaction of local conservation laws. If in the process of motion in the *P*-type cell the particles only of one species remain and the neighbouring cells are not empty (that means, the cells are not on the free boundary), then this cell transforms into the *F*-type cell. The mass of this cell equals the total mass of the particles containing in it and these particles are excluded from the calculation. The excluded particles release the computer memory which can be used for the new formed particles.

After finishing the Lagrangian step in cells of both types the values of mesh flow functions at the time moment t^{n+1} are calculated by the methods described above for each of united algorithms. Under considering the motion of multicomponent media the cells appear which contain the mixture of several substances. The pressure of the components in these cells is determined from the condition of continuity of the full pressure on the contact boundary and the additivity of specific volumes. Thus, for the determination of pressure p and the volume concentration the system of the following equations takes place

$$\sum_{\alpha=1}^N p_\alpha = p, \quad \sum_{\alpha=1}^N \sigma_\alpha = 1,$$

where N is the quantity of species in the cell, σ_α is the volume concentration of the substance of the species α . Under determination of the increment of specific internal energy of every particle different assumptions can be used. Usually it is assumed that these increments are identical.

The derived in [67] combined method significantly reduces the requirements for the resource of computer memory in comparison with the pure PIC-method. In that time this property allowed us to calculate the problems of the motion of contact surfaces and free boundaries in multicomponent media, and also of the collision of bodies with the destruction on the average capacity computers.

3.5. The example of application

Here we present an example of calculation by particle method of the axially symmetric motion of gases under the action of the applied pressure [69]. The interaction of two gases with different densities and state equations

$$p_\alpha = (\gamma - 1) \epsilon_\alpha \rho_\alpha, \quad \alpha = 1, 2.$$

are considered. The computational domain had the form of a rectangle with radius $R = 1.4$ and height $H = 1.1$ (Fig. 23a). In the internal part of the domain with radius $R_1 = 1$ and the height $H_1 = 1$ the gas with uniform density 0.07 was situated and the rest subdomain contained the gas with density 11.4. At the initial time moment the gases were in equilibrium, immovable and had zero temperature. The constant

pressure $p = 1$ acted on the outer boundary of the domain occupied by gases during the whole time of the calculation. The calculation domain splitted into square cells with the step $h = 0.05$. The total number of cells equaled $28 \times 22 = 616$. The first gas occupied the internal subdomain with $20 \times 20 = 400$ cells; in every of these cells four particles were placed at the initial moment. The second gas occupied outer subdomain with 216 cells, where 9 particles were in each cell.

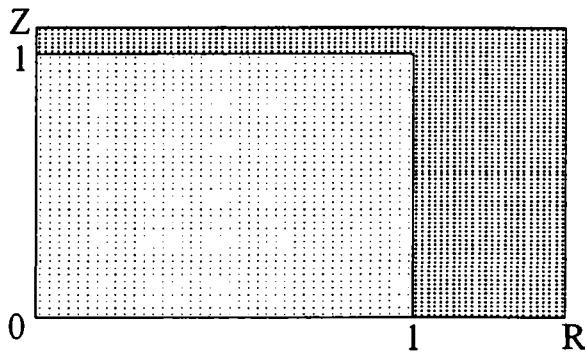


Fig. 23a. The computational domain and the initial particle distribution.

The particles in the cells were located uniformly and had the different mass proportional to the gas mass and to the particle distance from the axis. The total number of model particles was 3544. The symmetry conditions were set on the boundary $r = 0$ and $z = 0$: the normal velocity components and the derivatives of scalar functions with respect to normal equaled zero. On the rest boundaries the pressure was set under which action the gas was pressed.

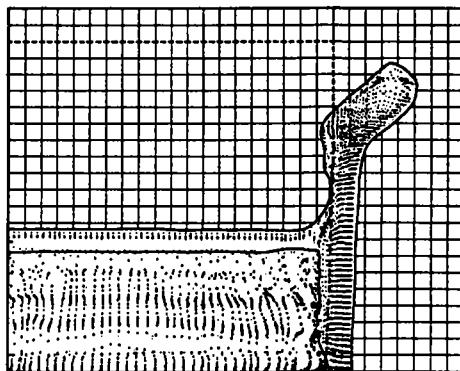


Fig. 23b. The location of particles, gas boundary and the contact boundary at the time moment $t = 1.3$.

Simultaneously with this the outer boundary displaced. In this case, the difficulties appeared concerning the determination of this boundary and also the use of external pressure under the calculation of the particle motion. In the calculations the boundary location was determined with using of empty cells: if the cell bordered with the empty one, then the pressure in it was assumed as equal to the outward applied pressure.

As the result the boundary was approximated by the graduated curve consisting of the coordinate line segments. It led to the fact that the boundary moved jump-wise: until all the particles would go from the boundary cell, the boundary would not move. The velocity in the boundary cells under the given boundary pressure was assumed equal to the velocity of the neighbouring internal cell.

In Fig. 23b the position of gases is shown at the time moment $t = 1.3$. The spatial separation of gases and also the position of the outer boundary are seen clear. The initial position of the less dense gas is shown by the dashed line.

Chapter 4

Vortex-in-cell methods

4.1. Introduction

There are numerous examples of fluid and gas flows that have in their larger part the character of a potential (irrotational) flow into which the small domains of a concentrated vorticity are imbedded. In this case, one of the dimensions of such domains is significantly less than the other characteristic scales of the flow. Examples are: boundary layers near solid surfaces, free shear layers, shed vortex sheets of airplane wings and propellers, aerodynamic wakes past bodies, cyclones in the Earth's atmosphere, etc. When the stabilizing influence of fluid viscosity is not enough, Kelvin-Helmholtz instabilities develop in similar vortex formations. As a result, connected vorticity domains can be divided into separate "islets" surrounded by an irrotational fluid. In contrast to particles of the Harlow particle method considered in the previous chapter these vortex particles are fully real discrete objects that have their individual characteristics. In theoretical hydrodynamics [1, 14] classic results are known concerning the dynamics of simplest isolated vortices and their systems. This coincidence in combination with a practical interest in such flows fully explains the early appearance of imitative vortex particle methods which modeled directly some of these flows. For the first time an imitative approach was used in paper [16] for the calculation of the vortex sheet roll-up into separate vortices. One can think that historically this was the first computation by the particle method. The formation of modern vortex particle-in-cell methods known in the literature as "vortex-in-cell" (VIC) was realized under a strong influence of similar methods for plasma physics problems [7, 20]. This influence became clear first of all in a methodical approach (compare Chapter 5). However, differences in the physical statements and mathematical apparatus for plasma and vortex fluid require a separate record of VIC-methods which is the subject of this chapter. A significant difference in the vorticity dynamics of plane and space flows leads to an additional division of vortex methods according to dimensionality.

Usually methods of discrete vortices are used for problems of incompressible and often inviscid flows. The presentation in this chapter is carried out for an incompressible fluid. Nevertheless, it should be noted that the methods considered below can be generalized for the case of a compressible gas. For the two-dimensional case this generalization was derived in paper [70].

4.2. Vorticity dynamics in two-dimensional flows

4.2.1. Basic equations

Two-dimensional flows of incompressible viscous fluid can be described [1] by the following system of differential equations

$$\frac{\partial \omega}{\partial t} + \mathbf{U} \frac{\partial \omega}{\partial \mathbf{r}} = Re^{-1} \Delta \omega, \quad (1)$$

$$\Delta \psi = -\omega, \quad (2)$$

$$u = -\frac{\partial \psi}{\partial y}, v = \frac{\partial \psi}{\partial x}. \quad (3)$$

Here $\mathbf{U} = (u, v)$ is a flow velocity vector, $\mathbf{r} = (x, y)$ is a radius-vector of the point at the stream plane, ψ is a stream function, ω is a scalar vorticity (pseudovector) oriented along the unit normal \mathbf{e}_z to the (x, y) -plane, Δ is a two-dimensional Laplace operator, and Re is the Reynolds number.

For system Eqs. (1)-(3) the initial-boundary problem is put which in terms of velocity field has the form

$$\begin{aligned} \mathbf{U}(\mathbf{r}, 0)|_{t=0} &= \mathbf{U}(\mathbf{r}), \quad \mathbf{r} \in D, \\ \mathbf{U}(\mathbf{r}, t) &= \mathbf{U}(\mathbf{r}_s, t), \quad \mathbf{r} \in S. \end{aligned} \quad (4)$$

In Eq. (4) D is the flow domain, S is its boundary composed by parts penetrable and impenetrable for the fluid. The solution of Eq. (2) in domain D is expressed in terms of the Green's function of the Poisson equation. In the simplest case of the free stream when the fluid is at rest at infinity, the latter is written [15] as

$$G(\mathbf{r}, \mathbf{r}_1) = -(2\pi)^{-1} \ln |\mathbf{r} - \mathbf{r}_1|,$$

and the solution Eq. (2) is represented in the form

$$\psi = -\frac{1}{2\pi} \int d\mathbf{r}_1 \omega \ln |\mathbf{r} - \mathbf{r}_1|.$$

The flow velocity taking into account the relations (3) is expressed by the Biot-Savart integral [1]

$$\mathbf{U}(\mathbf{r}, t) = -\frac{1}{2\pi} \int d\mathbf{r}_1 \frac{(\mathbf{r} - \mathbf{r}_1) \times \mathbf{e}_z \omega(\mathbf{r}_1)}{|\mathbf{r} - \mathbf{r}_1|^2} \quad (5)$$

Here and further in this point the integrals are taken over the whole flow plane $R^2(x, y)$.

From the relations (3), (5) it follows that

$$\operatorname{div} \mathbf{U} = 0$$

and the left side in Eq. (1) can be written down in the divergent form.

4.2.2. The conservation laws

For two-dimensional free inviscid flows when the right side in the transport vorticity equation vanishes, there is an uncountable set of conservation laws of the form

$$I_\phi = \int d\mathbf{r} \Phi(\omega) = \text{const}, \quad (6)$$

where Φ is an arbitrary smooth function. Really,

$$\frac{dI_\phi}{dt} = \int d\mathbf{r} \frac{d\Phi}{d\omega} \left(\frac{\partial \omega}{\partial t} + \mathbf{U} \frac{\partial \omega}{\partial \mathbf{r}} \right) = 0.$$

From the set (6) the following invariants of movement have physical sense

$$I_1 = \int d\mathbf{r} \omega, \quad I_2 = \int d\mathbf{r} \times \mathbf{e}_z \omega,$$

$$I_3 = \int d\mathbf{r} \mathbf{r} \times \mathbf{r} \times \mathbf{e}_z \omega,$$

$$I_4 = -\frac{1}{2\pi} \int \int d\mathbf{r} d\mathbf{r}_1 \omega(\mathbf{r}) \omega(\mathbf{r}_1) \ln |\mathbf{r} - \mathbf{r}_1|. \quad (7)$$

They correspondingly express the conservation laws of the total vorticity, the momentum, the angular momentum and kinetic energy [14].

By the direct integration of Eq. (1) with the corresponding weight function it can be shown that I_1 and I_2 are invariants also for viscous fluid.

4.3. The vortex-in-cell method in two-dimensional case

4.3.1. Splitting on physical processes

The general scheme of the particle-in-cell method given in Chapter 1 is realized here in the following way. From Eqs. (1), (5) it is seen that in the two-dimensional case the evolution of scalar vorticity is determined by two physical processes: by the transport in the self-consistent flow velocity field \mathbf{U} induced by distribution ω and by viscous diffusion. Splitting on these processes in the limits of time step τ leads to two auxiliary problems of the form

$$\frac{\partial \tilde{\omega}_1}{\partial t} = -\frac{\partial \mathbf{U} \tilde{\omega}_1}{\partial \mathbf{r}} \equiv -D\tilde{\omega}_1, \quad \tilde{\omega}_1(p\tau) = \omega(p\tau), \quad (8)$$

$$\frac{\partial \tilde{\omega}_2}{\partial t} = -\frac{1}{Re} \Delta \tilde{\omega}_2 \equiv R\tilde{\omega}_2, \quad \tilde{\omega}_2(p\tau) = \tilde{\omega}_1((p+1)\tau). \quad (9)$$

4.3.2. The systems of discrete vortices and their properties

The vorticity field $\omega(\mathbf{r}, t)$ is approximated by system of N vortices [2]

$$\tilde{\omega}(\mathbf{r}, t) = \sum_{i=1}^N \Gamma_i R(\mathbf{r}, \mathbf{r}_i), \quad (10)$$

where Γ_i is a circulation [1] (total vorticity) of i -vortex particle located at the point $\mathbf{r}_i = (x_i, y_i)$, $R(\mathbf{r}, \mathbf{r}_i)$ is a kernel of vortex particle. As usual, the norming condition imposed on it takes the form

$$\int d\mathbf{r} R(\mathbf{r}, \mathbf{r}_1) = 1.$$

In this case, the normalization allows to reproduce the vorticity conservation law for the discrete vortex system

$$I_1 = \int d\mathbf{r} \omega(\mathbf{r}, t) = \sum_{i=1}^N \Gamma_i.$$

The systems of discrete vortices Eq. (10) in ideal fluid were studied by many authors (see, for example, the survey [2]). The analytical theory is most advanced for singular particles when

$$\tilde{\omega}(\mathbf{r}, t) = \sum \Gamma_i \delta(\mathbf{r} - \mathbf{r}_i(t)), \quad (11)$$

$\delta(\mathbf{r})$ is the two-dimensional Dirac delta function.

The particles in this case represent infinite rectilinear vortex filaments of finite circulation perpendicular to the stream plane.

The equations of motion of vortices are derived by the substitution $\tilde{\omega}(\mathbf{r}, t)$ into Eqs. (8), (5).

Multiplying Eq. (8) by smooth finite function $\varphi(\mathbf{r})$ and integrating over the flow plane with taking into account the form of Eq. (11) we carry out

$$\begin{aligned} \sum_{i=1}^N \Gamma_i \frac{d\varphi(\mathbf{r}_i)}{d\mathbf{r}_i} \left(\frac{d\mathbf{r}_i}{dt} - \mathbf{U}(\mathbf{r}_i) \right) &= 0 \\ \mathbf{U}(\mathbf{r}_i) &= -\frac{1}{2\pi} \sum_{j=1}^{N'} \Gamma_j \frac{(\mathbf{r} - \mathbf{r}_i) \times \mathbf{e}_z}{(\mathbf{r}_i - \mathbf{r}_j)^2}, \end{aligned}$$

where the prime on the summation sign means that there is no term $i = j$ in it. By virtue of arbitrariness $\varphi(\mathbf{r})$ from here the equations of the vortex motion are obtained as follows

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{U}(\mathbf{r}_i), \quad i = 1, \dots, N$$

It is directly checked that the vortex dynamics can be described in the Hamiltonian form

$$\frac{dx_i}{dt} = \frac{1}{\Gamma_i} \frac{\partial H_N}{\partial y_i}, \quad \frac{dy_i}{dt} = -\frac{1}{\Gamma_i} \frac{\partial H_N}{\partial x_i}, \quad i = 1, \dots, N. \quad (12)$$

Here Cartesian coordinates (x_i, y_i) of the point vortices on the plane are canonical variables conjugated with respect to the Hamiltonian

$$H_N = -\frac{1}{4\pi} \sum_{i \neq j=1}^N \Gamma_i \Gamma_j \ln |\mathbf{r}_i - \mathbf{r}_j|. \quad (13)$$

Expression Eq. (13) is the result of substitution the approximation Eq. (11) into the energy integral I_4 Eq. (7).

As it is known [59], the Hamiltonian structure allows to come to conclusions about some integral invariants in the corresponding dynamic system. In particular, if the Hamiltonian doesn't depend explicitly on time and $\partial H_N / \partial t = 0$, then directly from Eq. (12) follows

$$\begin{aligned} \frac{dH_N}{dt} &= \sum_{i=1}^N \left(\frac{\partial H_N}{\partial x_i} \frac{dx_i}{dt} + \frac{\partial H_N}{\partial y_i} \frac{dy_i}{dt} \right) = \\ &= \sum_{i=1}^N \Gamma_i^{-1} \left(\frac{\partial H_N}{\partial x_i} \frac{\partial H_N}{\partial y_i} - \frac{\partial H_N}{\partial y_i} \frac{\partial H_N}{\partial x_i} \right) = 0. \end{aligned}$$

That means that H_N is the integral invariant along the trajectories of system Eqs. (12). From Eq. (13) one can see that it corresponds to the discrete realization of the energy conservation law. From invariance H_N with respect to translations and rotations of the coordinate system [59] the existence of discrete analogs of integral invariants I_2, I_3 , in other words, of momentum and angular momentum conservation laws, follows [14].

However, just as in other particle methods, singular (point) vortices are seldom used in practical calculations because they have general for these cases defects. In particular, singularity of interaction at short distances causes instabilities and non-physical stochastization in the flow field. Strong fluctuations of vorticity and velocity appear at transitions from the Lagrangian mesh to the Euler mesh and backwards.

Modern vortex-in-cell methods developed under the direct influence of particle methods in plasma because of deep physical analogies between them [2, 7]. That's why the choice of the kernel form $R(\mathbf{r}, \mathbf{r}_1)$ was also often borrowed from the calculations of plasma dynamics problems. Just as in the models of plasma particles, the transition towards vortices of finite dimensions (vortex blobs or clouds) is explained, first of all, by the necessity to reduce the contribution of singular interaction at approaching of individual vortices so that the model flow field

¹It's interesting to note that the similar Hamilton system of the particle method arises in the magnetized plasma dynamics problem when the "guiding center" approximation [8] is considered. This system also appears under modeling by the particle-in-cell method of two-dimensional Galaxies [7].

would mostly depend on co-operative effects as it is expressed by the Biot-Savart integral (5).

Besides, by corresponding choosing of kernel $R(\mathbf{r}, \mathbf{r}_1)$ it is possible to increase the approximation order of the method as it was shown in Chapter 1. Under the definite conditions concerning the distribution form $R(\mathbf{r}, \mathbf{r}_1)$ it becomes possible to obtain the satisfaction of conservation laws. For example, for particle invariance with respect to translations and rotations of Cartesian coordinates that is necessary for conservation of integrals I_2, I_3 , it is usually assumed that

$$R(\mathbf{r}, \mathbf{r}_1) = 1/\delta^2 \gamma(|\mathbf{r} - \mathbf{r}_1|/\delta), \quad (14)$$

where parameter σ determining the characteristic scale of the kernel is introduced in such a way that to keep the usual norming condition. In particular, the Hauss distribution is used quite often [2]

$$R(|\mathbf{r}|) = \frac{1}{\pi\delta^2} \exp(-r^2/\delta^2), \quad (15)$$

though, first of all, it concerns purely Lagrangian methods which are not considered here.

Finally, in some cases the vortex particles with rather specific distribution were used that is connected with the reproduction of some hydrodynamic effects, for example, with modeling of the boundary layer near the wall [71] or the spectrum of free hydrodynamic turbulence [72].

As it was noted in the previous chapters, the introduction of the finite dimension particles with some distribution $R(\mathbf{r})$ of the carried feature (charge, mass, vorticity) is accompanied by the series of conditionalities. This, in turn, leads to additional non-correspondences between the physical nature of the problem and the computational model.

Usually model particles move and interact without changing their kernel $R(\mathbf{r})$. For model vortices, the motion and interaction of which are characterized by inviscid approximation (8), it contradicts to the known property [14] that vorticity in ideal fluid has to move with fluid. In other words, the vortex elements during all the time of the motion stay as if being "attached" to fluid particles and have their instantaneous velocities. That's why in reality vortex elements aren't deformed only in a uniform flow. Neglecting the model particle deformation, we therefore introduce non-physical distortions into the flow picture.

The other conditionality of the model is a determination of vortex particle velocity. This question is analogous to the calculation of the force that acts on the charge model particle in the electrostatic plasma problem (Chapter 5). For model vortices it is more often assumed that their velocity coincides with flow velocity in a geometric center of the vortex. Then the equations of motion for vortices with centers in points \mathbf{r}_i are written as follows

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{U}(\mathbf{r}_i, t), \quad i = 1, \dots, N \quad (16)$$

where \mathbf{U} is determined by Biot-Savart integral (5) over vorticity field of model particles (10).

In the other way the model particle velocity is obtained by the averaging $\bar{\mathbf{U}}(\mathbf{r}, t)$ over the vorticity distribution in the kernel

$$\bar{\mathbf{U}}(\mathbf{r}_i, t) = \int R(\mathbf{r}_i, \mathbf{r}) \mathbf{U}(\mathbf{r}, t) d\mathbf{r}.$$

Under this determination of the vortex velocity with using smooth kernels of the form (15), the dynamic equations of model system can again be presented in the Hamiltonian form (13) [2]. The Hamiltonian H_N , as above, is proportional to the energy of system and is derived by the substitution of the approximation (10) with kernel (15) into the energy integral I_4 .

The Hamiltonian dynamics of model particles in addition to satisfaction the discrete conservation laws gives us the possibility to use the apparatus of kinetic equations [7] for the analysis of computational approximation, stability, dispersion and dissipative properties of the method.

However, under numerical modeling, especially in the considered here schemes, specialists usually don't strive to use the properties of Hamiltonian dynamics. In practice a preference is given to the simple kernels, the choice of velocity in the particle center and the finite-difference schemes that are not high approximation order on time. The last circumstance usually leads to violation of energy integral I_4 invariance. Generally one can say that the balance between the theoretical fault lessness and the economicity of algorithms in practice usually removes towards the latter.

The kernel of PIC-model

$$R(\mathbf{r}, \mathbf{r}') = \begin{cases} (h_1 h_2)^{-1}, & |x - x'| \leq h_1/2 \\ & |y - y'| \leq h_2/2, \\ 0, & |x - x'| > h_1/2 \\ & |y - y'| > h_2/2, \end{cases} \quad (17)$$

in combination with the simple projection operator P onto the Euler mesh is used rather often (see also Chapters 1, 3, 5). Here h_1, h_2 are the steps of the Euler cells in the computational domain. It is seen that for (17) the norming condition is satisfied, that means that for the system of model vortices the total vorticity (integral I_1 in (7)) is invariant of motion. Besides, the dependence on the modulus of Cartesian coordinate difference means that $R(\mathbf{r}, \mathbf{r}')$ is invariant with respect to translations. It follows from here that I_2 is also an invariant. However I_3, I_4 are not already invariants.

The prevalence of such a superposition of smoothing and projection operators (see Chapters 1, 3, 5) is explained by the simplicity of interpolation from the Lagrangian mesh of vortices (in this case) to the Euler mesh of the nodes and backwards. The interpolation programs for particles with such a kernel are presented in Supplement B.

4.3.3. The computational cycle of the method

The computational cycle at the time step τ can be described in such a way (see also the scheme Fig. 24)

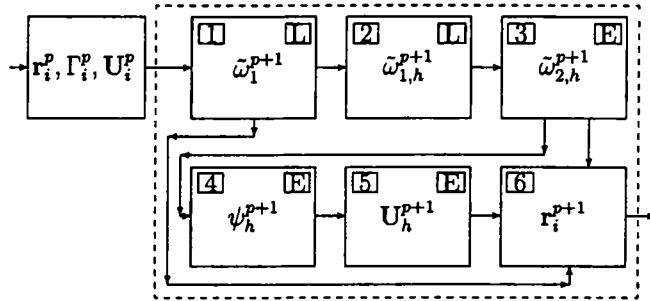


Fig. 24. The block-scheme of the vortex-in-cell method for two-dimensional flows. Blocks belonging to the Lagrangian and Euler steps are denoted by letters *L* and *E* correspondingly.

The Lagrangian step.

1. Before the beginning of the next $(p+1)$ -time step the locations of the Lagrangian vortex mesh (r_1^p, \dots, r_n^p), their circulations (strengths) ($\Gamma_1^p, \dots, \Gamma_n^p$) and the collection of values of flow velocities $\{U_i^p\}_{i=1}^N$ in location points are fixed. By the finite-difference approximation of the dynamic Eqs. (16) the new distribution of vorticity $\tilde{\omega}_1^{p+1}$ is calculated, that means the new coordinates of vortex particle centers are calculated. The simplest explicit scheme

$$\frac{x_i^{p+1} - x_i^p}{\tau} = u_i^p, \quad \frac{y_i^{p+1} - y_i^p}{\tau} = v_i^p, \quad i = 1, \dots, N. \quad (18)$$

is often used.

2. The vorticity mesh function $\{\tilde{\omega}_{1,h}^{p+1}\}$ is calculated. The interpolation from the Lagrangian mesh to the Euler one is determined by the choice of kernel $R(r)$ and by the projection operator P on Euler mesh. Let the Euler mesh be formed by the centers of Cartesian cells. In the case of kernel Eq. (17) every vortex with coordinates $r_n^{p+1} = (x_n^{p+1}, y_n^{p+1})$ represents a rectangle with sides h_1, h_2 and it contributes into the values of the vorticity mesh function in the four nearest nodes (see Fig. 25).

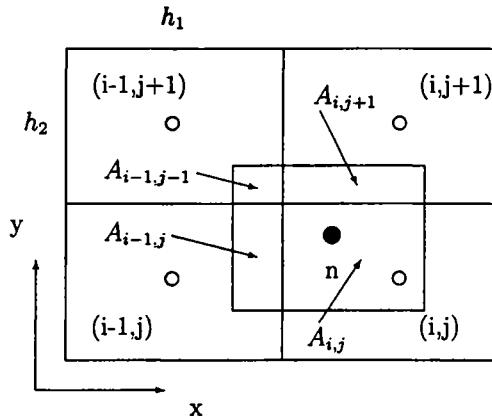


Fig. 25. The scheme of vorticity interpolation from the Lagrangian mesh of the vortices to the nodes of the Euler mesh.

As it is seen from Fig. 25, vorticity assigned to the l -node at $p + 1$ -time step is expressed as follows

$$\tilde{\omega}^{p+1}(l) = \frac{A_l}{h_1 h_2} \tilde{\omega}_{1,n}^{p+1}, \quad (19)$$

Then the Euler step is computed.

The Euler step.

3. Parabolic Eq. (9) at the time step τ is approximated by some economical difference scheme, for example, by the scheme of alternating direction method [68].

$$\begin{aligned} \frac{\tilde{\omega}_{2,ij}^{p+1} - \tilde{\omega}_{1,ij}^{p+1}}{\tau} &= \frac{1}{2}(\Lambda_{xx}\tilde{\omega}_{2,ij}^{p+1} + \Lambda_{yy}\tilde{\omega}_{1,ij}^{p+1}) \\ \frac{\tilde{\omega}_{2,ij}^{p+1} - \tilde{\omega}_{1,ij}^{p+1}}{\tau} &= \frac{1}{2}(\Lambda_{xx}\tilde{\omega}_{1,ij}^{p+1} + \Lambda_{yy}\tilde{\omega}_{2,ij}^{p+1}). \end{aligned} \quad (20)$$

4. By some difference approximation of the Poisson Eq. (2) on the Euler mesh the stream mesh function $\psi_{1,h}^{p+1}$ is calculated. In commercial codes special fast algorithms (Poisson solvers) are usually used for it [31]. If the flow can be embedded into the rectangular domain and the regular mesh can be used, then the Poisson equation is effectively solved by the Fast Fourier Transform (FFT) [73]. For this purpose the number of nodes must be chosen as $K = 2^M$, and an amount of computational labour is estimated by the magnitude $O(K \log_2 K)$. If the flow is periodic along one coordinate, one can use the splitting scheme on coordinates and combine the procedure FFT on the periodic coordinate with scalar sweep method on the other one. At last, though with less effectiveness, one can use for calculation of $\psi_{1,h}^{p+1}$ iterative methods in combination with the splitting schemes [68].

Approved computer programs for two-dimensional Poisson equation using the procedure FFT or its combination with the iterative method of the upper relaxation are given in Supplements B1, B2.

5. Using the stream mesh function ψ_h^{p+1} the values of the flow velocity \mathbf{U}_h^{p+1} in the Euler mesh nodes are calculated. For example, the components U_h^{p+1} can be determined from Eqs. (3) by central differences

$$u_{i,j}^{p+1} = -\frac{\psi_{i,j+1}^{p+1} - \psi_{i,j-1}^{p+1}}{2h_2}, \quad v_{i,j}^{p+1} = \frac{\psi_{i+1,j}^{p+1} - \psi_{i-1,j}^{p+1}}{2h_1}.$$

By the calculation of \mathbf{U}_h^{p+1} the Euler step is completed.

6. From the Euler mesh nodes the velocity is interpolated on the Lagrangian mesh of vortices with centers in the points $\{\mathbf{r}_i^{p+1}\}_{i=1}^N$.

The velocity of n -vortex is obtained as a sum of contributions from the four nearest nodes (see Fig. 25) by the formula

$$\begin{aligned} \mathbf{U}_n^{p+1} = \frac{1}{h_1 h_2} & (\mathbf{U}_{ij}^{p+1} A_{i,j} + \mathbf{U}_{i,j+1} A_{i,j+1} + \mathbf{U}_{i+1,j} A_{i,j-1} + \\ & + \mathbf{U}_{i+1,j+1} A_{i+1,j+1}). \end{aligned} \quad (21)$$

Here $A_{i,j}$ is an area of intersection of the n -vortex particle with the cell containing (i, j) -node. The computing procedure for this interpolation is presented in Supplement B1.

With use of the mesh vorticity $\omega_{2,h}^{p+1}$ by formulas Eqs. (14) new values of circulation Γ_i^{p+1} of Lagrangian particles are calculated. Calculations at the $(p+1)$ -time step are completed by this.

Most often the outlined method is used for calculation of flows when the fluid can be considered as inviscid, for example, in the limit of high Reynolds numbers. In this case, at the Euler step the computation of vorticity diffusion Eq. (20) is omitted. The general destiny of the step remains the smoothing of the flow velocity field through the reproduction of the cooperative character of vortex interaction.

4.4. The dynamics of vortices in three-dimensional flows

4.4.1. The vorticity evolution in the three-dimensional case

The VIC-methods for modeling of three-dimensional flows are less worked out and up to now they practically didn't used. The principal difficulty consists in the choice of the quantity which characterizes the Lagrangian particle. This quantity must be an additive function of fluid volume and it must determine its total vorticity.

In the two-dimensional case the vortex particle circulation serves by such a quantity. In fact, it has the property of additivity on the flow

plane. In particular, this leads to simple procedures Eqs. (19), (21) of interpolation of vorticity between the Euler and the Lagrangian meshes.

If to determine vorticity in the three-dimensional case by the formula [1]

$$\Omega = \text{rot} \mathbf{U} = \nabla \times \mathbf{U}, \quad (22)$$

where $\mathbf{U} = (u, v, w)$ is a flow velocity vector, then from the momentum equation for inviscid incompressible fluid [1]

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \frac{\partial \mathbf{U}}{\partial \mathbf{r}} = - \nabla p \quad (23)$$

the vorticity dynamics equation follows in the form

$$\frac{\partial \Omega}{\partial t} + \mathbf{U} \frac{\partial \Omega}{\partial \mathbf{r}} = \Omega \frac{\partial \mathbf{U}}{\partial \mathbf{r}}. \quad (24)$$

Comparing Eq. (24) with Eq. (1) describing the vorticity evolution in two-dimensional inviscid flows, one can see that in the right side of Eq. (24) the new term appears. It describes the stretching of vortex tubes [14] that is absent in the two-dimensional flows.

In the three-dimensional case circulation can serve as the characteristic only of individual vortex tubes. The tubes in addition to the stretching by the inhomogeneous flow also undergo the arbitrary bending deformations. As a result, very tangled configurations of these tubes appear in fluid. Because this the circulation is not already an additive function of the fluid volume.

A few studies are known where for the construction of the purely Lagrangian methods of discrete vortices in three-dimensional case the dynamics of vortex tubes was used, the elements of which interacted according to the Biot-Savart law [2]. For developing methods of the VIC-type such an approach is absolutely unfit because of its awkwardness. The only alternative is in the fact that it's necessary to find such a characteristics of the Lagrangian particles which is simply enough connected with the vorticity Ω and is additive over fluid volume.

4.4.2. The equation for the Lamb impulse density

In the paper [74] in the capacity of such a quantity the use of the Lamb impulse of the fluid particle was proposed. Traditionally it is determined in the following way [15]

$$\mathbf{P} = \frac{1}{2} \int_V dV [\mathbf{r} \times \Omega]. \quad (25)$$

In other terminology this quantity is called by hydrodynamic impulse. It has a physical sense of some part of total momentum. It is important that one can introduce the local volume density of the Lamb impulse $I(\mathbf{r}, t)$ using the presentation of the flow velocity field in the form

$$\mathbf{U}(\mathbf{r}, t) = \mathbf{I}(\mathbf{r}, t) + \mathbf{c}, \quad \mathbf{c} = \nabla\varphi. \quad (26)$$

If $\mathbf{I}(\mathbf{r}, t)$ has the sense of density, then

$$\mathbf{P} = \int_V d\mathbf{v} \mathbf{I}(\mathbf{r}, t)$$

and quantity \mathbf{P} really has the necessary property of additivity. Besides, from Eqs. (25), (26) it is seen that the Lamb impulse differs from zero only for those fluid particles which carry vorticity. Moreover from Eq. (26) it follows

$$\boldsymbol{\Omega} = \text{rot} \mathbf{U} = \text{rot} \mathbf{I}.$$

Below we denote $\mathbf{I}(\mathbf{r}, t)$ by a term "vortex impulse". For ideal fluid the equation for the $\mathbf{I}(\mathbf{r}, t)$ is derived by the substitution of expansion (26) into the momentum Eq. (23) [75]. The sought for equation looks as follows

$$\frac{\partial \mathbf{I}}{\partial t} + \mathbf{u} \frac{\partial \mathbf{I}}{\partial \mathbf{r}} = -(\mathbf{I} \nabla) \nabla \varphi \quad (27)$$

under the additional normalization condition

$$\Phi = p + \frac{\partial \varphi}{\partial t} + \frac{1}{2} (\nabla \varphi)^2 \equiv C(t) \quad (28)$$

or that the same

$$\nabla \Phi = 0$$

Taking into consideration the incompressibility condition

$$\nabla \mathbf{U} = 0 \quad (29)$$

by means of simple differentiation one can show that from the Eq. (27) the Eq. (24) for vorticity follows and from the norming condition pressure is expressed in the form

$$\Delta p = -\nabla(\mathbf{U} \nabla) \mathbf{U}.$$

This coincides with the equation for pressure calculating with using the known velocity field which is deduced directly from the Euler Eqs. (23), (29). Thus, the newly introduced representation of the velocity field (26) is correct and leads to the proper distributions of vorticity and pressure.

If the density distribution of the Lamb impulse is known, then the potential component of velocity taking into account Eq. (29) satisfies the equation

$$\Delta \varphi = -\nabla \mathbf{I}, \quad (30)$$

$$\mathbf{c} = -(4\pi)^{-1} \nabla \int_D d\mathbf{r}_1 g(\mathbf{r}, \mathbf{r}_1) \nabla \mathbf{I}, \quad (31)$$

where $g(\mathbf{r}, \mathbf{r}_1)$ is the Green's function of the Laplace operator in the flow domain D .

4.5. The vortex-in-cell scheme for three-dimensional flows

4.5.1. The splitting scheme

Using the general scheme of particle-in-cell method described in Chapter 1 the evolution equation for the vortex impulse $\mathbf{I}(\mathbf{r}, t)$ can be written in the form of the next splitting scheme at the time step τ

$$\frac{\partial \tilde{\mathbf{I}}_1}{\partial t} + \nabla(\mathbf{U}\tilde{\mathbf{I}}_1) = 0, \quad \tilde{\mathbf{I}}_1(p\tau) = \mathbf{I}(p\tau), \quad (32)$$

$$\frac{\partial \tilde{\mathbf{I}}_2}{\partial t} + (\tilde{\mathbf{I}}_{33}\nabla)\nabla\varphi = 0, \quad \tilde{\mathbf{I}}_2((p+1)\tau) = \tilde{\mathbf{I}}_1((p+1)\tau). \quad (33)$$

The chosen in this case Lagrangian and Euler steps can be considered as the transport of vortices in the self-induced (self-consistent) velocity field $\mathbf{U}(\mathbf{r}, t)$ and changing of vortex impulses by vortex line stretching. One can notice that the scheme of Eqs. (32), (33) is similar to Harlow particle method considered in Chapter 3. Here it's relevant to emphasize that one can also consider the other realization of the method which is close to the approach used traditionally in the rarefied plasma (see Chapter 5). Its distinction from the Harlow method is in the fact that the Lagrangian step at which the particles change their states is not only reduced to changing of their Cartesian coordinates (locations) in the computational domain. In plasma, for example, changing of particle momentums are also introduced into the evolution of the Lagrangian mesh.

In this case, we can introduce the mapping of the second auxiliary problem Eq. (33) on the set of particles. By such a way changing of the vortex impulses will be included into the Lagrangian step. Then the Euler step will be contained only calculation by finite-difference methods of the flow velocity field acting on vortex particles. The corresponding analog in the particle method for plasma is in the calculation on the Euler mesh of the electrostatic potential field and its space derivatives that are the forces acting on charged particles.

4.5.2. The Lagrangian vortex mesh

The transition to the discrete system (mesh) of the Lagrangian particles is realized by the representation of the vortex impulse field in the form

$$\mathbf{I}(\mathbf{r}, t) = \sum_{i=1}^N \mathbf{p}_i(t) R(\mathbf{r}, \mathbf{r}_i), \quad (34)$$

where kernel R is now determined on each argument in \mathbb{R}^3 . From the usual norming condition (see Chapter 1)

$$\int_{R^3} d\mathbf{r} R(\mathbf{r}, \mathbf{r}_1) = 1$$

in this case the the conservation law of full vortex impulse follows

$$\mathbf{P}(t) = \int_{R^3} d\mathbf{r} \mathbf{I}(\mathbf{r}, t) = \sum_{i=1}^N \mathbf{p}_i(t), \quad (35)$$

which is also fulfilled for the initial system of Eqs. (27).

Substitution Eq. (34) into Eq. (32) and integration with arbitrary smooth finite weight function $\varphi(\mathbf{r})$ gives us

$$\begin{aligned} & \sum_{i=1}^N \frac{d\mathbf{p}_i}{dt} \int d\mathbf{r} \varphi(\mathbf{r}) R(\mathbf{r}, \mathbf{r}_i) + \\ & \sum_{i=1}^N \mathbf{p}_i \int d\mathbf{r} R(\mathbf{r}, \mathbf{r}_i) \frac{\partial \varphi}{\partial \mathbf{r}} \left(\frac{d\mathbf{r}_i}{dt} - \mathbf{U}(\mathbf{r}_i) \right) = 0, \end{aligned}$$

where in the second term the relation

$$\frac{\partial R(\mathbf{r}, \mathbf{r}_i)}{\partial \mathbf{r}_i} = - \frac{\partial R(\mathbf{r}, \mathbf{r}_i)}{\partial \mathbf{r}}$$

was used. Because of the arbitrary choice of the function $\varphi(\mathbf{r})$ we obtain the following system of equations of vortex particle motion

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{U}(\mathbf{r}_i, t), \quad \mathbf{r}_i = \mathbf{r}_i^p \quad i = 1, \dots, N. \quad (36)$$

$$\frac{d\mathbf{p}_i}{dt} = 0 \quad \mathbf{p}_i = \mathbf{p}_i^p$$

If we want to include changing of particle impulses into the Lagrangian step it's necessary to substitute the representation Eq. (34) into the Eq. (33). Then for particle impulses one can carry out the dynamic system of the form

$$\frac{d\mathbf{p}_i}{dt} = -(\mathbf{p}_i \nabla) \nabla \varphi, \quad \mathbf{p}_i = \mathbf{p}_i^p \quad i = 1, \dots, N. \quad (37)$$

$$\frac{d\mathbf{r}_i}{dt} = 0 \quad \mathbf{p}_i = \mathbf{p}_i^p$$

In this case, the successive integration of systems Eqs. (36), (37) will be included into the Lagrangian step. And the Euler step will be restricted to the calculation on the fixed mesh of derivatives entering the right side of the system Eqs. (37).

Directly from Eqs. (37) it can be checked that in contrast to Eq. (35) the quantity $\sum_{i=1}^N \mathbf{p}_i^2$ is not the integral of this system. Thus, the systems Eqs. (36), (37) reflect correctly the main features of vorticity kinematics that are its transport and stretching by the self-induced inhomogeneous velocity field.

If we consider the point vortices with kernel $R(\mathbf{r}, \mathbf{r}_1) = \delta(\mathbf{r} - \mathbf{r}_1)$ and a dipole interaction law $\Phi_{ij} \sim |\mathbf{r}_i - \mathbf{r}_j|^{-3}$, then systems Eqs. (36), (37) can be written in a canonical form [74]. In this case, dynamic variables $\{\mathbf{r}_i\}_{i=1}^N$ and $\{\mathbf{p}_i\}_{i=1}^N$ appear to be conjugated with respect to the Hamiltonian [75] in the form

$$H = -\frac{1}{4\pi} \sum_{i < j}^N \left[\frac{\mathbf{p}_i \mathbf{p}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} - 3 \frac{(\mathbf{p}_i, \mathbf{r}_i)(\mathbf{p}_j, \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^5} \right]. \quad (38)$$

4.5.3. Computational cycle of the three-dimensional scheme

Because of the description of vortex particles in terms of vortex impulses the scheme of the computational cycle in many aspects is close to the scheme for the two-dimensional case presented in p.4.3. The block-scheme of the cycle for the case of dynamic system Eqs. (36) is shown in Fig. 26. It is supposed that in the computational flow domain $D \subset R^3$ the Euler mesh of the nodes is assigned which steps h for simplicity are taken uniform along all three coordinates.

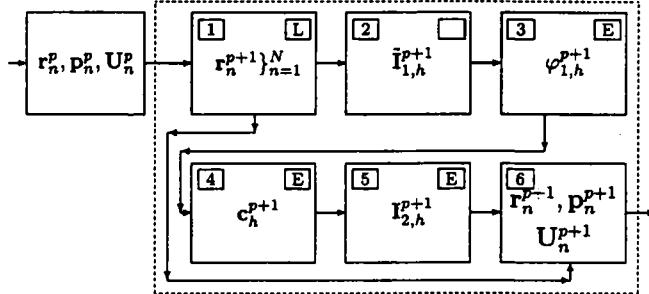


Fig. 26. The block-scheme of the vortex-in-cell method for three-dimensional flows.

The Lagrangian step.

1. As it is seen from the scheme in Fig. 26, under transition to the $(p + 1)$ -time moment the Lagrangian mesh of vortex particles is determined by their locations $\{r_n^p\}_{n=1}^N$ and by the vortex impulses $\{p_n^p\}_{n=1}^N$ replacing scalar circulations Γ_i in the plane case p. 4.3. In the points where the particles are located the values of flow velocity $\{U^p\}_{h=1}^N$ are known. For the calculation of the new position of vortices some finite-difference scheme for the system of ordinary differential Eqs. (36) is used. For example, the analogous Eq. (18), but supplemented by the equations for the third coordinate z .

2. The values of the auxiliary density mesh function of the vortex impulse $\{\tilde{I}_{1,h}^{p+1}\}$ is calculated in the nodes of the Euler mesh. If we restrict ourselves to the homogeneous distribution of impulse density in

the cubic particle with edge h equal to the mesh step and choose the kernel

$$R(\mathbf{r}, \mathbf{r}') = \begin{cases} 0 \\ h^{-3}, & |\mathbf{r} - \mathbf{r}'| < \frac{h}{2} \end{cases} \quad (39)$$

and a simple projection operator P on the set of nodes (see PIC-model in Chapter 1), then every i -vortex distributes its impulse $\tilde{\mathbf{p}}_{i,1}^{(p+1)}$ among eight Euler cells. If V_l is a volume of intersection of i -particle with the Euler cell and its center is in the node with multi-index l , then the impulse density introduced by this particle into this node is determined by the formula

$$\tilde{\mathbf{I}}_l^{p+1}(l) = \frac{V_l}{h^6} \mathbf{p}_{i,1}^{(p+1)}. \quad (40)$$

The value of mesh function $\tilde{\mathbf{I}}_{1,h}^{p+1}(l)$ in the fixed node l is determined by the sum of contributions of all particles having nonempty intersection with this cell. In this case for every coordinate of vector $\tilde{\mathbf{I}}_{1,h}^{p+1}(l)$ algebraic sums are calculated.

The further calculations can be referred to the Euler step.

The Euler step.

3. Accordingly to the mesh function $\{\tilde{\mathbf{I}}_{1,h}^{p+1}\}$ on the symmetric seven-point mold with the central node $l = (i, j, k)$ the difference divergence of impulse density $\zeta_{i,j,k}^{(p+1)}$ is calculated

$$\zeta_l^{p+1} = \text{div}_h \tilde{\mathbf{I}}_{1,h}^{(p+1)} = \frac{1}{2h} (\Delta J_i + \Delta Q_j + \Delta S_k), \quad (41)$$

where l is a multi-index of the space mesh node; $\{J\}_l, \{Q\}_l, \{S\}_l$ are the sets of x -, y - and z -components of auxiliary vector function $\tilde{\mathbf{I}}_{1,h}^{p+1}$,

$$\Delta J_i = J_{i+1,j,k} - J_{i-1,j,k} \quad (42)$$

and analogously the other differences are determined. As a result, we derive the set of the right side values for discrete approximation of the Poisson equation (30). Determined in such a way the boundary-value problem can be solved by one of the methods mentioned in p. 2 for the two-dimensional case.

4. Using the obtained solution of the mesh Poisson equation $\{\varphi_{1,h}^{p+1}\}$ the potential component $\{c_h^{p+1}\}$ of flow velocity field is calculated. For this one can use again central differences determined on the symmetrical mold. If we denote through $\{u_{i,j,k}\}, \{v_{i,j,k}\}, \{w_{i,j,k}\}$ the sets of components of the mesh vector function $\{c_h^{p+1}\}$, then

$$u_{i,j,k} = \frac{\varphi_{i+1,j,k}^{p+1} - \varphi_{i-1,j,k}^{p+1}}{2h}$$

and analogously other two projections are calculated.

5. By sets of the potential component of the velocity field $\{c_h^{p+1}\}$, using the calculated in block 2 auxiliary mesh function $\{\tilde{\mathbf{I}}_{1,h}^{p+1}\}$ and some

difference scheme for problem (33), one can calculate the values of vortex impulse density $\{\tilde{I}_{2,h}^{p+1}\}$ at the following $(p+1)$ -time moment. For example, the possible variant in the vector form looks like as follows

$$\begin{aligned} & \frac{\tilde{I}_{2,l}^{p+1} - \tilde{I}_{1,l}^{p+1}}{\tau} + J_l \frac{c_{i+1,j,k}^{p+1} - c_i^{p+1}}{h} + \\ & + Q_l \frac{c_{i,j+1,k}^{p+1} - c_i^{p+1}}{h} + S_l \frac{c_{i,j,k+1}^{p+1} - c_i^{p+1}}{h} = 0. \end{aligned} \quad (43)$$

At last, in accordance with the formula (26) the set of values of velocity mesh function is calculated

$$U_l^{p+1} = I_l^{p+1} + c_l^{p+1}. \quad (44)$$

6. For the completion of the cycle the velocity field $\{U_l^{p+1}\}$ is interpolated onto the Lagrangian mesh of vortex particles located in points $\{r_n^{p+1}\}_{n=1}^N$ which are calculated in the block 1.

If the kernel of the form (39) is used, then the velocity of the n -vortex located in the cell with center-node l is derived as the sum of contributions from the eight nearest to the point r_n^{p+1} nodes.

The new values of the vortex impulses of particles are recalculated in the same way with using the mesh function of impulse density $\{\tilde{I}_l^{p+1}\}$. The general formula can be written down in the form

$$Z_n^{p+1} = h^{-3} \sum_{(l)} Z_l^{p+1} V_l, \quad (45)$$

where V_l is a volume of intersection of the cubic vortex particle having number n with the nearest Euler cells, Z is a recalculated characteristic. As a result, we obtain the reconstructed mesh of particles with coordinates and vortex impulses $\{r_n^{p+1}, p_n^{p+1}\}_{n=1}^N$ and a set of their velocity values $\{U_n^{p+1}\}_{n=1}^N$. The cycle is completed by this.

Under including into the calculation dynamic system Eqs. (47) the succession of calculations is changed. In order don't deviate too much from the already described scheme presented in Fig. 26 one can modify it in the following way. After calculation of the potential component of velocity $\{c_h^{p+1}\}$ realized in block 4, the sets of its finite-difference derivatives are calculated in addition using the central differences. For example, the set of derivatives on x is approximated by relations:

$$\{c_{x,l}^{p+1}\} = \left\{ \left\{ \frac{\Delta u_i}{2h} \right\}, \left\{ \frac{\Delta v_i}{2h} \right\}, \left\{ \frac{\Delta w_i}{2h} \right\} \right\}, \quad (46)$$

where the differences of the components are defined by formulas similar to Eq. (42). The calculated in such a way sets of mesh vector-functions $\{c_{x,l}^{p+1}\}$, $\{c_{y,l}^{p+1}\}$, $\{c_{z,l}^{p+1}\}$ are interpolated into centers of particles $\{r_n^{p+1}\}_{n=1}^N$ by formulas (45). The right sides in the dynamic system Eqs. (37) are determined by this. Further the values of vortex impulses at $(p+1)$ -time moment are calculated from it. In particular, for x -component of impulses the difference scheme is written in the following form:

$$\frac{p_{n,x}^{p+1} - p_{n,x}^p}{\tau} = -\frac{1}{2h}(p_{n,x}^p \Delta u_{i,n} + p_{n,y}^p \Delta u_{j,n} + p_{n,z}^p \Delta u_{k,n}). \quad (47)$$

Here $p_{n,x}, p_{n,y}, p_{n,z}$ are the projections of vortex impulse of n -particle onto the corresponding axes; the indices show on what coordinate of multi-index l the increments by Eqs. (42) are taken. Index n in differences means that they are interpolated into point r_n^{p+1} by Eq. (45).

In the analogous difference equations for components $p_{n,y}^{p+1}, p_{n,z}^{p+1}$ the differences $\Delta v, \Delta w$ appear.

The values of the particle impulses calculated by formulas (40) are recalculated in the mesh density field of the vortex impulse $\{I_{2,h}^{p+1}\}$. After this on the basis of formulas (44) the set $\{U_h^p\}$ of flow velocity mesh function is calculated and it is then projected onto the particles in accordance with formulas (45). The cycle is completed by this.

It's easy to see that under such a scheme the calculations referring to the Euler and Lagrangian steps are alternated during the cycle. As the logic of calculations is clear, then it is easy to suggest a scheme in which these steps will be explicitly delineated.

4.6. The examples of applications

Methods of discrete vortices got the widest spreading under modeling of ideal fluid flows with little domains of the concentrated vorticity and also of slightly viscous and turbulized flows (the case of high Reynolds numbers Re).

In the calculations of such flows on the basis of finite-difference methods it's often necessary to use the significant refinement of meshes in order to achieve, for example, at least the minimum resolution of the boundary layer or the shock transition. So, the order of the width of the boundary layer is estimated [14] by the magnitude $\delta \sim O(Re^{-\frac{1}{2}})$. If we require $h \sim \delta$, then it leads to the choice of the mesh spacing $h \sim O(Re^{-\frac{1}{2}})$.

The more strict requirement is connected with the necessity to reduce the scheme viscosity which appears at the difference approximation of convective derivatives in the left side of the Eqs. (32). As it is known [68], the scheme viscosity is estimated as $\mu_h \sim O(h)$. It's obvious that if μ_h exceeds physical (or turbulent) viscosity, then the computational picture of the current can be strongly distorted. Hence the restriction appears

$$\mu_h \sim h \sim Re^{-1}$$

which is more burdensome because it must be fulfilled in the whole computational domain.

Even from these estimations which in practice must make strictly it is seen that for flows with high Re numbers the requirements for the mesh

can exceed the possibilities of mean power computers. The alternative is the transition to mesh-free algorithms to which the Lagrangian methods of discrete vortices belong. The imitative approach towards modeling of vortex flows leads in a very natural way to the purely Lagrangian algorithms. Because of this the Lagrangian methods of vortices are mostly wide spread.

The interest towards the VIC-methods appeared under consideration of problems which demand the reproduction of thin details of the vorticity dynamics. This is explained by the fact that they allow to use significantly greater quantity of model particles with the simultaneous reducing of calculation errors. In fact, the main amount of computational labour in the Lagrangian methods at the time step τ is connected with the determination of the flow velocity in the location points of N vortices. In this case, for any discrete approximation of the Biot-Savart integral (5) the amount of calculations is estimated by the value $O(N^2)$. In the computational practice this restricts the quantity of model vortices by number $N \sim k \cdot 10^3$ [2].

For VIC-methods the procedure of vorticity interpolation from the Lagrangian mesh of vortices into the nodes of the Euler mesh and the inverse interpolation of the velocity field (see p. 4) require the calculations of the order $O(N)$. If we use the economical direct method for solving the Poisson difference equation (30) (Poisson solver), for example, the procedure FFT, then the amount of calculations at the mesh with M -nodes is $O(M \log_2 M)$. As a rule, $M \approx N$. That's why VIC-methods under corresponding program realization are nearly optimal on the amount of computational labour. This gives the possibility to increase significantly (for two-three orders) the number of the Lagrangian points in comparison with the purely Lagrangian vortex methods.

The first works on the practical use of VIC-method belong to G.P. Christiansen who created the universal code VORTEX for calculation of two-dimensional vortex flows. In the paper [19] the possibilities of the method and code VORTEX are demonstrated on several classical problems of vorticity dynamics in ideal fluid. In particular, the destruction of the direct and circular vortex layers of the finite thickness was calculated under development of Kelvin-Helmholtz instability. The formation of Carman vortex street was also modeled. Besides, the problems of interaction of two Rankin vortices and of penetration of the plane jet into the half-space of ideal fluid were solved. In every example the evolution of vorticity was traced during several characteristic times determined by the vorticity transport on the space scale of the problem. It should be noted that finite-difference methods in all these cases allow to reproduce satisfactorily only the small initial interval of the evolution. Further because of the fast non-physical vorticity diffusion caused by the scheme viscosity the picture of the process is absolutely distorted.

In calculations [19] the regular mesh with steps $h_x = h_y = 1$ and the number of nodes $M = 64 \times 64$ was used. The quantity of vortices

with the rectangular kernel (17) was taken $N = 3200$. For the Poisson equation (2) the economical algorithm was used with the volume of calculations of the order $O(M \log_2 M)$. The three quarters of the processor time on one time step τ were spent for solving of the Poisson equation.

Practically, at the whole computed time interval up to 1000 time steps well-defined vorticity distributions were obtained consistent with physical representations and estimations. The example of visualization for the problem of the Carman street formation is presented in Fig. 27. In this case, the street is formed under destruction of two vortex bands with vorticity of the opposite sign. The points represent centers of vortex particles. One can clearly see the appearance of waves of the finite length ($t = 200$) from the small scale perturbations ($t = 120$). Then these waves are transformed into the vortex street ($t = 280$). The last cadre is rather significant in the sense of many details interesting for specialists in hydrodynamics. One can note the chess configuration of the street, the oval form of vortices with the clear boundary, the characteristic inclination of their main axes, the magnitude of intercentre distances, specific thin cross connections ("braides") between the vortices of the same sign, etc.

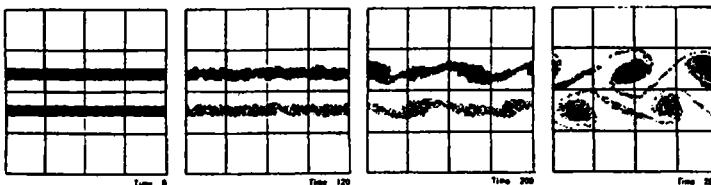


Fig. 27. The formation of the Carman vortex street.

The such a quality of computational information are accessible in this case only for the vortex-in-cell method.

Let us consider one more example when by VIC-method the applied problem of flow past slender delta wing was solved. In the paper [18] the space evolution of the vortex sheet shed from the back edge of the finite wing was modeled. The formation of the sheet is explained by pressure difference on the stream lines enveloping the airfoil from above and from below. At the ends of the wing the sheet rolls-up into conic spiral vortices.

The transition from the stationary problem in space $R^3(x, y, z)$ to the two-dimensional non-stationary one in $R_+(t) \times R^2(x, y)$ is justified by the transformation $Z = Ut$. It's clear that two-dimensional distributions of vorticity taken at the fixed time moments must be considered as cross-sections of vortex configuration behind the wing by the planes $Z=\text{const}$.

In calculations up to $N = 2000$ vortices and uniform Euler mesh $M = 129 \times 129$ were used. The vortex particles had the same circulation and the rectangular PIC-kernel Eq. (17). The Poisson equation for

stream function was solved by the same fast algorithm which was used in the paper [19]. Above a half of processor time per time step τ was spent for solving of Poisson equation. The example of the vortex distribution is shown in Fig. 28. In these calculations the vortex sheet characteristics of practical interest were obtained.

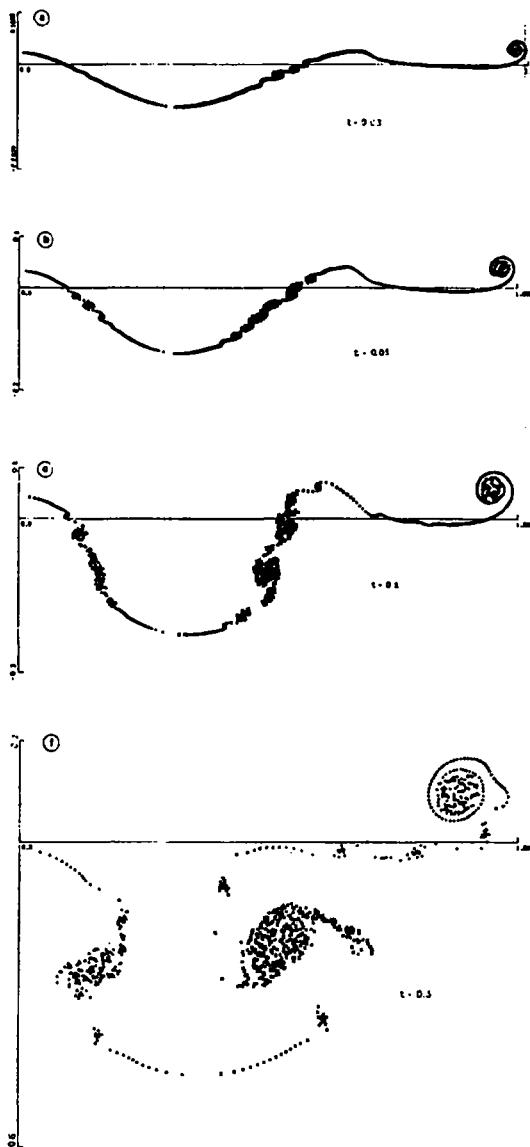


Fig. 28. The roll-up of the vortex sheet in trailing vortex under flow past finite wing.

The investigations carried out in the papers [2, 18, 19] also contain the useful information about the nature of computational errors and practical accuracy of two-dimensional VIC-methods.

Three-dimensional VIC-methods nowadays only start their development.

In the conclusion it can be noted that vortex-in-cell methods considered in this chapter don't play such a significant role in the computational aerohydrodynamics as it can be said about the particle methods in plasma physics (see Chapter 5) or in the rarefied gas dynamics (see Chapter 6). But we want to emphasize that these methods have their own field of application connected with precision modeling of vortex flows where their priority is obvious. The growth of the interest of specialists towards such problems taking place in the last years and the spread introduction of powerful personal computers will lead undoubtedly to their significant spreading in the nearest future.

Chapter 5

Particle-in-cell methods in collisionless plasma dynamics

5.1. Introduction

Particle-in-cell methods became very widely used in solving plasma physics problems. Plasma is a medium containing free particles with positive and negative charges. Positive particles are always ions and negative ones are usually electrons. It can be several species of ions that differ in mass and charge. Since plasma is basically neutral, it is, at least, a two-component medium consisting of electrons and ions.

In many cases plasma can be considered as collisionless. This means that the characteristic spatial scales of such a plasma are significantly less than the mean free path between binary collisions. In this case, the processes of *collective interaction* of electromagnetic field and particles [7, 38] are determining effects. In plasma physics this term designates the totality of different nonlinear interactions of plasma vibrations with charge carriers.

The particle method is the most universal computational tool widely used for numerical solving of collisionless plasma physics problems. The first one-dimensional version of particle method is the "flat sheets" model suggested by O.Buneman in 1959 [21]. He investigated a one-dimensional electronic beam consisting of particles which had the form of discs with a diameter equal to the beam diameter. It was a purely Lagrangian algorithm, where all particles interacted in pairs.

The particle method in plasma is clearly imitative in character. Model particles obey the same laws as real ions and electrons, but the number of model particles per unit volume is much less than the real plasma density. This fact due to the limited computer capacities. Therefore it is important to examine the change of physical properties caused by the transfer from real plasma to model "particle" medium.

Let a value $\alpha = n/n_M$ is a ratio of real and model plasma densities. Hereinafter index M designates the model plasma characteristics. Then from the conservation conditions of total charge and mass it follows that

$$e_M = \alpha e, \quad m_M = \alpha m,$$

where m, e and m_M, e_M are the masses and charges of individual particles in real plasma and model one respectively. In this case, the particle

dynamics does not change as far as $e_M/m_M = e/m$. The electron and ion plasma frequencies are also conserved [22]:

$$\omega_{e,M} = (4\pi e_M^2 n_M / m_{e,M})^{1/2} = \omega_e, \quad \omega_{i,M} = \omega_e \left(\frac{m_e}{m_i} \right)^{1/2} = \omega_i.$$

If we assume that Debye length is invariant [38]

$$\lambda_{D,M} = (T_{e,M}/4\pi e_M^2 n_M)^{1/2} = \lambda_D,$$

then the model and real plasma temperatures will be different: $T_{e,M} = \alpha T_e$ though the thermal velocities remain the same

$$v_{T,M} = (T_{e,M}/m_M)^{1/2} = v_T.$$

Therefore the level of shot noises is much higher in the model plasma. This is a drawback when a cold plasma is simulated.

When modeling a quasi-neutral plasma, Debye length λ_D plays an important role. Within a sphere of radius λ_D the binary interaction effects between individual particles are strongly manifest themselves and the cooperative effects of particle interaction predominate beyond the sphere. Plasma is of interest in spatial scales much more than λ_D . That is why it is necessary that Debye's length in the model plasma should be sufficiently small.

As it usually is, $\alpha \gg 1$ the mean free path in the model plasma is much smaller, and the collision frequency is much higher than those in the real plasma,

$$\lambda_M = \lambda/\alpha,$$

$$\nu_M = v_T/\lambda_M = \nu\alpha.$$

Simultaneously the number of model particles in Debye sphere decreases. All this brings the model plasma closer a state when the particle collisions should be taken into account. However, if the mean free path of model particles remains much greater than the spatial scales of the problem, collisionless simulation is possible [7].

In other problems distortions of another properties are essential. Of course, all these drawbacks can be overcome by increasing the number of model particles. However, in this case the possibilities of the ordinary researcher are usually limited. Besides, various non-physical effects can be significantly reduced by special modifications of the method. But, as a rule, the price of such an improvement is a further complication of algorithm and an increase in the computation time.

The chapter is organized as follows. In p. 5.2 the plasma dynamics equations in the approximation of a self-consistent field are briefly described. The equations of motion of charged particles, which are universal for both the real plasma and the model one are also presented. Point 5.3 contains a general scheme of the particle-in-cell method for

the plasma problems. Some numerical schemes for the particle dynamics equations with various fields in the medium are also presented in detail. The computational errors due to the violation of the energy and momentum conservation laws are described in p. 5.4. In p. 5.5 some examples of the simulation of collisionless plasma problems by the particle method are considered.

5.2. Collisionless plasma basic equations

5.2.1. Kinetic equation with a self-consistent field

Collisionless fully ionized plasma is described by the system of Vlasov kinetic equations for distribution functions of different species of particles [7] $f_\alpha(t, \mathbf{r}, \mathbf{v})$:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \frac{\partial f_\alpha}{\partial \mathbf{r}} + \frac{\mathbf{F}_\alpha}{m_\alpha} \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}} = 0, \quad (1)$$

$$\mathbf{F}_\alpha = -\mathbf{a} + q_\alpha (\mathbf{E} + \frac{1}{c} \mathbf{v} \times \mathbf{B}).$$

Here $\alpha = 1, \dots, s$ denotes particle species that can be electrons or ions of various types, q_α, m_α are charge and mass of α -species, \mathbf{E} is an electric field, \mathbf{B} is a magnetic induction vector, c is a light speed, \mathbf{a} is any external force of non-electromagnetic nature.

Kinetic equations are supplemented by Maxwell equations for electromagnetic fields

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial t} &= \text{crot} \mathbf{B} - 4\pi \mathbf{j}, \\ \frac{\partial \mathbf{B}}{\partial t} &= -\text{crot} \mathbf{E}, \\ \text{div} \mathbf{E} &= 4\pi \rho, \\ \text{div} \mathbf{B} &= 0. \end{aligned} \quad (2)$$

In these equations current density \mathbf{j} and charge density ρ are defined by particle distribution functions as follows

$$\rho(t, \mathbf{r}) = \sum_\alpha q_\alpha \int f_\alpha(t, \mathbf{r}, \mathbf{v}) d\mathbf{v}, \quad (3)$$

$$\mathbf{j}(t, \mathbf{r}) = \sum_\alpha q_\alpha \int f_\alpha(t, \mathbf{r}, \mathbf{v}) \mathbf{v} d\mathbf{v}, \quad (4)$$

Let us omit below summing by index α and consider only one particle species. It is possible in collisionless approximation as particles of various types do not interact with one another directly.

5.2.2. Particle motion equations

The system of dynamic equations of model particles is derived by substitution in Eq. (1) of distribution function in the form

$$f(\mathbf{r}, \mathbf{v}, t) = \sum_{j=1}^M R(\mathbf{r} - \mathbf{r}_j(t))\delta(\mathbf{v} - \mathbf{v}_j(t)), \quad (5)$$

where M is the total number of model particles.

By multiplying (1) on smooth finite function $\psi(\mathbf{r}, \mathbf{v})$ and integrating over $R^3(\mathbf{r}) \times R^3(\mathbf{v})$ we get the following expression

$$\begin{aligned} & \sum_{j=1}^M \left[\frac{d\mathbf{r}_j(t)}{dt} - \mathbf{v}_j(t) \right] + \sum_{j=1}^M \left[\frac{d\mathbf{v}_j(t)}{dt} - \right. \\ & \left. \frac{1}{m} \int \mathbf{F}(t, \mathbf{r}, \mathbf{v}_j(t))R(\mathbf{r} - \mathbf{r}_j(t))d\mathbf{r} \right] = 0. \end{aligned} \quad (6)$$

The latter will be satisfied identically under the conditions

$$\frac{d\mathbf{r}_j(t)}{dt} = \mathbf{v}_j(t), \quad (7)$$

$$\begin{aligned} m \frac{d\mathbf{v}_j(t)}{dt} &= \int \mathbf{F}(t, \mathbf{r}, \mathbf{v}_j(t))R(\mathbf{r} - \mathbf{r}_j(t))d\mathbf{r} \equiv \mathbf{F}_j(t) \\ j &= 1, 2, \dots, M. \end{aligned} \quad (8)$$

Conditions (7), (8) are the equations of particle motion.

The integral term in Eq. (8) is a force acting on model particle with index j , described by kernel $R(\mathbf{r}, \mathbf{r}_j(t))$. The electric and magnetic fields in the expression for Lorentz force are determined on discrete set of nodes of Eulerian mesh. However, for calculating $\mathbf{F}_j(t)$ one must know the distribution of fields in the whole domain. Therefore it is necessary to interpolate the force field. Depending on the interpolation various quadrature formulas are obtained for approximation (8).

Let us present a force $\mathbf{F}(\mathbf{r}, t)$ through its values $\mathbf{F}(\mathbf{r}_\alpha, t)$ at the mesh nodes by interpolation formula

$$\mathbf{F}(\mathbf{r}, t) = \sum_{\alpha} \mathbf{F}_{\alpha}(t) S(\mathbf{r} - \mathbf{r}_{\alpha}), \quad (9)$$

where S is the interpolating function satisfied the normalization condition

$$\sum_{\alpha} S(\mathbf{r} - \mathbf{r}_{\alpha}) = 1$$

for all values of \mathbf{r} . Function S determines the character of force interpolation.

Let us consider a number of examples in one-dimensional case [22]. Let $\{x_{\alpha}\}$ be cell centers of uniform mesh with step h . For NGP-model we have

$$1. \quad S(x - x_\alpha) = \begin{cases} 1, & |x - x_\alpha| \leq h/2, \\ 0, & |x - x_\alpha| > h/2, \end{cases}$$

$$R(x - x_j) = \delta(x - x_j).$$

Then

$$F_j = \int \sum_{\alpha} F_{\alpha} S(x - x_{\alpha}) \delta(x - x_j) dx =$$

$$\sum_{\alpha} F_{\alpha} \int S(x - x_{\alpha}) \delta(x - x_j) dx = \sum_{\alpha} F_{\alpha} S(x_j - x_{\alpha}) = F_{\alpha'}, \quad (10)$$

where α' is the index of mesh node nearest to the particle with coordinate x_j , that is $x_{\alpha'} \in [x_j - h/2, x_j + h/2]$.

For PIC-model the corresponding functions take the form

$$2. \quad S(x - x_{\alpha}) = \begin{cases} 1, & |x - x_{\alpha}| \leq h/2, \\ 0, & |x - x_{\alpha}| > h/2, \end{cases}$$

$$R(x - x_j) = \begin{cases} h^{-1}, & |x - x_j| \leq h/2, \\ 0, & |x - x_j| > h/2. \end{cases}$$

Here

$$F_j = h^{-1} \sum_{\alpha} F_{\alpha} \int_{x_j-h/2}^{x_j+h/2} S(x - x_{\alpha}) dx.$$

If $x_{\alpha-1} < x_j < x_{\alpha}$, then

$$F_j = h^{-1}[(x_{\alpha} - x_j)F_{\alpha-1} + (x_j - x_{\alpha-1})F_{\alpha}]. \quad (11)$$

As seen from this formula, the force acting on the particle is determined by inverse linear interpolation between field values in the two nodes of the space mesh nearest to the particle.

For TSC-model one can write

$$3. \quad S(x - x_{\alpha}) = \begin{cases} 1, & |x - x_{\alpha}| \leq h/2, \\ 0, & |x - x_{\alpha}| > h/2; \end{cases}$$

$$R(x) = \begin{cases} h^{-1}(1 - |x|/h), & |x| \leq h, \\ 0, & |x| > h \end{cases}$$

In this case,

$$F_j = \sum_{\alpha} F_{\alpha} \int S(x - x_{\alpha}) R(x - x_j) dx = \sum_{\alpha} F_{\alpha} \int_{x_{\alpha}-\frac{h}{2}}^{x_{\alpha}+\frac{h}{2}} R(x - x_j) dx.$$

At $x_{\alpha} - \frac{h}{2} \leq x_j \leq x_{\alpha} + \frac{h}{2}$ we get the following expression for force:

$$F_j = \frac{1}{2h^2} \left\{ \left(x_j - x_{\alpha} - \frac{h}{2} \right)^2 F_{\alpha-1} + \left[\frac{3}{2}h^2 - 2(x_j - x_{\alpha})^2 \right] F_{\alpha} + \right.$$

$$+ \left(x_j - x_\alpha + \frac{h}{2} \right)^2 F_{\alpha+1} \Biggr\}. \quad (12)$$

It is easy to obtain special cases by formula (12). Let $x_j = x_\alpha - \frac{h}{2}$. In this case, the particle is located exactly in the middle between nodes α and $\alpha - 1$. Then the force acting on the particle is $F_j = (F_\alpha + F_{\alpha-1})/2$. If $x_j = x_\alpha$ and the particle is in the node, then the force acting on it is $(F_{\alpha-1} + 6F_\alpha + F_{\alpha+1})/8$, which is not equal to force F_α due to some "smearing" of the particle charge on the mesh nodes. It follows from formula (12) that the force acting on the particle is calculated by quadratic interpolation.

5.3. General scheme and computation cycle of the method

Let us assume that regular Eulerian mesh is introduced in computational domain of coordinate space. The mesh nodes, that are the Cartesian cell centers, compose a mesh domain G_h .

5.3.1. Initial data

The choice of plasma particle initial distribution on coordinates and velocities in discrete models is extremely important because the choice affects the possibility of adequate investigation of considered physical processes.

Let us first consider space distribution of particles. If the particles to put in space completely chaotically, then as the analysis shows the major part of electric field energy is concentrated in long wavelength harmonics. There at an electrostatic energy density $\epsilon \propto k^{-2}$, and indefinitely increase with the decreasing of wave number k . However, in reality, as it follows from plasma theory, spectral density of electric field energy is to be in proportion to $(2 + k^2 D^2)^{-1}$, where D is Debye length. Hence, chaotic particle space distribution is not satisfactory. On the other hand, an uniform distribution of model particles of both charges in computational domain of coordinate space $R^3(\mathbf{r})$ can be also unsatisfactory [38]. In real plasma particles of different charges are distributed in space in such a way that Debye screening takes place over Debye radius D . If mesh effects are insignificant it is possible to set particle spatial distribution in such a manner to simulate Debye screening [76]. In this case, ions are distributed in computation domain in G_h first turn. Then for each of them the position of electron "partner" is sampled by Monte Carlo method. For two-dimensional case random distance is simulated with probability density

$$p(r) = (\pi D^2)^{-1} K_0(\sqrt{2r}/D), \quad (13)$$

where $K_0(x)$ is Macdonald function [34]. In paper [76] an effective way of approximate sampling of random values with probability density (13) is given. Let ξ be a random value uniformly distributed over the interval $[0,1]$. Then the required distance between electron and ion can be calculated by formula

$$r = 2^{-1/2} D \left(1.0721a + 0.6601a^2 + 0.0697a^3 \right) / \\ (1 + 0.0867a - 0.1840\ln(a)),$$

where $a = -\ln(\xi)$. Another effective method of sampling random value r with distribution (13) is based on the Neumann rejection method (see Chapter 6, Procedure 3.1).

Particle velocity distribution is simulated by Maxwell distribution with initial temperature T . For this purpose one can use a set of M random numbers $\xi_1, \xi_2, \dots, \xi_M$ uniformly distributed in the interval $[0,1]$. In this case, the value defined by formula

$$v_M = v_T \left(\sum_{i=1}^M \xi_i - \frac{M}{2} \right) \left(\frac{M}{12} \right)^{-1/2}$$

is normally distributed. If plasma moves as a whole, then it is taken into account by introducing the corresponding macroscopic velocity \mathbf{U} in simulating function [30]. Simple program for preparing such initial distributions is presented in Supplement A.

Debye distribution gives an initial level of heat fluctuations in accordance with the given value N_D which is a number of particles in Debye sphere in real plasma. It can differ significantly from corresponding N_D in a model plasma. To investigate phenomena with energies lower than heat fluctuations in model plasma, one can recommend uniform distribution on coordinates and for velocities it is possible to use low noise level method called quiescent start [77].

The method consist in the following. Computation domain is divided into a number of subdomains with the same number of particles N from total number J of particles. In the case of spatially uniform plasma density distribution the cells of spatial mesh can be chosen as such subdomains. Instead of presenting Maxwell function by J different velocity values only N different velocities are used for reproducing Maxwell function in each subdomain, and in all subdomains the same N values of particle velocities are sampled. Thus, each velocity value is assigned to J/N particles, that is there are N velocity groups (beams). All beams are spatially homogeneous and fluctuations are decreased up to round-off errors.

Quiescent start method makes it possible to investigate a development of plasma instabilities and nonlinear wave interaction with energies lower than thermal fluctuations caused by a limited number of N_D in a computational model. However, as model plasma is presented as a set of beams, quiescent start predetermines beam instability.

Increment of these instabilities at small $\Delta v/v_T$, where Δv is beam velocity difference, approximately equals to $\gamma \approx (2\pi)^{-1}k\Delta v|\ln(\Delta v/v_T)|$ [78]. Thus, modeling can be carried out with a small fluctuations up to time $t < \gamma_{max}^{-1} \approx 2\pi(k_{max}\Delta v_{max})^{-1} \times (5 \div 10)$.

As $\Delta v_{max} \sim v_T/K$, where K is the number of beams for wave length $k_{max}D \approx 1$, we get $\tau \approx (5 \div 10)\frac{2\pi}{\omega_{0e}}K$.

In order words, at the number of beams $K \geq 10$ calculations up to the time values $t \approx 10^2\frac{2\pi}{\omega_{0e}}$ are possible which is sufficient for many practical problems. The time of beam instability development can be increased by decreasing the value Δv , but in this case it is necessary to increase the number of particles.

When using the described algorithm of setting the initial values the magnitudes Δv of intervals increase with advancing towards high velocity values. Increase of Δv leads to the growth of beam instability increments. It is possible to decrease increment by setting the same intervals Δv with the help of particle "weighting" [78]. In this case, all particles as before are divided into several beams, but with the same velocity difference Δv . After that mass and charge of each particle of beam with velocity v is multiplied by $\exp(-v^2/v_T^2)$. In such a way particle dynamics does not change as it is determined by ratio of particle charge to mass, but time of development undesirable beam instability increases.

When external electric and magnetic fields exist in plasma, their values at the initial moment should be assigned in mesh nodes G_h . Mesh functions of self-consistent fields are obtained by interpolation of charges and currents from particle Lagrangian mesh. An example of subroutine for calculating mesh density is given in Supplement B2.

5.3.2. Computation cycle

In collisionless plasma problems particle-in-cell method at the recurrent time step τ is also implemented in two stages which correspond to the splitting on physical processes. At the Lagrangian step collisionless particle motion is computed which is described by dynamic system Eqs. (7), (8). At the Eulerian step on the basis of the obtained particle distribution self-consistent electromagnetic fields are calculated by Maxwell equations (3). The general scheme of finite-volume method for Maxwell equations is given in Chapter 2. On regular meshes the Eulerian step is carried out by finite-difference schemes. The block diagram of particle method for collisionless plasma problems is designed on the Fig. 29.

Lagrangian step.

At this stage Lagrangian particle mesh evolution is calculated in accord with equations (7), (8). Some approaches to their integration complying with specific of plasma problems are presented in the next

point. The distribution functions (5) for particles of finite size are usually used. It allows to decrease fluctuations of plasma parameters which are connected with the transition from cell to cell of particles with finite charge and mass. Besides, finite size of particles increases their effective range that corresponds to the collective character of interaction in the collisionless plasma.

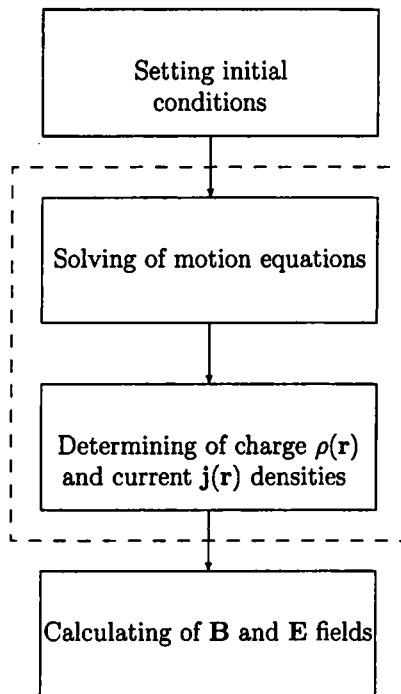


Fig. 29. Block diagram of particle method for collisionless plasma problems.

After calculating the arrays of new coordinates $\{\mathbf{r}_j\}_{j=1}^M$ and new velocities $\{\mathbf{v}_j\}_{j=1}^M$ of particles the charge density and average particle velocities in mesh nodes are calculated by formulas

$$\rho_k(t) = \sum_{j=1}^M q_j \bar{R}(\mathbf{r}_k - \mathbf{r}_j(t)), \quad (14)$$

$$\mathbf{v}_k(t) = \frac{1}{\rho_k(t)} \sum_{j=1}^M q_j \mathbf{v}_j(t) \bar{R}(\mathbf{r}_k - \mathbf{r}_j(t)), \quad (15)$$

These formulas are obtained by substitution of distribution function in equations (3) and (4).

Eulerian step.

At this step Maxwell equations are solved by finite-difference methods and electric and magnetic mesh fields are computed. Here transition to Eulerian mesh allows to avoid singular interaction of particles at small range and to decrease amount of computations in comparison with pure Lagrangian particle methods. We do not describe different types of Maxwell equations under various physical assumptions and corresponding schemes of solving them. Here we restrict ourselves to a few examples of finite-difference methods used for computations on Eulerian step in the last two points of the chapter. Besides the program for integrating of the Maxwell equations for the problem of advancing of electric field impulse in a square domain is given in Supplement E.

5.3.3. Numerical schemes for dynamic equations of model particles

The particle motion equations (7), (8) are ordinary differential equations for solving of which there are many numerical methods. However, numerical schemes to be used in applications of particle method to plasma problems are to satisfy certain requirements. First of all numerical scheme is to reproduce physical properties of particle motion such as reversibility in time, energy conservation, etc. with sufficient accuracy. The scheme is to be simple for computation, to save computation time and to be one-step one to save computer memory.

Let us consider some of schemes [7, 22] for solving the equations of particle motion.

1°. *Particle motion in electric field.* In this case, $\mathbf{B} = 0$ and electric field can be determined through electric potential: $\mathbf{E} = -\nabla\varphi$. The most widely used scheme is a "leapfrog" scheme where velocities and coordinates of particles are calculated on shifted with respect to each other time meshes. It is written in the following form

$$\begin{aligned}\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\tau} &= \frac{q}{m} \mathbf{E}(\mathbf{r}^n), \\ \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} &= \mathbf{v}^{n+1/2}.\end{aligned}\tag{16}$$

Here τ is time step, a particle coordinates \mathbf{r}^n is calculated at time $t^n = n\tau$, and a particle velocity $\mathbf{v}^{n+1/2}$ is calculated at $t^{n+1/2} = (n + 1/2)\tau$. The scheme have the second approximation order on step τ and is stable.

Let us consider now the realization of the requirements stipulated by the specific character of the particle method. It is easy to see that the scheme is very simple in realization. It is explicit and one-step. For checking of the scheme reversibility in time it can be replaced τ by $(-\tau)$ and indices $(n + 1), (n + 1/2)$ by $(n - 1), (n - 1/2)$ correspondingly. As a result we obtain

$$\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{-\tau} = \frac{q}{m} \mathbf{E}(\mathbf{r}^n),$$

$$\frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{(-\tau)} = \mathbf{v}^{n-1/2},$$

which is equivalent to the initial scheme. This means that under changing of the time direction to the inverse one the particle returns into its initial state.

The energy conservation law for a charged particle in the constant electric field has the form

$$\frac{mv^2(t)}{2} - \frac{mv^2(0)}{2} = q[\varphi(\mathbf{r}(t)) - \varphi(\mathbf{r}(0))].$$

Let us calculate the change of the particle kinetic energy during one time step for scheme (16). For simplicity we assume that the coordinate axis x is directed along the electric field and the particle moves along the axis x . Then

$$\begin{aligned} \frac{m}{2}[(v^{n+1/2})^2 - (v^{n-1/2})^2] &= \frac{m}{2}(v^{n+1/2} + v^{n-1/2})(v^{n+1/2} - v^{n-1/2}) = \\ &= (v^{n+1/2} + v^{n-1/2}) \frac{\tau q}{2} (v^{n+1/2} + v^{n-1/2}) \frac{\partial \varphi(x^n)}{\partial x} = \\ &- \frac{\tau q}{2} (v^{n+1/2} + v^{n-1/2}) \left[\frac{\varphi(x^n + \frac{\tau}{2}v^{n+1/2}) - \varphi(x^n - \frac{\tau}{2}v^{n-1/2})}{\frac{\tau}{2}(v^{n+1/2} + v^{n-1/2})} + O(\tau^2) \right] = \\ &= -q \left[\varphi \left(\frac{x^n + x^{n+1}}{2} \right) - \varphi \left(\frac{x^{n-1} + x^n}{2} \right) \right] + O(\tau^3). \end{aligned}$$

Hence, it follows that for the finite number N of time steps, when $N\tau \sim 1$, the energy is conserved with the accuracy $O(\tau^2)$. The program on Fortran-77 realizing the scheme (16) is presented in Supplement B.

The defect of the scheme is in the fact that the particle coordinates and velocities are calculated on different time levels.

If the coordinates and velocities are calculated at the same time moment, then we can suggest the following scheme

$$\left\{ \begin{array}{l} \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\tau} = \frac{q}{m} \mathbf{E}(\mathbf{r}^n + \frac{\tau}{2} \mathbf{v}^n), \\ \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} = \frac{\mathbf{v}^{n+1} + \mathbf{v}^n}{2}. \end{array} \right.$$

2°. The particle motion in the magnetic field.

In this case, the best scheme is

$$\begin{aligned} \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\tau} &= \left[\frac{\mathbf{v}^n + \mathbf{v}^{n+1}}{2}, \boldsymbol{\omega} \right], \\ \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} &= \frac{\mathbf{v}^n + \mathbf{v}^{n+1}}{2}, \quad \boldsymbol{\Omega} = \frac{q}{mc} \mathbf{B}. \end{aligned} \tag{17}$$

It has the second order of the approximation, it is absolutely stable and one-step. The scheme is implicit, but because of its linearity one can easily derive the explicit expressions for the velocity components. They have the form

$$v_x^{n+1} = (1 + s^2)^{-1} [(1 - s^2)v_x^n + 2sv_y^n],$$

$$v_y^{n+1} = (1 + s^2)^{-1} [(1 - s^2)v_y^n - 2sv_x^n], \quad s = \frac{|\Omega|\tau}{2},$$

from which it follows that the kinetic energy is conserved by the scheme (17) exactly.

As it is known, the charged particle in the constant uniform magnetic field moves with the constant velocity on the circle in the plane which is perpendicular to the vector of the magnetic field. The radius of the circle equals to $R_0 = |\mathbf{v}|/|\Omega|$ and the rotation period is $T_0 = 2\pi/|\Omega|$. It can be shown that the particle whose coordinates and velocity are calculated according to the scheme (17) in the constant magnetic field will be moved on the circle, and the radius of this circle will coincide with the theoretical value. The rotation period will be equal to

$$T = \frac{2\pi\tau}{\arcsin[|\Omega|\tau/(1 + |\Omega|^2\tau^2/4)]}.$$

3°. The particle motion in the electric and magnetic fields.

The motion of an individual particle can be locally expanded into two independent components. The first one is the motion along the magnetic field lines. It takes place under the action of the electric field component which is parallel to the magnetic field \mathbf{B} . The second component of motion locates on the plane perpendicular to magnetic field \mathbf{B} . It can also be expanded into two components that are the rotation on some circle with frequency $\Omega = |q|\mathbf{B}/mc$ and with radius $R = |\mathbf{v}|/|\Omega|$, and the displacement of this circle in the direction which is perpendicular to the electric and magnetic fields. The latter motion is realized with the so called drift velocity:

$$\mathbf{v}_d = \frac{q}{m|\Omega|^2} [\mathbf{E}, \Omega] = \frac{c}{B^2} [\mathbf{E}, \mathbf{B}].$$

These peculiarities of the movement can serve as a good test for checking the quality of numerical schemes.

In this case, the simplest scheme is the generalization of the scheme (17)

$$\begin{aligned} \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\tau} &= \frac{q}{m} \mathbf{E}(\mathbf{r}^n) + \left[(\mathbf{v}^n + \mathbf{v}^{n+1}), \frac{\Omega}{2} \right], \\ \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} &= \frac{\mathbf{v}^n + \mathbf{v}^{n+1}}{2}. \end{aligned} \tag{18}$$

Writing down the equations (18) in the components, one can obtain the expressions for the velocity at the time moment $t = t^{n+1}$

$$\begin{aligned} v_x^{n+1} &= (1 + s^2)^{-1} \left[(1 - s^2)v_x^n + 2sv_y^n + q\frac{\tau}{m}(E_x + sE_y) \right], \\ v_y^{n+1} &= (1 + s^2)^{-1} \left[(1 - s^2)v_y^n - 2sv_x^n - q\frac{\tau}{m}(sE_x - E_y) \right]. \end{aligned} \quad (19)$$

This scheme reproduces exactly the trajectory and satisfies the energy conservation law in constant electric and magnetic fields. It is time-reversible and has the second order of accuracy on τ . If the particle velocity is calculated at the time moment $t^{n+1/2} = (n + 1/2)\tau$, then the scheme takes the form

$$\begin{aligned} \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\tau} &= \frac{q}{m}\mathbf{E}(\mathbf{r}^n) + \left[(\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}), \frac{\Omega}{2} \right], \\ \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} &= \mathbf{v}^{n+1/2}. \end{aligned} \quad (20)$$

If we are not interested in cyclotron rotation, but in some average particle movement, then it is desirable to carry out the calculations with greater steps τ in comparison with $|\Omega|^{-1}$ ($|\Omega|\tau \gg 1$). In this case, the algorithm (18) describes the drift in crossed fields \mathbf{E}, \mathbf{B} . Really, from formulas (19) at $s \gg 1$ it follows:

$$\frac{v_x^n + v_x^{n+1}}{2} = cE_y/B, \quad \frac{v_y^n + v_y^{n+1}}{2} = -cE_x/B, \quad B \equiv |\mathbf{B}|,$$

that coincides with the known formula for the drift velocity

$$\mathbf{v}_d = c \frac{[\mathbf{E}, \mathbf{B}]}{B^2}.$$

Let us construct the algorithm using the splitting of particle motion on the cyclotron rotation and drift in fields \mathbf{E}, \mathbf{B} . For this the velocities is calculated in three stages:

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{v}^n - c \frac{[\mathbf{E}, \mathbf{B}]}{B^2}, \\ (\mathbf{v}_2 - \mathbf{v}_1)/\tau &= \left[\frac{\mathbf{v}_1 + \mathbf{v}_2}{2}, \frac{\Omega}{2} \right], \\ \mathbf{v}^{n+1} &= \mathbf{v}_2 + c \frac{[\mathbf{E}, \mathbf{B}]}{B^2}. \end{aligned} \quad (21)$$

For determining of the particle coordinates one can use the scheme (18).

Let us consider the splitting of particle motion in two components, one of which is defined only by electric field and other is defined only by magnetic field. Then we obtain the following scheme [79]:

$$\begin{aligned}\frac{\mathbf{v}_1 - \mathbf{v}^n}{\tau/2} &= q \frac{\mathbf{E}}{m}, \\ \frac{\mathbf{v}_2 - \mathbf{v}_1}{\tau} &= \left[\frac{\mathbf{v}_1 + \mathbf{v}_2}{2}, \frac{\Omega}{2} \right], \\ \frac{\mathbf{v}^{n+1} - \mathbf{v}_2}{\tau/2} &= q \frac{\mathbf{E}}{m}.\end{aligned}\quad (22)$$

In this scheme the motion in the magnetic field is centered relatively to the time moment $t^{n+1/2}$.

When the magnetic field is directed at the arbitrary angle to coordinate axes, the equations for the components of vector \mathbf{v}_2 are rather cumbersome. Because of that the realization of the second step in the schemes (21), (22) is more convenient to fulfill in two stages [79]:

$$\mathbf{v}_3 = \mathbf{v}_1 + a_1 [\mathbf{v}_1, \mathbf{B}], \quad \mathbf{v}_2 = \mathbf{v}_1 + a_2 [\mathbf{v}_3, \mathbf{B}], \quad (23)$$

where

$$a_1 = \tan(q\tau B/2mc)/B, \quad a_2 = \frac{2a_1}{1 + a_1^2 H^2}, \quad B \equiv |\mathbf{B}|.$$

The relativistic generalization of the leapfrog scheme (20) looks like as follows

$$\begin{aligned}\mathbf{U}^{n+1/2} - \mathbf{U}^{n-1/2} &= 2\alpha \left(\mathbf{E}^n + \frac{1}{\gamma^n c} [\mathbf{U}^n, \mathbf{B}^n] \right), \\ \mathbf{r}^{n+1} - \mathbf{r}^n &= \tau \frac{\mathbf{U}^{n+1/2}}{\gamma^{n+1/2}}.\end{aligned}\quad (24)$$

Here $\mathbf{U} \equiv \gamma \mathbf{v}$ is a relativistic velocity, $\gamma \equiv \sqrt{1 + \frac{U^2}{c^2}}$ is a relativistic factor, $\alpha = q\tau/2m$.

Under realization scheme (24) one can substitute \mathbf{U}^n by its average magnitude

$$\mathbf{U}^n = \frac{\mathbf{U}^{n-1/2} + \mathbf{U}^{n+1/2}}{2},$$

and for the approximation γ^n can be used the fact that the motion in the magnetic field leads only to the turn of the velocity vector of the particle round the axis parallel \mathbf{B} and it doesn't change its absolute value. Therefore for calculation of the magnitude γ^n with the second order on τ it's enough to set

$$\gamma^n \approx \gamma^- = \sqrt{1 + \frac{|\mathbf{u}^-|^2}{c^2}},$$

where

$$\mathbf{u}^- = \mathbf{U}^{n-1/2} + \alpha \mathbf{E}^n,$$

After determination of γ^n one can determine vector $\mathbf{U}^{n+1/2}$ directly from (24), because for finding of the three components of this vector we have the system of three algebraic equations. The value of the relativistic velocity $\mathbf{U}^{n+1/2}$ can be found in another way using the variable \mathbf{u}^- and the variable \mathbf{u}^+ defined as follows

$$\mathbf{u}^+ = \mathbf{U}^{n+1/2} - \alpha \mathbf{E}^n.$$

In these variables the first equation of the system (24) takes the form:

$$\mathbf{u}^+ - \mathbf{u}^- = [(\mathbf{u}^+ + \mathbf{u}^-), t], \quad (25)$$

where

$$\mathbf{t} = \frac{\alpha |\mathbf{B}^n|}{\gamma^n c} \mathbf{b}^n, \quad \mathbf{b}^n = \frac{\mathbf{B}^n}{|\mathbf{B}^n|}.$$

The sequence of steps for finding \mathbf{u}^+ was suggested in [79]. For determination \mathbf{u}^+ from (25) we first find the magnitude \mathbf{u}' :

$$\mathbf{u}' = \mathbf{u}^- + \mathbf{u}^- \times \mathbf{t},$$

And then values of \mathbf{u}^+ is determined from the formula

$$\mathbf{u}^+ = \mathbf{u}^- + \frac{2}{1 + |\mathbf{t}|^2} \mathbf{u}' \times \mathbf{t}. \quad (26)$$

The correctness (26) can be easily proved if to use the initial equation (25) and the equations obtained from (25) as the result of vector and scalar multiplication by \mathbf{t} .

Finally, the value of the relativistic velocity $\mathbf{U}^{n+1/2}$ is determined after the second "semi-acceleration" by the electric field from the expression

$$\mathbf{U}^{n+1/2} = \mathbf{u}^+ + \alpha \mathbf{E}^n.$$

After this from the second equation (24) the location of the particle is derived at the time moment $t = t^{n+1}$. The magnitude $\gamma^{n+1/2}$ is determined by the expression

$$\gamma^{n+1/2} = \sqrt{1 + \left(\frac{\mathbf{U}^{n+1/2}}{c} \right)^2}.$$

5.4. Conservation laws in model plasma

In many physical processes in real plasma the conservation laws of mass, charge, momentum and full energy are satisfied. The specific character of plasma problems usually requires that these laws would be satisfied with good accuracy also in the numerical simulation. In the particle method mass and charge of particles are conserved exactly

because they are individual features of the particles which don't change in the time.

Let us consider the question of conservation of the full momentum and full energy on a simple example of a one-dimensional electrostatic problem with one type of particles. In this case, the electric field and charge density are connected with the potential by formulas

$$E = -\frac{\partial \varphi}{\partial x}, \quad \frac{\partial^2 \varphi}{\partial x^2} = -4\pi\rho(x). \quad (27)$$

The full momentum of the system of particles is equal to

$$P = \sum_{j=1}^M m_j v_j,$$

Assume that the integration on time of dynamic equations is realized exactly:

$$\frac{dP}{dt} = \sum_j m_j \dot{v}_j = \sum_j F(x_j). \quad (28)$$

Here the force $F(x_j)$ is determined on formulas (8), (9) in the form

$$F(x_j) = q \int_{-\infty}^{+\infty} R(x' - x_j) \sum_i E_i S(x' - x_i) dx',$$

where i is a current index of the mesh nodes on which the summation is carried out. Substituting into (28), one can obtain

$$\frac{dP}{dt} = \sum_j q_j \sum_k E_k \int_{-\infty}^{\infty} R(x' - x_j) S(x' - x_k) dx'.$$

Let us introduce the special mesh kernel \bar{R} by the relation

$$\int_{-\infty}^{\infty} R(x' - x_j) S(x' - x_i) dx' = h \bar{R}(x_j - x_i). \quad (29)$$

Then we have

$$\frac{dP}{dt} = h \sum_j q_j \sum_i E_i \bar{R}(x_j - x_i) = h \sum_i \rho_i E_i, \quad (30)$$

where the charge density in i -th node of the mesh is determined by the formula

$$\rho_i = \sum_{j=1}^M q_j \bar{R}(x_i - x_j).$$

Using the Fourier representation (1.48), one can write down equation (30) in the form

$$\begin{aligned} \frac{dP}{dt} &= h \sum_i \rho_i \int_{-\pi/h}^{\pi/h} \frac{dk}{2\pi} \hat{E}(k) \exp(ikx_i) = \\ &\int_{-\pi/h}^{\pi/h} \frac{dk}{2\pi} \hat{E}(k) h \sum_i \rho_i \exp(ikx_i) = \int_{-\pi/h}^{\pi/h} \frac{dk}{2\pi} \hat{E}(k) \hat{\rho}(-k). \end{aligned} \quad (31)$$

Let us choose the follow difference approximation of the equations (27):

$$\begin{aligned} E_i &= -\frac{\varphi_{i+1} - \varphi_{i-1}}{2h}, \\ \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{h^2} &= -4\pi\rho_i. \end{aligned} \quad (32)$$

Substitution into (32) of the Fourier representation of the mesh functions gives us the following connections between the Fourier transforms

$$\begin{aligned} \hat{E}(k) &= -i\psi(k)\hat{\varphi}(k), \\ 4\pi\hat{\rho}(k) &= K^2(k)\hat{\varphi}(k), \end{aligned} \quad (33)$$

where $\psi(k) = \frac{1}{h}\sin kh$, $K(k) = \frac{2}{h}\sin \frac{kh}{2}$. By substitution Eqs. (33) into Eq. (31) we derive

$$\frac{dP}{dt} = -i \int_{-\pi/h}^{\pi/h} \frac{dk}{2\pi} \psi(k) K^2(k) \hat{\varphi}(k) \hat{\varphi}(-k). \quad (34)$$

As $\psi(k)$ is an odd function, the integrand is odd and the integral equals zero. Thus, the conservation law of the full momentum for model particle system will be satisfied if the symmetric approximation (32) of the Poisson equation and special interpolation (29) of the force field are used. In general case the momentum may not conserve.

Now let us consider the question of the full energy conservation. The full electric field energy in our one-dimensional electrostatic problem can be written in two equivalent forms [22]

$$W^{(1)} = \int \frac{E^2}{8\pi} dx, \quad W^{(2)} = \int \frac{\rho\varphi}{2} dx, \quad W^{(1)} = W^{(2)}.$$

The simplest quadrature formulas approximated this integrals have the form:

$$W_h^{(1)} = h \sum_i \frac{E_i^2}{8\pi}, \quad W_h^{(2)} = h \sum_i \frac{1}{2} \rho_i \varphi_i,$$

and in general case, $W_h^{(1)} \neq W_h^{(2)}$. In order to show this, we represent the mesh functions by formulas (1.48) and substitute into the expressions for energy. It can be deduced

$$W_h^{(1)} = \frac{1}{16\pi} \int_{-\pi/h}^{\pi/h} \hat{E}(k) \hat{E}(-k) dk,$$

$$W_h^{(2)} = \frac{1}{16\pi} \int_{-\pi/h}^{\pi/h} \hat{E}(k) \hat{E}(-k) \cos^{-2} \frac{kh}{2} dk.$$

While deducing the formula for $W_h^{(2)}$ the first of the relations was used. Thus, the considered quadrature formulas for field energy are not equivalent, but at small steps of the mesh h this difference is of the order (h^2).

In [80] it was shown that the change of the field energy and the kinetic energy of particles can be written in the following form

$$\frac{dW_h^{(1)}}{dt} = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{E}(-k) \hat{j}(k) \frac{kh}{2} \operatorname{ctg} \frac{kh}{2} dk,$$

$$\frac{dW_h^{(2)}}{dt} = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{E}(-k) \hat{j}(k) \frac{kh}{\sin kh} dk,$$

$$\frac{dK}{dt} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{E}(-k) \hat{j}(k) dk,$$

where $\hat{j}(k)$ is the Fourier transform of the mesh current density. From these expressions it follows that the change of full energy $W_k^{(1,2)} + K$ is not equal to zero but at small h this difference is small.

If we change the calculation algorithm of the force acting on the particle and don't use the formula (32), then it can be achieved the conservation of the full energy $K + W_h^{(2)}$ [81]. Let us introduce the potential energy

$$W_j(x_j) = q_j h \sum_i \bar{R}(x_i - x_j) \varphi_i \quad (35)$$

and calculate the force acting upon the particle with number j as follows

$$F_j = -\frac{\partial}{\partial x_j} W_j(x_j) = -q_j h \sum_i \varphi_i \frac{\partial}{\partial x_j} \bar{R}(x_i - x_j). \quad (36)$$

In contrast to the algorithm considered above the numerical differentiation of the potential is not realized here, and the function determining the mesh kernel is differentiated analytically.

Let us find the change of the kinetic energy using the expression (36) for the force

$$\frac{dK}{dt} = \frac{d}{dt} \sum_j m_j v_j^2 / 2 = \sum_j v_j m_j \dot{v}_j = \sum_j v_j F_j =$$

$$-h \sum_i \varphi_i \sum_j \dot{x}_j \frac{\partial}{\partial x_j} q_j \bar{R}(x_i - x_j) = -h \sum_i \varphi_i \frac{d}{dt} \sum_j q_j \bar{R}(x_i - x_j).$$

The sum $\sum_j q_j \bar{R}(x_i - x_j)$ is the change of density in the node with index i . Therefore the change of the kinetic energy is equal to

$$\frac{dK}{dt} = -h \sum_i \dot{\rho}_i \varphi_i.$$

The full energy of the system is equal to $K + W_h^{(2)}$. Let us find the change of the full energy:

$$\frac{d}{dt} (K + W_h^{(2)}) = (h/2) \sum_i (\dot{\rho}_i \varphi_i - \rho_i \dot{\varphi}_i).$$

Using the Fourier harmonic amplitudes and the connection between $\hat{\rho}(k)$ and $\hat{\varphi}(k)$, from the difference approximation of the Poisson equation we obtain that

$$\sum_i (\dot{\rho}_i \varphi_i - \rho_i \dot{\varphi}_i) = 0$$

and, consequently, the full energy is conserved. In the considered algorithm we can use any symmetric approximation of the Poisson equation. However, in this algorithm the full momentum of the system is not conserved. Really,

$$\begin{aligned} \frac{dP}{dt} &= \sum_j F_j = -h \sum_j q_j \sum_i \varphi_i \frac{\partial}{\partial x_j} \bar{R}(x_i - x_j) = \\ &= h \sum_j q_j \sum_i \varphi_i \int_{-\infty}^{+\infty} \frac{dk}{2\pi} i k \hat{R}(k) \exp[i k(x_i - x_j)] = \\ &= \sum_j q_j \int_{-\infty}^{+\infty} \frac{dk}{2\pi} i k \hat{R}(k) \varphi(-k) \exp(-ikx_j) = \\ &= \int_{-\pi/h}^{+\pi/h} \frac{dk}{2\pi} \varphi(-k) \sum_p ik_p \hat{R}(k_p) \sum_j q_j \exp(-ik_p x_j). \end{aligned} \quad (37)$$

If in expression (37) we neglect by the mesh harmonics, that means that we assume $p = 0$, then

$$\frac{dP}{dt} = i \int_{-\pi/h}^{+\pi/h} \frac{dk}{2\pi} k \varphi(-k) \rho(k).$$

As far as the integrand is odd with respect to k , as in Eq. (34), then

$$\frac{dP}{dt} = 0$$

and the total momentum of the system is conserved. However, the presence of mesh harmonics leads to the fact that the considered algorithm satisfy the energy conservarion law, but doesn't conserve the total momentum.

5.5. Examples of applications

Some examples are presented in this point which demonstrate the possibilities of the particles-in-cells method in the characteristic problems of the computational plasma physics. The examples follow in the order of increasing of the computational complexity beginning from the one-dimensional non-stationary problem with particles of one species up to the three-dimensional non-stationary problem with two-component model plasma. The choice of problems reflects the chronology of the development of computational aspects of the method expressed in the successive increasing of the number of model particles, growth a number of nodes in the Euler mesh, the transition towards two-dimensional and lately to three-dimensional problems. Here it can be mentioned a curious fact concerning the founder of the particle-in-cell method F.Harlow who believed that the method never to be used for three-dimensional problems. The examples illustrate the level of computational complexity reached in the plasma physics problems and can serve for the orientation of specialists working in other areas of computational mathematics. Simultaneously the examples show simulation possibilities of the method because the physical effects not accessible by other computational methods are reproduced by it. But it should be noted that from last point of view they can be estimated only by the readers who are specializing in the plasma physics problems.

5.5.1. The dynamics of ion-acoustic waves of the arbitrary amplitude in non-isothermal plasma

As the first example we present the use of the particle method in the problem of decay of arbitrary discontinuity of ion density in non-isothermal plasma when the electron temperature significantly exceeds the temperature of ions: $T_e \gg T_i$. Such conditions are often realized in laboratory experiments and in the plasma of solar wind [7]. For simplicity we consider the one-dimensional problem.

For solution of this problem the combined model was chosen in which the motion only of the ion component is considered and the density of electrons is described by the Boltzmann distribution

$$\rho(x) = \rho_0 \exp\left(\frac{e\varphi}{T}\right).$$

The necessity of such combined an approach is caused by a high difference between characteristic times of motion of electrons and ions. We assume that the temperature of ions to be equal to zero i.e. $T_i = 0$.

The system of equations initially has the following form

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} - \frac{e}{m_i} \frac{\partial \varphi}{\partial x} \frac{\partial f}{\partial u} = 0, \quad (38)$$

$$\Delta\varphi = 4\pi e \left[n_0 \exp\left(\frac{e\varphi}{T}\right) - \int_{-\infty}^{\infty} f du \right], \quad (39)$$

$$E(x) = -\frac{\partial\varphi}{\partial x}, \quad (40)$$

where f is an ion distribution function, u is velocity of ions, φ is a potential of electric field, T is a temperature of electrons, n_0 is non-perturbed plasma number density, $E(x)$ is an electric field. From Eq. (38) in the usual way we obtain the equations of particle motion (7), (8). In this case, we have the following system of equations

$$\begin{aligned} \frac{dx_j}{dt} &= u_j, & \frac{du_j}{dt} &= E(x_j), \\ \beta \frac{\partial^2 \varphi}{\partial x^2} &= \exp(\varphi) - \rho. \end{aligned} \quad (41)$$

Here j is a particle index, the potential is measured in units T_e/e , velocity u is expressed in units of ion-acoustic velocity $c_s = T_e/m_i$, $\beta = (D/L_0)^2$, $D = (4\pi n_0 e^2/T)^{-1/2}$ is the Debye radius, L_0 is the characteristic length.

We divide interval $[0, L]$ into K equal subintervals (cells) by the nodes with coordinates $x_k = kh$, $k = 0, 1, \dots, K$; h is a step of the space mesh. At the initial time moment N_k immovable particles are placed into every cell $[x_{k-1}, x_k]$, the number of them is proportional to the plasma density. Potential φ_k and density ρ_k are calculated in the mesh nodes, and the electric field $E_{k-1/2}$ is calculated in the cell centers $(x_{k-1} + x_k)/2$.

Density ρ_k is calculated by NGP-model (1.31), that means the mesh kernel

$$\bar{R}(x) = \begin{cases} \frac{1}{h}, & |x| \leq \frac{h}{2}, \\ 0, & |x| > \frac{h}{2}, \end{cases}$$

is used in the formula

$$\tilde{\rho}_{k-1/2} = \sum_j \bar{R}(x_{k-1/2}, x_j).$$

For reducing the fluctuations a mesh density in nodes x_k is calculated with using of the formula

$$\rho_k = \frac{1}{2}(\tilde{\rho}_{k+1/2} + \tilde{\rho}_{k-1/2}). \quad (42)$$

The electric field is interpolated into point x_j of particle location by formula (11)

$$E(x_j) = \frac{1}{h} \left[(x_{k+1/2} - x_j) E_{k-1/2} + (x_j - x_{k-1/2}) E_{k+1/2} \right],$$

where

$$x_j \in [x_{k-1/2}, x_{k+1/2}].$$

The solution of Eq. (41) at the next time step τ is realized by two stages.

The Lagrangian step. For every particle with index j with the known values of velocity u_j^n and coordinate x_j^n at the time moment $t^n = n\tau$ a new velocity u_j^{n+1} and coordinate x_j^{n+1} are determined at the next time moment $t^{n+1} = (n+1)\tau$ by formulas

$$u_j^{n+1} = u_j^n + \tau E(x_j^n),$$

$$x_j^{n+1} = x_j^n + \tau u_j^{n+1}.$$

Then the number of particles which appeared in every cell is calculated, and the plasma density ρ_k^{n+1} is determined in the nodes x_k of the space mesh according to formula (42).

The Euler step. The Poisson equation for the potential is solved with the found values of density ρ_k^{n+1} and the electric field is determined.

Let us consider the solution of the Poisson equation

$$\beta \frac{\partial^2 \varphi^{n+1}}{\partial x^2} = \exp(\varphi^{n+1}) - \rho^{n+1}(x) \quad (43)$$

under boundary conditions

$$\varphi(0) = \varphi_0, \quad \varphi(L) = 0.$$

For solving of Eq. (43) one can use the method of quasilinearization in which the solution of nonlinear equation (43) φ^{n+1} is determined as limit of sequence $\{\varphi^{n+1,s}\}$ of solutions of the linear equation

$$\beta \frac{\partial^2 \varphi^{n+1,s+1}}{\partial x^2} = \exp(\varphi^{n+1,s})(1 + \varphi^{n+1,s+1} - \varphi^{n+1,s}) - \rho^{(n+1)}(x). \quad (44)$$

As the initial approach $\varphi^{n+1,0}$ we used the solution of equation (43) at the previous time moment that means that $\varphi^{n+1,0} = \varphi^n$. In finite differences Eq. (44) has the following form:

$$\begin{aligned} & \frac{\beta}{h^2} \varphi_{k+1}^{n+1,s+1} - \left(\frac{2\beta}{h^2} + \exp(\varphi_k^{n+1,s}) \right) \varphi_k^{n+1,s+1} + \\ & + \frac{\beta}{h^2} \varphi_{k-1}^{n+1,s+1} = \exp(\varphi_k^{n+1,s})(1 - \varphi_k^{n+1,s}) - \rho_k^{n+1}, \end{aligned} \quad (45)$$

which solution is determined by the sweep method. The iterative process is continued up to fulfillment of the condition

$$\max_k |\varphi_k^{n+1,s+1} - \varphi_k^{n+1,s}| < \varepsilon.$$

The mesh with the number of nodes $K = 100$ was used in the calculations. The full quantity of model particles was 3000. $\beta = 0.0025$, that means that there were 20 Debye radii in the unit of length.

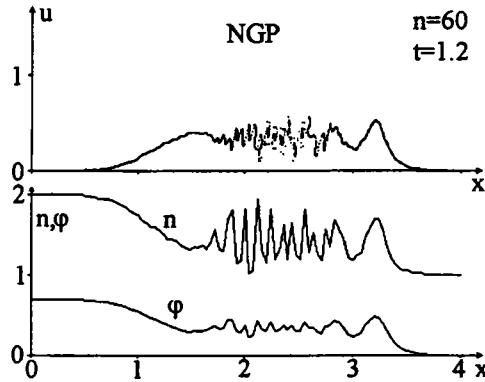


Fig. 30. The decay of the density discontinuity in plasma calculated by the NGP-model ($A = 2$, $t = 1.2\omega_{pi}^{-1}\beta^{-1/2}$).

Let us consider the results of the problem solution. At comparatively small density drop $A = n^-(x_p)/n^+(x_p) < 5$ in the discontinuity point x_p , as it should be expected, the decay of the step-like data leads to the formation of going to the right laminar shock wave and going to the left rarefaction wave (Fig. 30). At the upper plot the positions of individual model particles in the plane (x, u) are shown. At the lower plot the profiles of ion density and the potential are shown. As it is seen from the figures, the shock wave has oscillation leading edge caused by dispersion at the separation of charges which is accompanied by the segment with the approximately constant amplitude φ . The amplitude of the front soliton increases up to some value $\varphi_{max} < \varphi_*$. The breakdown of the front and intermixing of particles don't take place. For comparison in Fig. 31 the solution of this problem is shown with the use of the kernel of PIC-model. The full number of model particles here was 1000.

At initial density drops $5 < A < 13$ the amplitude of the front soliton increases up to values $\varphi_{max} > \varphi_*$, and after this the breakdown of the wave takes place with the formation of fast particles reflected by the shock wave front. The calculations show that if $\varphi < \varphi_*$, then ejection of particles has the pulsation character. In this conditions after increasing of the amplitude of the front soliton up to $\varphi_{max} > \varphi_*$ the reflection of some number of particles by the front takes place, and the front soliton amplitude decreases up to $\varphi < \varphi_*$, then again the amplitude of the soliton increases using the energy of the step-wise data, and a new ejection of particles occurs and so on.

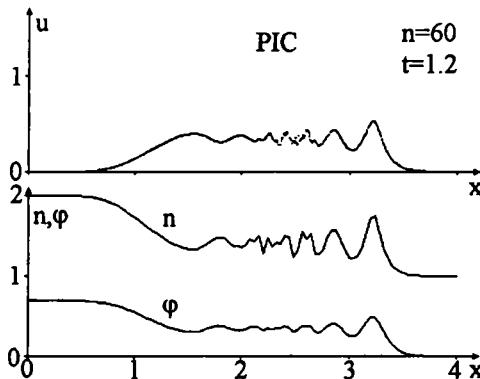


Fig. 31. The decay of the plasma density discontinuity calculated by the PIC-model ($A = 2$, $t = 1.2\omega_{pi}^{-1}\beta^{-1/2}$).

The increasing of the density drop amplitude A and, correspondingly, the amplitude of potential φ , when $A > 13$, $\varphi > \varphi_*$ leads to formation of the shock wave without regular oscillations but with a sharp front. In this case, the continuous reflection of particles from the front takes place, and it is accompanied by the deceleration of the basic wave (see Fig.32).

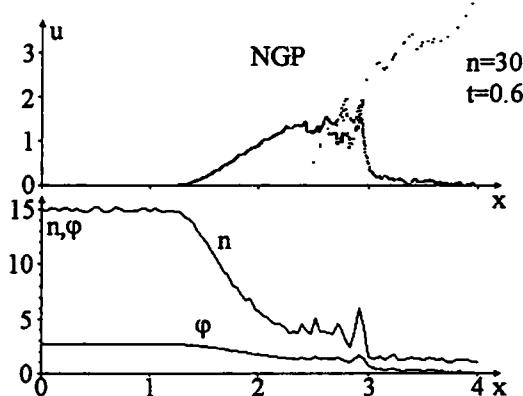


Fig. 32. The decay of plasma density discontinuity calculated by NGP-model ($A = 15$, $t = 1.2\omega_{pi}^{-1}\beta^{-1/2}$).

From the upper plot it is seen that at $x > 3$ there are two groups of particles in the computational domain which differ essentially from each other by their velocities. The first group consists of ions of initially stationary front, and the second one consists of the particles accelerated by the electric field of the shock wave front. It is obviously, that by hydrodynamic approximation, where an every cell has the only mean velocity it's impossible to describe such flows.

However, in the most cases the real plasma is multi-component consisting of electrons and ions of several species. This is, for instance, the laser plasma, solar wind, ionospheric plasma, laboratory plasma in some experimental setups, etc. For description of the structure of ion-acoustic waves in the non-isothermal multi-component plasma with cold ions at the amplitudes below some critical ones, the system of hydrodynamic equations can be used. For the first time the structure analysis of ion-acoustic solitons in plasma with two kinds of unit charge ions (heavy and light) was carried out in the paper [82]. In this paper it was shown that if the concentration of the light ion component exceeds some critical value, then in such a system stationary soliton solutions is absent.

The critical value of dimensionless concentration of the light component $\alpha = n_2/n_0$ is determined by the formula

$$\alpha_{cr} = \frac{U_{cr}^{-2}(e^{U_{cr}^2/2\mu} - 1) + (1 - \frac{1}{\mu})^{\frac{1}{2}} - 1}{(1 - \frac{1}{\mu})^{\frac{1}{2}}(1 - (1 - \frac{1}{\mu})^{\frac{1}{2}})}, \quad (46)$$

where $\mu = \frac{m_1}{m_2}$ is a mass ratio of the heavy and light components, U_{cr} is a critical velocity of the ion-acoustic soliton in the three-component strongly non-isothermal plasma. At $\mu \gg 1$ and small α it's easy to obtain

$$U_{cr}^2 = 1 + 4\mu\alpha + \frac{1}{3\mu}. \quad (47)$$

At the amplitudes which are more than the critical amplitude the full reflection of ions of the light component from the "hill" of electrostatic potential will takes place. It is clear, that within of the stationary problem the formation of such ion-acoustic waves appears to be impossible because we should take into consideration the influence of the reflected ion beam on the structure of ion-acoustic waves. It is well known (see [83]) that the ion beam reflected from potential "hills" brings the specific mechanism of collisionless dissipation into the system, and it leads to formation of a collisionless ion-acoustic shock wave.

For description of the evolution of localized perturbations with the arbitrary amplitudes in multicomponent plasma the system of equations of electrostatic plasma was used. This model is the generalization of the system Eqs. (38), (39) in a case of multicomponent plasma:

$$\frac{\partial f_l}{\partial t} + u \frac{\partial f_l}{\partial x} - \frac{Z_l e}{m_l} \frac{\partial \varphi}{\partial x} \frac{\partial f_l}{\partial u} = 0, \quad (48)$$

$$\frac{\partial^2 \varphi}{\partial x^2} = 4\pi e(n_0 \exp\left(\frac{e\varphi}{T_e}\right) - \sum_{l=1}^N Z_l n_l), \quad (49)$$

$$n_l = \int f_l(x, u, t) du, \quad (50)$$

Here index l enumerates of ion species.

At the Lagrangian step the motion of each ion species was traced, the location of particles were determined and the interpolation of ion density of every component was derived into the nodes of the Euler mesh with the use of the PIC-model kernel. Integration of equations of the motion of each ion species was realized with the use of scheme (16).

At the Euler step the Poisson equation was solved. For its solution the method of quasilinearization Eq. (45) was used. The length of modeling domain in calculations was $L = 500D$, D is a Debye radius, the number of the Euler mesh cells was $I = 2000$, and the full number of model ions of both species was $N = 10^4 \div 5 \cdot 10^4$.

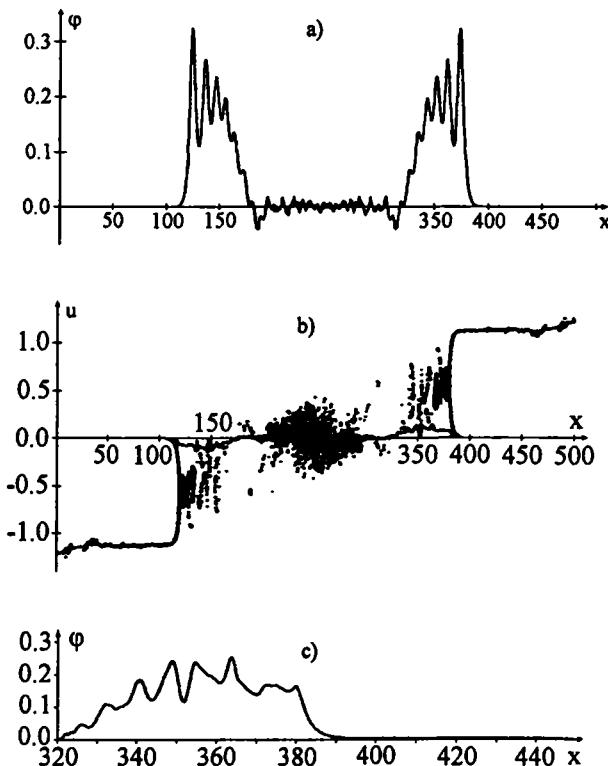


Fig. 33. The decay of the local density perturbation with $A = 1.5$:
a) — the distribution of the potential in two-component plasma;
b) — the phase plane (u, x); c) — potential ($\alpha_1 = 0.99$, $\alpha_2 = 0.01$, $m_1/m_2 = 4$).

Let us consider the problem of a decay of the local density perturbations in three-component plasma consisted of the electrons, and heavy and light ions. The problem is formulated in the following way. At

the initial time moment $t = 0$ in plasma (in the domain center) the local perturbation of plasma concentration with half-width $\Delta x = 20D$ and the definite ratio α_1/α_2 of concentrations of the heavy and light components is initiated.

The results of simulations show that the structure of plasma formations arising during the decay of localized perturbation is governed by both the value of the initial density perturbation A and the content of the plasma components. For simplicity, we restrict ourselves to the case of singly charged ions $Z_1 = Z_2 = 1$. We assume that the value of the initial density perturbation $A = 1.5$ is fixed and analyze the variation of the structure of plasma perturbations depending on the ratio of the densities of the heavy and light plasma components (in simulations, we used $\mu = m_1/m_2 = 4$).

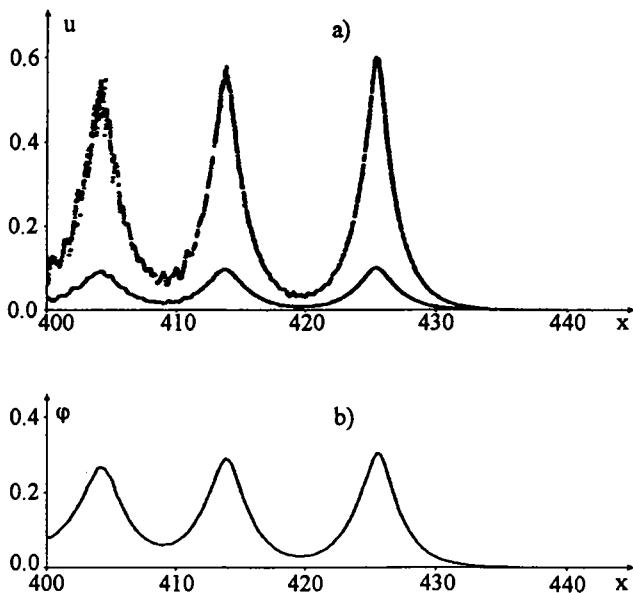


Fig. 34. "Unreflecting" regime of the local density perturbation with $A = 1.5$ at the time moment $t = 200$ for the case $\alpha_1 = 0.80$, $\alpha_2 = 0.20$, $m_1/m_2 = 4$; a) — the phase plane, b) — the potential distribution.

The decay of the localized density perturbation in a two-component plasma is accompanied by the formation of two symmetric, counterpropagating trains of solitons. Figure 33a displays the electrostatic-potential profile established during the decay of the localized density perturbations with $A = 1.5$ at the time moment $t = 100$. We can see two counterpropagating trains consisting of five solitons whose corresponding maximum amplitudes are nearly identical. The structure of plasma

perturbations changes significantly when a low-density light component is added to the plasma. Figure 33b shows the phase plane (u, x) for $\alpha_1 = 0.99, \alpha_2 = 0.01$ at the time $t = 2 \cdot 10^2$. In this case, all the incident light ions are reflected from the front of the potential barrier. The reflected ions, in turn, have a significant impact on the structure of this front: a plateau is appeared on the lower part of the potential profile (Fig. 33c).

When the density of the light ions increases (i.e. $0.05 \geq \alpha_2 \leq 0.15$), the ions are no longer reflected completely; instead, only a fraction of the particles is reflected from the head soliton at regular intervals, i.e., the regime of "repeated" reflections is established. When the density of the light ions increases further ($\alpha_2 \geq 0.15$) interaction occurs without reflections, i.e., no light ions are reflected from the front of the head soliton. Figures 34a and 34b show the corresponding phase plane for $\alpha_2 = 0.2$ and the potential profile.

It is obvious that for each ratio of heavy and light component concentrations the initial density drop A always exists at which the "unreflecting" regime of the decay of local perturbation will be realized.

5.5.2. Shock waves in plasma with the magnetic field

Here we consider the solution of the problem about one-dimensional non-stationary shock waves propagating across the magnetic field in rarefied quasineutral plasma. In such a plasma shock waves with amplitudes which are higher than some critical value must be considered on the basis of kinetic equations (1). At studying such shock waves in contrast to the examples considered in s.p. 5.5.1, it is necessary to calculate the motion of both ions and electrons. But the use of the particle method for calculation of both components of plasma is a technically difficult problem because of the large difference at time scales of the motion of ions and electrons.

Let us consider the statement of the problem about the evolution of collisionless shock waves in plasma with the magnetic field [85]. We assume that all functions depend only on coordinate x , the magnetic field is directed along axis z , the direction of the wave propagating coincides with axis x . This is the case of the perpendicular shock wave.

Equations of model ion motion derived from the kinetic Eq. (1) have the following form:

$$m_i \frac{du_i}{dt} = e \left(E_x + \frac{1}{c} v_i H \right),$$

$$m_i \frac{dv_i}{dt} = e \left(E_y - \frac{1}{c} u_i H \right) + m_e \nu (v_e - \langle v_i \rangle), \quad (51)$$

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

where x_i are coordinates of the ions, u_i , v_i are x, y -components of the velocity of individual ions, v_e , $\langle v_i \rangle$ are y -components of the macroscopic velocity of electrons and ions, H is a z -component of the magnetic field, ν is an effective collision frequency.

As plasma is in this problem quasineutral, then $n_i = n_e = n$. Therefore for x -component of macroscopic velocity of electrons we have $u_e = \langle u_i \rangle$.

For simplification one can describe the electronic component of plasma by the equations of gas dynamic type because the electron movement is closely connected with force lines of the magnetic field [22]

$$\begin{aligned} m_e \left(\frac{\partial u_e}{\partial t} + u_e \frac{\partial u_e}{\partial x} \right) &= -e \left(E_x + \frac{1}{c} v_e H \right) - \frac{1}{n} \frac{\partial p_e}{\partial x}, \\ m_e \left(\frac{\partial v_e}{\partial t} + u_e \frac{\partial v_e}{\partial x} \right) &= -e \left(E_y - \frac{1}{c} u_e H \right) - m_e \nu (v_e - \langle v_i \rangle), \end{aligned} \quad (52)$$

$$\frac{\partial T}{\partial t} + u_e \frac{\partial T}{\partial x} = (\gamma - 1) \left[-T \frac{\partial u_e}{\partial x} + m_e \nu \left(\frac{c}{4\pi en} \frac{\partial H}{\partial x} \right)^2 + \frac{\chi}{n} \frac{\partial}{\partial x} \left(n \frac{\partial T}{\partial x} \right) \right]. \quad (53)$$

In the equation for electron temperature T Joule heating of electrons under collisions with frequency ν and the electron heat conductivity with coefficient χ are taken into account. As the concrete forms of the effective collision frequency and heat conductivity doesn't influence the basic features of considered phenomena, we assume that $\nu = \text{const}$, $\chi = \text{const}$. The ratio of specific heats is $\gamma = \frac{5}{3}$, $\frac{m_i}{m_e} = 1836$.

The Maxwell equations for the magnetic field have the form

$$\begin{aligned} \frac{\partial H}{\partial t} &= -c \frac{\partial E_y}{\partial x}, \\ \frac{\partial H}{\partial x} &= \frac{4\pi en}{c} (v_e - \langle v_i \rangle), \end{aligned} \quad (54)$$

From the equations of electron motion (52) and Maxwell equations (54) one can derive the formulas for the components of the electric field

$$\begin{aligned} E_x &= -\frac{1}{(1+\beta)en} \frac{\partial}{\partial x} \left[p_e + (1-\beta) \frac{H^2}{8\pi} \right], \\ E_y &= \frac{1}{c} u_e H - \frac{m_e c \nu}{4\pi e^2 n} \frac{\partial H}{\partial x} - \frac{m_e c}{4\pi e^2 (1+\beta)} \left(\frac{\partial}{\partial t} + u_e \frac{\partial}{\partial x} \right) \left(\frac{1}{n} \frac{\partial H}{\partial x} \right), \end{aligned} \quad (55)$$

where $\beta = \frac{m_e}{m_i}$, $p_e = nTe$.

The further consideration is carried out in dimensionless variables. As characteristic scales we choose density n_0 , the magnetic field H_0 , the velocity $V_A = H_0/(4\pi m_i n_0)^{1/2}$, the length $c(4\pi n_0 e^2/m_i)^{-1/2}$, the electric field $V_A H_0/c$, the temperature $H_0^2/(8\pi n_0)$, the time $cm_i/(H_0 e)$.

In this case, the system of the equations takes the following form:

$$\begin{aligned} \frac{du_i}{dt} &= E_x + v_i H, \\ \frac{dv_i}{dt} &= E_y - u_i H + \frac{\kappa}{n} \frac{\partial H}{\partial x}, \\ \frac{dx_i}{dt} &= u_i, \\ E_x &= -\frac{1}{2n} \frac{\partial}{\partial x} (nT + H^2), \\ E_y &= uH - \frac{\kappa}{n} \frac{\partial H}{\partial x} - \beta \left(\frac{\partial}{\partial t} + u \frac{\partial}{\partial x} \right) \left(\frac{1}{n} \frac{\partial H}{\partial x} \right), \\ \frac{\partial H}{\partial t} + \frac{\partial}{\partial x} (uH) &= \kappa \frac{\partial}{\partial x} \left(\frac{1}{n} \frac{\partial H}{\partial x} \right) + \beta \frac{\partial}{\partial x} \left(\frac{\partial}{\partial t} + u \frac{\partial}{\partial x} \right) \left(\frac{1}{n} \frac{\partial H}{\partial x} \right), \\ \frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} &= (\gamma - 1) \left[-T \frac{\partial u}{\partial x} + 2\kappa \left(\frac{1}{n} \frac{\partial H}{\partial x} \right)^2 + \frac{\chi}{n} \frac{\partial}{\partial x} \left(n \frac{\partial T}{\partial x} \right) \right]. \end{aligned} \quad (56)$$

Here $\kappa = m_e c v / (e H_0)$.

At the initial time moment the ion density and the magnetic field have the break of value A

$$\frac{n^-(x_p)}{n^+(x_p)} = \frac{H^-(x_p)}{H^+(x_p)} = A.$$

The ion velocities equal zero, the temperature is constant

$$u(x, 0) = 0, \quad T(x, 0) = T_0.$$

At the boundaries of the computational domain the unperturbed conditions are set for all variables.

The domain $0 \leq x \leq L$, in which we search for a solution of the problem, is divided into K cells with step h . In correspondence with the initial conditions we set the values of functions n, u, H, T in the mesh nodes $x_k = kh$ ($k = 0, 1, \dots, K$). Into every cell of the domain we put some number of particles which is proportional to the density. The velocities of the particles are equal to zero. The typical number of particles in calculations was $M = 5 \cdot 10^3 \div 10^4$.

At the next time step τ the following calculations are realized.

1. The Lagrangian step. The equations of ion motion are solved according to explicit schemes:

$$\begin{aligned} u_j^{m+1} &= u_j^m + \tau[(E_x)_j^m + v_j^m(H)_j^m], \\ v_j^{m+1} &= v_j^m + \tau \left[\left(E_y + \frac{\kappa}{n} \frac{\partial H}{\partial x} \right)_j^m + u_j^m(H)_j^m \right], \\ x_j^{m+1} &= x_j^m + \tau u_j^{m+1}, \quad j = 1, 2, \dots, J. \end{aligned}$$

The fields E_x , E_y , H are interpolated linearly from the mesh nodes into the particle locations by formulas of the PIC-model (see Chapter 1).

Further with use of particle coordinates and velocities the plasma density n_k and the macroscopic ion velocity $\langle u_i \rangle_k = u_k$ are calculated in the mesh nodes by formulas

$$\begin{aligned} n_k^{m+1} &= \sum_{j=1}^M \bar{R}(x_j^{m+1} - x_k), \\ u_k^{m+1} &= \frac{1}{n_k^{m+1}} \sum_{j=1}^M u_j^{m+1} \bar{R}(x_j^{m+1} - x_k), \end{aligned}$$

Here $\bar{R}(x)$ is the mesh kernel of the PIC-model (1.32).

2. The Euler step. The Maxwell equations and also the equations for electron component of plasma are solved.

In our case it is convenient to introduce the auxiliary function

$$Q = H - \beta \frac{\partial}{\partial x} \left(\frac{1}{n} \frac{\partial H}{\partial x} \right),$$

with the use of which the equation for the magnetic field is transformed into the more simple equation

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(uQ) = \frac{\kappa}{\beta}(H - Q).$$

In the finite-difference approximation the equations for temperature and function Q are written by the explicit schemes:

$$\begin{aligned} Q_k^{m+1} &= Q_k^m - \frac{\tau}{h} \left(u_k^{m+1} \Delta Q_k^m + \frac{u_{k+1}^{m+1} - u_{k-1}^{m+1}}{2} Q_k^m \right) + \\ &\quad + \frac{\kappa \tau}{4\beta} (H_{k-1}^m + 2H_k^m + H_{k+1}^m - Q_{k-1}^m - 2Q_k^m - Q_{k+1}^m), \\ T_k^{m+1} &= T_k^m - \frac{\tau}{h} \left(u_k^{m+1} \Delta T_k^m + \frac{u_{k+1}^{m+1} - u_{k-1}^{m+1}}{3} T_k^m \right) + \frac{\kappa \tau}{3h^2} \left(\frac{H_{k+1}^m - H_{k-1}^m}{n_k^{m+1}} \right)^2 + \\ &\quad + \frac{\chi \tau}{3h^2 n_k^{m+1}} [(n_{k+1}^{m+1} + n_k^{m+1})(T_{k+1}^m - T_k^m) - (n_k^{m+1} + n_{k-1}^{m+1})(T_k^m - T_{k-1}^m)]. \end{aligned}$$

Here the velocity sign is taken into account. After this we find the magnetic field distribution using the values of function Q_k^{m+1} calculated in the mesh nodes. The system of three-point equations is solved by the sweep method:

$$\begin{aligned} \frac{H_{k+1}^{m+1}}{n_{k+1}^{m+1} + n_k^{m+1}} - \left(\frac{1}{n_{k+1}^{m+1} + n_k^{m+1}} + \frac{1}{n_k^{m+1} + n_{k-1}^{m+1}} + \frac{h^2}{2\beta} \right) H_k^{m+1} + \\ + \frac{H_{k-1}^{m+1}}{n_k^{m+1} + n_{k-1}^{m+1}} = - \frac{h^2}{2\beta} Q_k^{m+1}. \end{aligned}$$

At the end the components E_x and E_y of the electric field are calculated

$$E_x^{m+1} = \frac{n_{k-1}^{m+1} T_{k-1}^{m+1} + (H_{k-1}^{m+1})^2 - n_k^{m+1} T_k^{m+1} - (H_k^{m+1})^2}{h(n_k^{m+1} + n_{k-1}^{m+1})},$$

$$\begin{aligned} \left(E_y + \frac{\kappa \partial H}{n \partial x} \right)_k^{m+1} = u_k^{m+1} H_k^{m+1} - \frac{\beta}{\tau h} \left(\frac{H_{k+1}^{m+1} - H_{k-1}^{m+1}}{n_k^{m+1}} - \frac{H_{k+1}^m - H_{k-1}^m}{n_k^m} \right) - \\ - \frac{2\beta u_k^{m+1}}{h^2} \left(\frac{H_{k+1}^{m+1} - H_k^{m+1}}{n_{k+1}^{m+1} + n_k^{m+1}} - \frac{H_k^{m+1} - H_{k-1}^{m+1}}{n_k^{m+1} + n_{k-1}^{m+1}} \right). \end{aligned}$$

By this the cycle of calculations is completed. Then it is repeated again until the whole time interval which is interesting for us will be covered.

The calculations with use presented above hybrid model showed that at the initial density and magnetic field drop $A < 12$ the decay of the break leads to the formation of laminar shock wave with subcritical parameters going in the positive direction of x coordinate and rarefaction wave propagating in the opposite direction. The shock wave has the oscillatory front caused by the dispersion connected with the inertia of electrons. The presence of the finite conductivity determines the greater width of the magnetic field profile in comparison with the profiles of the longitudinal velocity and density of plasma particles. The breakdown of the front and intermixing of particles don't take place.

With the increase of the initial density drop character of the flow changes. At $12 < A < 25$ the shock wave during little time from the beginning of the breaking is a unsteady laminar wave with the oscillatory front. But after some time the amplitude of front oscillation on the profile of the longitudinal ion velocity (and electron velocity) increases up to the critical one, the breakdown of the wave takes place accompanied by the formation of fast particles reflected by the front, and also by appearing of the "precursor" on the profile of the magnetic field. After ejection of the particles from the front oscillation its amplitude decreases; then the amplitude of the next oscillation increases up to the critical one, and ejection of particles from it takes place, etc. The reduction of front oscillation amplitude and the formation of fast particles are accompanied by deceleration of the shock wave. Then the wave velocity increases, a new group of fast particles appears, etc. Therefore, the pulsation regime takes place.

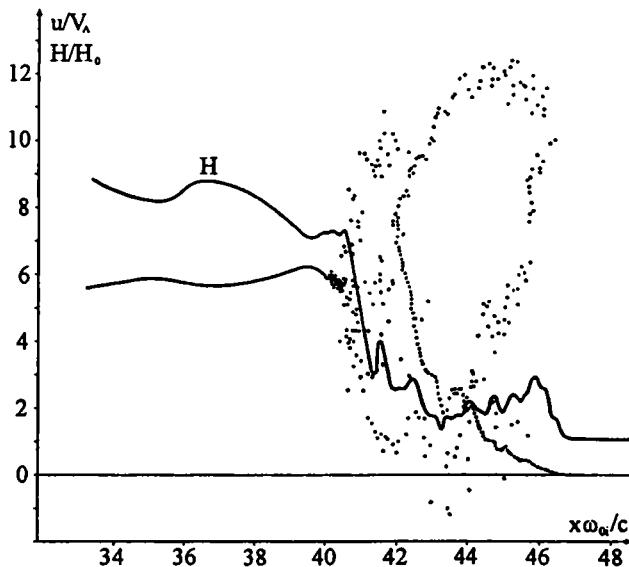


Fig. 35. The decay of the plasma density break in the magnetic field ($A = 30$, $t = 3\Omega_H^{-1}$).

At $A > 25$ the decay character of the discontinuity again changes. In this case, the shock wave amplitude always remains supercritical and ejection of particles is realized continuously. In this group of fast particles the ions from the unperturbed domain transit which are reflected by the wave front and also the ions which came with the wave. These two subgroups of ions receive two oppositely directed transverse velocities. On the magnetic field profile the clearly expressed "precursor" is formed which differs from the leading edge velocity by its amplitude and velocity. The space scale of this "precursor" is determined by the ion Larmor radius. In Fig. 35 the magnetic field profile and the positions and longitudinal velocities of ions are shown. In contrast to the strong shock wave in non-isothermal plasma without magnetic field when the "precursor" velocity equals the maximum velocity of ions, in the considered case with the magnetic field the "precursor" velocity is less than the maximum longitudinal velocity of ions. It is connected with the magnetic field influence which turns the particles. The particle distribution before the front of the shock wave on the plane x, u has the characteristic loop a form of which can't be described by the hydrodynamic equations.

In conclusion we note that described above algorithm without any difficulties can be generalized on a case of shock waves propagating at an arbitrary angle towards the magnetic field [86].

5.5.3. Numerical simulation of explosion phenomena in collisionless plasma at super-alfvenic speed

In this point we consider the application of particle method within the framework of the plasma hybrid model for calculation of collisionless interaction of plasma flows in the magnetic field. As was explained above in this model the electron component is described by the hydrodynamic equations and the ion component obeys the Vlasov kinetic equation.

It should be noted that the question about the nature and mechanisms of collisionless interaction of magnetized plasma flows, i.e. about the mechanisms of energy and momentum relaxation of plasma formations in the conditions, when the mean free path of plasma particles under binary Coulomb collisions considerably exceeds the characteristic scale of the system, it is one of the most fundamental in the plasma physics.

In the Universe there are numerous explosion phenomena characterized by huge energy and mass releases and accompanied by the formation of high-speed plasma structures, e.g., expanding spherical and circular shells, powerful jets, etc. As examples of such events we can mention: the Supernova explosions with mass ejection which can be compared with the mass of the Sun ($M_{\odot} = 2 \cdot 10^{33}$ g) and speeds $10^8 - 10^9$ cm/sec; Nova Stars explosions with mass $M \sim 10^{30}$ g; solar chromospheric flares with mass ejection $M \leq 10^{17}$ g and speeds $\sim 10^8$ cm/sec; the formation of planetary nebulae with mass $\sim 10^{32}$ g at explosions of Red Giants; the very intensive activity of the galactic object SS433 situated at a distance of 11 thousand light years and ejected two jets of a medium with speed $10^9 \div 10^{10}$ cm/sec in opposite directions.

Simular events on a essentially less energetic and space-time scales, of course, are also initiated by people's scientific and technical activity. These are the barium and lithium releases in the magnetotail of the Earth's magnetosphere under active experiments on the programs AMPTE and CRESS [84]; high-altitude nuclear explosions [87, 88]; an expansion of laser plasma clouds on inertial fusion; laboratory simulations of supernova remnants, expanding through interstellar matter; and solar wind around the Earth's magnetosphere [89, 90].

For the quantitative description of the deceleration process it is useful to introduce so called radii of the dense plasma cloud deceleration. The deceleration by a magnetic field is defined by the magnetic radius R_H , and coupling between the cloud and ambient plasmas is characterized by the gasdynamic radius \tilde{R} . Equating the cloud kinetic energy to the energy of the magnetic field pushed from the spherical volume of radius R_B yields

$$MV_0^2/2 \simeq (H_0^2/8\pi)(4\pi R_B^3/3)$$

$$R_H = (3MV_0^2/H_0^2)^{1/3}. \quad (57)$$

Here M is a mass of the cloud, V_0 is its initial velocity, H_0 is undisturbed magnetic field.

In the case of the interaction with the background medium a plasma cloud expands, scattering ambient matter and sweeping up it in a joint motion. During this cloud expansion, the mass of background ions seized by the moving cloud are increased. The sphere radius in which these parameters for both the cloud and the ambient plasma is equal is called the gasdynamic radius of deceleration

$$M = \frac{4}{3}\pi\tilde{R}^3n_*m_*, \tilde{R} = \left(\frac{3M}{4\pi n_*m_*}\right)^{1/3}. \quad (58)$$

Here n_* , m_* are the concentration and mass of background ions.

The specific deceleration mechanism, either magnetic or gasdynamical, depends on values of the ratio R_H/\tilde{R} . It's easy to obtain that $R_H/\tilde{R} = M_A^{2/3}$ where $M_A = \sqrt{4\pi n_*m_*}V_0/H_0^{1/2}$ is the Alfvén-Mach number of the expanding plasma cloud relatively to the background plasma medium.

From here it follows that at large Alfvén-Mach numbers ($M_A \gg 1$) a slowing down of the cloud is caused by cloud-background interaction while for sub-Alfvénic flows ($M_A \ll 1$) a cloud looses the energy on ousting and deformation of the magnetic field. Here we consider only the case $M_A \gg 1$ when the deceleration of the cloud is caused by the interaction with the background medium.

The first analysis of the collisionless deceleration of plasma clouds in a magnetized plasma at large M_A was represented in the papers [91, 92]. In these investigations the cloud ions were described by the Vlasov kinetic equation, and the electron fluid velocity \mathbf{V}_e which is necessary for obtaining the current density \mathbf{j} from the simplest form Ohm's law

$$\mathbf{E} + [\mathbf{V}_e \mathbf{H}] / c = 0.$$

This form means that the inertia and pressure of electrons are ignored.

Obtained in these studies analytical solutions for an initial phase of the plasmoid expansion through an ambient plasma made it possible to estimate the fraction of the energy transferred from cloud ions to background ions. This energy fraction is proportional to the interaction parameter $\delta = (\tilde{R}/R_i)^2$ [91, 92]. Here $R_i \equiv \frac{V_0 m_* c}{Z_* e H_0}$ is a Larmor radius of background plasma ions.

The numerical simulation of the process of collisionless deceleration of a cloud in the background medium on the basis of the hybrid plasma models was carried out in the series of papers [91, 92, 93, 94]. Here we consider the statement of the problem, and the algorithm of the numerical realization of the two-dimensional hybrid model in the cylindrical

(r, z) geometry, and some results of the calculation of the collisionless interactions of plasma flows.

The problem of the interaction of collisionless plasma flows in the general form is formulated in the following way. At the initial time moment $t = 0$ the point explosion forms the dense plasma cloud containing N particles with the kinetic energy W_0 . The surrounding space is filled by ambient plasma of density n_* , imbedded into the magnetic field \mathbf{B}_0 . Expansion of the cloud can take place as in the uniform magnetized background with $n_* = \text{const}$, as well in the presence of the density gradient of background plasma directed along the magnetic field: $\nabla n_* \parallel \mathbf{H}_0$.

Below we give equations of the two-dimensional axial-symmetric hybrid model. In this case, $\partial/\partial\varphi = 0$, $\mathbf{U} = \{U_r, U_\varphi, U_z\}$, $\mathbf{v} = \{v_r, v_\varphi, v_z\}$, $\mathbf{H} = \{H_r, H_\varphi, H_z\}$, $\mathbf{E} = \{E_r, E_\varphi, E_z\}$. Because of the fact that smallest scale in this problem is the ion dispersion length $\delta_i = \frac{c}{\omega_{0i}}$, further we assume $\beta \equiv \frac{m_e}{m_i} = 0$. Then the system of equations of the hybrid model in dimensionless variables has the following form:

$$\frac{dr}{dt} = v_r, \quad (59)$$

$$\frac{dz}{dt} = v_z, \quad (60)$$

$$\frac{dv_r}{dt} = \frac{v_\varphi^2}{r} + E_r + v_\varphi H_z - v_z H_\varphi + \frac{\nu}{n} \frac{\partial H_\varphi}{\partial z}, \quad (61)$$

$$\frac{dv_\varphi}{dt} = -\frac{v_\varphi v_r}{r} + E_\varphi + v_z H_r - v_r H_z - \frac{\nu}{n} \left(\frac{\partial H_r}{\partial z} - \frac{\partial H_z}{\partial r} \right), \quad (62)$$

$$\frac{dv_z}{dt} = E_z + v_r H_\varphi - v_\varphi H_r - \frac{\nu}{n r} \frac{1}{r} \frac{\partial}{\partial r} (r H_\varphi), \quad (63)$$

$$E_r = V_z H_\varphi - V_\varphi H_z - \frac{1}{2n} \frac{\partial p}{\partial r} - \frac{\nu}{n} \frac{\partial H_\varphi}{\partial r}, \quad (64)$$

$$E_\varphi = V_r H_z - V_z H_r + \frac{\nu}{n} \left(\frac{\partial H_r}{\partial z} - \frac{\partial H_z}{\partial r} \right), \quad (65)$$

$$E_z = V_\varphi H_r - V_r H_\varphi - \frac{1}{2n} \frac{\partial p}{\partial z} + \frac{\nu}{n r} \frac{1}{r} \frac{\partial}{\partial r} (r H_\varphi), \quad (66)$$

$$\begin{aligned} n \left(\frac{\partial T}{\partial t} + V_r \frac{\partial T}{\partial r} + V_z \frac{\partial T}{\partial z} \right) &= -(\gamma - 1) n T \left\{ \frac{1}{r} \frac{\partial}{\partial r} (r V_r) + \frac{\partial V_z}{\partial z} \right\} + \\ &(\gamma - 1) \frac{2\nu}{n} \left[\left(\frac{\partial H_\varphi}{\partial z} \right)^2 + \left(\left(\frac{\partial H_r}{\partial z} \right)^2 - \left(\frac{\partial H_z}{\partial r} \right)^2 + \frac{1}{r^2} \left(\frac{\partial}{\partial r} (r H_\varphi)^2 \right) \right) \right] + \\ &(\gamma - 1) \left[\frac{1}{r} \frac{\partial}{\partial r} \left(\chi r \frac{\partial T}{\partial r} \right) + \frac{\partial}{\partial z} \left(\chi \frac{\partial T}{\partial z} \right) \right], \end{aligned} \quad (67)$$

$$V_r = U_r + \frac{1}{n} \frac{\partial H_\varphi}{\partial z}, \quad (68)$$

$$V_\varphi = U_\varphi - \frac{1}{n} \left(\frac{\partial H_r}{\partial z} - \frac{\partial H_z}{\partial r} \right), \quad (69)$$

$$V_z = U_z - \frac{1}{rn} \frac{\partial}{\partial r} (r H_\varphi), \quad (70)$$

$$\frac{\partial H_r}{\partial t} = \frac{\partial E_\varphi}{\partial z}, \quad (71)$$

$$\frac{\partial H_\varphi}{\partial t} = \frac{\partial E_z}{\partial r} - \frac{\partial E_r}{\partial z}, \quad (72)$$

$$\frac{\partial H_z}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (r E_\varphi). \quad (73)$$

In the system of the Eqs. (59)-(73) \mathbf{v} and \mathbf{U} are an individual and macroscopic ion velocities, ν is effective collision frequency of ions with electrons, \mathbf{V} is an average velocity of the electron fluid; \mathbf{E}, \mathbf{H} are the electric and magnetic fields; T is an electron temperature, χ is the electron heat conductivity. The pressure is connected with the electron temperature by the state equation of the ideal gas $p_e = nT$. As characteristic scales the following values are used: the magnetic field H_0 , the density n_0 , the velocity $V_A = H_0/\sqrt{4\pi m_i n_0}$, the temperature $H_0^2/8\pi$, the time $c/V_A \omega_{0i}$, the length c/ω_{0i} . Here $\omega_{0i} = (4\pi e^2 n_0/m_i)^{1/2}$, where m_i is the mass of hydrogen ions.

In accordance with the considered problem the initial data have the form:

$$\begin{aligned} H_r(r, z, 0) &= H_\varphi(r, z, 0) = 0, \\ H_z(r, z, 0) &= H_0 = \text{const}, \\ T(r, z, 0) &= T_0 = \text{const}, \\ n_*(r, z, 0) &= n_0 = \text{const}, \end{aligned} \quad (74)$$

$$V_r(r, z, 0) = V_\varphi(r, z, 0) = V_z(r, z, 0) = 0,$$

for background ions:

$$v_r(r, z, 0) = v_\varphi(r, z, 0) = v_z(r, z, 0) = 0, \quad (75a)$$

for the cloud ions:

$$\begin{aligned} v_r(r, z, 0) &= V_0 r / R_0, \\ v_\varphi(r, z, 0) &= 0, (\sqrt{r^2 + z^2} \leq R_0), \\ v_z(r, z, 0) &= V_0 z / R_0, \end{aligned} \quad (75b)$$

V_0 is the maximum value of the ion cloud velocity.

In the case of inhomogeneous background the initial distribution of the background plasma is defined as follows:

$$n_*(z) = \begin{cases} n_1, & -L_z \leq z \leq \frac{1}{2}\Delta, \\ n_1 + (n_2 - n_1)(z/\Delta + 1/2), & -\frac{1}{2}\Delta < z \leq \frac{1}{2}\Delta, \\ n_2, & \frac{1}{2}\Delta < z \leq L_z. \end{cases} \quad (76)$$

Here Δ is the spatial scale for the variation of background.

The solution is considered in the domain $0 \leq r \leq L_r$, $-L_z \leq z \leq L_z$. The boundary conditions have the following form:

$$\begin{aligned} r = 0 : \quad & \frac{\partial f(0, z, t)}{\partial r} = 0; \quad f = \{H_z, E_z, V_z, v_z, n, T\}; \\ & v_r(0, z, t) = V_r(0, z, t) = 0; \\ & v_\varphi(0, z, t) = V_\varphi(0, z, t) = 0; \\ & E_\varphi(0, z, t) = E_r(0, z, t) = 0; \\ & H_\varphi(0, z, t) = H_r(0, z, t) = 0; \\ r = L_r : \quad & f(L_r, z, t) = f(L_r, z, 0), \\ f = \{ & H_r, H_\varphi, H_z, E_r, E_\varphi, E_z, V_r, V_\varphi, V_z, v_r, v_\varphi, v_z, n, T \}; \\ z = \pm L_z : \quad & f(r, \pm L_z, t) = f(r, \pm L_z, 0), \\ f = \{ & H_r, H_\varphi, H_z, E_r, E_\varphi, E_z, V_r, V_\varphi, V_z, v_r, v_\varphi, v_z, n, T \}. \end{aligned} \quad (77)$$

The setting of unperturbed (initial) values of all variables at the exterior boundaries L_r , $\pm L_z$ allows to continue the computation only up to the moment when the disturbance comes on boundary.

The realization of the method.

In the computational domain the uniform mesh is introduced with steps $h_1 = L_r/I$, $h_2 = 2L_z/K$ where I, K are the numbers of cells along the axes r and z correspondingly. Mesh functions V_r, U_r, E_r, H_r are determined in the mesh nodes $r_i = ih_1$, $z_k = kh_2$, functions $V_\varphi, U_\varphi, E_\varphi, H_\varphi$ are determined in the points $r_i, z_{k-1/2} = (k - 0.5)h_2$, and functions n, T, V_z, U_z, E_z, H_z are defined in the centers of the cells $r_{i-1/2} = (i - 0.5)h_1, z_{k-1/2}$.

Let us consider separate steps of the calculation.

1. The Lagrangian step. The equations of ion motion are solved with the use of the scheme [79]. In this case, a solution is obtained in the Cartesian coordinates with the further recalculation of the particle coordinates and velocities into cylindrical coordinates:

$$\begin{aligned} \tilde{v}_r &= v_r^m + \tau(E_r^m + v_\varphi^m H_z^m - v_z^m H_\varphi^m), \\ \tilde{v}_\varphi &= v_\varphi^m + \tau(E_\varphi^m + v_z^m H_r^m - v_r^m H_z^m), \\ v_z^{m+1} &= v_z^m + \tau(E_z^m + v_r^m H_\varphi^m - v_\varphi^m H_r^m), \end{aligned}$$

$$\begin{aligned}
z^{m+1} &= z^m + \tau v_z^{m+1}, \\
x &= r^m + \tau \tilde{v}_r, \\
y &= \tau \tilde{v}_\varphi, \\
r^{m+1} &= \sqrt{x^2 + y^2}, \\
\sin \alpha &= y/r^{m+1}, \\
\cos \alpha &= x/r^{m+1}, \\
v_z^{m+1} &= \tilde{v}_r \cos \alpha + \tilde{v}_\varphi \sin \alpha, \\
v_\varphi^{m+1} &= \tilde{v}_\varphi \cos \alpha - \tilde{v}_r \sin \alpha.
\end{aligned}$$

Here v_r , v_φ , v_z , r , z are velocities and coordinates of individual particles, τ is a time step; E_r , E_φ , E_z , H_r , H_φ , H_z are components of the electric and magnetic fields interpolated from the mesh nodes into the particle locations.

Then we determine the mesh functions of the ion density n and mean ion velocities U_r , U_φ , U_z in the cells with the use of the interpolation procedure "particles-mesh" according to the PIC-model (1.32).

2. The Euler step. From the Maxwell equations the components of electron velocity are determined:

$$\begin{aligned}
V_{r,i,k}^{m+1} &= U_{r,i,k}^{m+1} + 4(H_{\varphi,i,k+1/2}^m - H_{\varphi,i,k-1/2}^m)/ \\
&/h_2(n_{i+1/2,k+1/2}^{m+1} + n_{i-1/2,k+1/2}^{m+1} + n_{i+1/2,k-1/2}^{m+1} + n_{i-1/2,k-1/2}^{m+1}), \\
V_{\varphi,i,k-1/2}^{m+1} &= U_{\varphi,i,k-1/2}^{m+1} + \\
&+ \frac{2}{n_{i+1/2,k-1/2}^{m+1} + n_{i-1/2,k-1/2}^{m+1}} [(H_{r,i,k}^m - H_{r,i,k-1}^m)/h_2 - \\
&- (H_{z,i+1/2,k-1/2}^m - H_{z,i-1/2,k-1/2}^m)/h_1], \\
V_{z,i-1/2,k-1/2}^{m+1} &= U_{z,i-1/2,k-1/2}^{m+1} - \frac{r_i H_{\varphi,i,k-1/2}^m - r_{i-1} H_{\varphi,i-1,k-1/2}^m}{r_{i-1}/n_{i-1/2,k-1/2}^{m+1} h_1}.
\end{aligned}$$

Using the calculated electron velocities, we find electric fields from equations (64)-(66).

Then we calculate magnetic fields. We write down the equations for the magnetic fields in the form:

$$\begin{aligned}
\frac{\partial H_r}{\partial t} &= \frac{\partial E'_\varphi}{\partial z} + \frac{\partial}{\partial z} \left(\frac{\nu}{n} \frac{\partial H_r}{\partial z} \right), \\
\frac{\partial H_z}{\partial t} &= -\frac{1}{r} \frac{\partial}{\partial r} (r E''_\varphi) + \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{\nu r}{n} \frac{\partial H_z}{\partial r} \right), \\
\frac{\partial H_\varphi}{\partial t} &= \frac{\partial E'_z}{\partial r} - \frac{\partial E'_r}{\partial z} + \frac{\partial}{\partial r} \left(\frac{\nu}{nr} \frac{\partial}{\partial r} (r H_\varphi) \right) + \frac{\partial}{\partial z} \left(\frac{\nu}{n} \frac{\partial H_\varphi}{\partial z} \right).
\end{aligned}$$

In these equations

$$\begin{aligned} E_{\varphi}' &= V_r H_z - V_z H_r - \frac{\nu}{n} \frac{\partial H_z}{\partial r}, \\ E_{\varphi}'' &= V_r H_z - V_z H_r + \frac{\nu}{n} \frac{\partial H_z}{\partial r}, \\ E_z' &= V_{\varphi} H_r - V_r H_{\varphi} - \frac{1}{2n} \frac{\partial p}{\partial z}, \\ E_r' &= V_z H_{\varphi} - V_{\varphi} H_z - \frac{1}{2n} \frac{\partial p}{\partial r}. \end{aligned}$$

The finite-difference schemes for the calculation of the magnetic fields have the following form:

$$\begin{aligned} \frac{H_{r,i,k}^{m+1} - H_{r,i,k}^m}{\tau} &= \frac{1}{h_2} \left(\frac{\nu_{i,k+1/2}^m}{n_{i,k+1/2}^m} \frac{H_{r,i,k+1}^{m+1} - H_{r,i,k}^{m+1}}{h_2} - \frac{\nu_{i,k-1/2}^m}{n_{i,k-1/2}^m} \frac{H_{r,i,k}^{m+1} - H_{r,i,k-1}^{m+1}}{h_2} \right) \\ &\quad + \left(\frac{\partial E_{\varphi}'}{\partial z} \right)_{i,k}^m, \\ \frac{H_{z,i-1/2,k-1/2}^{m+1} - H_{z,i-1/2,k-1/2}^m}{\tau} &= \\ \frac{1}{h_1^2(i-1/2)} \left[i \frac{\nu_{i+1/2,k-1/2}^m + \nu_{i-1/2,k-1/2}^m}{n_{i+1/2,k-1/2}^m + n_{i-1/2,k-1/2}^m} (H_{z,i+1/2,k-1/2}^{m+1} - H_{z,i-1/2,k-1/2}^{m+1}) - \right. \\ &\quad \left. (i-1) \frac{\nu_{i-1/2,k-1/2}^m + \nu_{i-3/2,k-1/2}^m}{n_{i-1/2,k-1/2}^m + n_{i-3/2,k-1/2}^m} (H_{z,i-1/2,k-1/2}^{m+1} - H_{z,i-3/2,k-1/2}^{m+1}) \right] - \\ &\quad \frac{1}{(i-1/2)h_1} [i E_{\varphi,i,k-1/2}''^m - (i-1) E_{\varphi,i-1,k-1/2}''^m], \\ \frac{H_{\varphi,i,k-1/2}^{m+1/2} - H_{\varphi,i,k-1/2}^m}{\tau} &= \\ \frac{1}{h_1^2} \left[\frac{\nu_{i+1/2,k-1/2}^m}{r_{i+1/2} n_{i+1/2,k-1/2}^m} (r_{i+1} H_{\varphi,i+1,k-1/2}^{m+1/2} - r_i H_{\varphi,i,k-1/2}^{m+1/2}) - \right. \\ &\quad \left. \frac{\nu_{i-1/2,k-1/2}^m}{r_{i-1/2} n_{i-1/2,k-1/2}^m} (r_i H_{\varphi,i,k-1/2}^{m+1/2} - r_{i-1} H_{\varphi,i-1,k-1/2}^{m+1/2}) \right], \\ \frac{H_{\varphi,i,k-1/2}^{m+1} - H_{\varphi,i,k-1/2}^{m+1/2}}{\tau} &= \\ \frac{1}{h_2^2} \left[\frac{\nu_{i,k}^m}{n_{i,k}^m} (H_{\varphi,i,k+1/2}^{m+1} - H_{\varphi,i,k-1/2}^{m+1}) - \frac{\nu_{i,k-1}^m}{n_{i,k-1}^m} (H_{\varphi,i,k-1/2}^{m+1} - H_{\varphi,i,k-3/2}^{m+1}) \right] &+ \\ &\quad \left(\frac{\partial E_z'}{\partial r} - \frac{\partial E_r'}{\partial z} \right)_{i,k-1/2}^m. \end{aligned}$$

For the calculation of electron temperature we used the splitting scheme taking into account the velocity sign. To increase the computational stability each fractional step of this scheme is realized implicitly:

$$\begin{aligned}
& \frac{T_{i-1/2,k-1/2}^{m+1/2} - T_{i-1/2,k-1/2}^m}{\tau} - \frac{V_{r,i-1/2,k-1/2}^m}{h_1} \Delta_r T_{i-1/2,k-1/2}^{m+1/2} = \\
& \frac{\gamma - 1}{r_{i-1/2} h_1^2 n_{i-1/2,k-1/2}^{m+1}} \left[r_i \chi_{i,k-1/2}^m \left(T_{i+1/2,k-1/2}^{m+1/2} - T_{i-1/2,k-1/2}^{m+1/2} \right) - \right. \\
& \left. r_{i-1} \chi_{i-1,k-1/2}^m \left(T_{i+1/2,k-1/2}^{m+1/2} - T_{i-1/2,k-1/2}^{m+1/2} \right) \right] - (\gamma - 1) T_{i-1/2,k-1/2} \\
& \left[\frac{r_i V_{r,i,k-1/2}^m - r_{i-1} V_{r,i-1,k-1/2}^m}{h_1 r_{i-1/2}} + \frac{V_{z,i-1/2,k+1/2}^m - V_{z,i-1/2,k-3/2}^m}{2h_2} \right] \\
& + \frac{2(\gamma - 1) \nu_{i-1/2,k-1/2}^m}{n_{i-1/2,k-1/2}^m} \left[\left(\frac{H_{\varphi,i-1/2,k-1/2}^m - H_{\varphi,i-1/2,k-3/2}^m}{2h_2} \right)^2 + \right. \\
& \left(\frac{H_{r,i-1/2,k}^m - H_{r,i-1/2,k-1}^m}{h_2} - \frac{H_{z,i+1/2,k-1/2}^m - H_{z,i-3/2,k-1/2}^m}{2h_1} \right)^2 + \\
& \left. \frac{1}{r_{i-1/2}^2} \left(\frac{r_i H_{\varphi,i,k-1/2}^m - r_{i-1} H_{\varphi,i-1,k-1/2}^m}{h_1} \right)^2 \right], \\
& \frac{T_{i-1/2,k-1/2}^{m+1} - T_{i-1/2,k-1/2}^{m+1/2}}{\tau} - \frac{V_{z,i-1/2,k-1/2}^{m+1}}{h_2} \Delta_z T_{i-1/2,k-1/2}^{m+1} = \\
& \frac{\gamma - 1}{r_{i-1/2,k-1/2}^{m+1} h_2^2} \left[\chi_{i-1/2,k}^{m+1/2} \left(T_{i-1/2,k+1/2}^{m+1} - T_{i-1/2,k-1/2}^{m+1} \right) - \right. \\
& \left. \chi_{i-1/2,k-1}^{m+1} \left(T_{i-1/2,k-1/2}^{m+1} - T_{i-1/2,k-3/2}^{m+1} \right) \right].
\end{aligned}$$

Here

$$\begin{aligned}
\Delta_r T_{i-1/2,k-1/2} &= \begin{cases} \frac{1}{h_1} (T_{i-1/2,k-1/2} - T_{i-3/2,k-1/2}), & V_{r,i-1/2,k-1/2}^{m+1} \geq 0, \\ \frac{1}{h_1} (T_{i+1/2,k-1/2} - T_{i-1/2,k-1/2}), & V_{r,i-1/2,k-1/2}^{m+1} < 0, \end{cases} \\
\Delta_z T_{i-1/2,k-1/2} &= \begin{cases} \frac{1}{h_2} (T_{i-1/2,k-1/2} - T_{i-1/2,k-3/2}), & V_{z,i-1/2,k-1/2}^{m+1} \geq 0, \\ \frac{1}{h_2} (T_{i-1/2,k+1/2} - T_{i-1/2,k-1/2}), & V_{z,i-1/2,k-1/2}^{m+1} < 0. \end{cases}
\end{aligned}$$

Let us consider some results of calculations carried out on the basis of the hybrid model presented above. In Fig. 36 the time dependence of kinetic energy of the cloud is depicted. It can be seen that at $\delta > 100$ the energetic curves have a universal form, i.e., further increasing in δ doesn't significantly influence their shape. The calculations show that the cloud loses asymptotically on time approximately 80 percent of initial energy during time $t \simeq 2\bar{R}/V_0$.

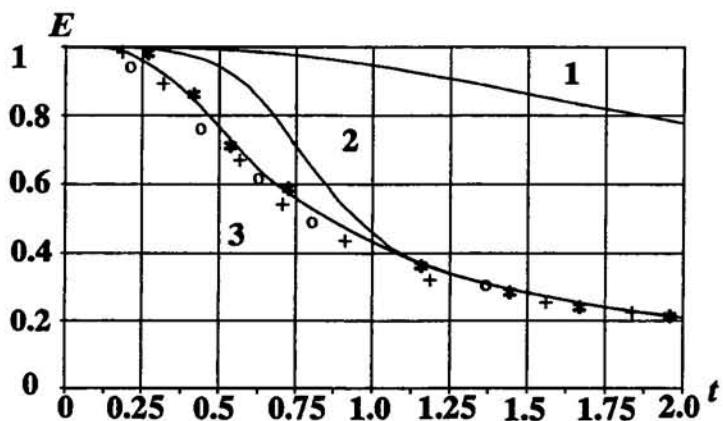


Fig. 36. The time dependence ($\tilde{t} = V_0 t / \tilde{R}$) of the cloud kinetic energy $E = W_1/W_0$ on the parameter δ at $M_A = 10$: 1 - $\delta = 1$; 2 - $\delta = 10$; 3 - $\delta = 100$; * - $\delta = 200$; + - $\delta = 300$; o - $\delta = 400$.

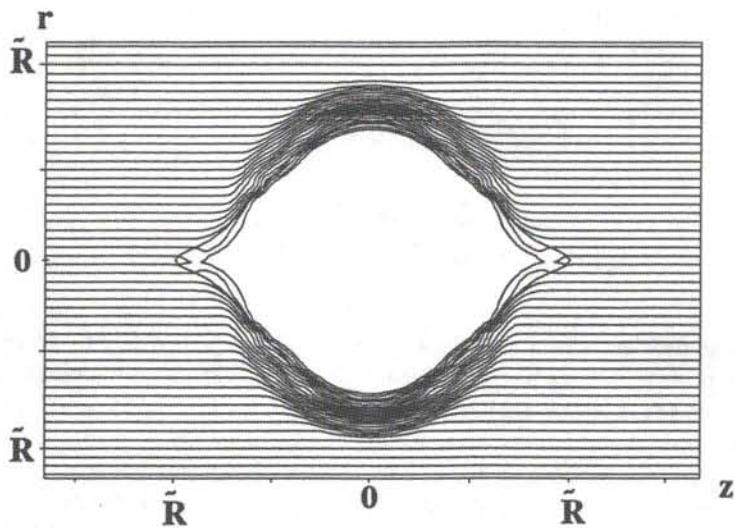


Fig. 37. Magnetic field lines at $M_A = 10$, $\delta = 100$, $t = \tilde{R}/V_0$.

As a result of the cloud expansion and acceleration of the ambient medium a plasma cavity, i.e., the domain with the reduced concentration of the background plasma, $n < n_*$ is formed. The density cavity correlates with the magnetic cavity with radius $R \sim \tilde{R}$ where the magnetic field value is less than undisturbed one, i.e., $H < H_0$. A typical

picture of the magnetic field lines for the case $M_A \gg 1, \delta \gg 1$ is represented in Fig. 37. Let us note that at $M_A \approx 10$ and $\delta \leq 4$ these results are in well agreement with the data of the corresponding numerical calculations and the approximate analytical theory developed in the papers [91, 92].

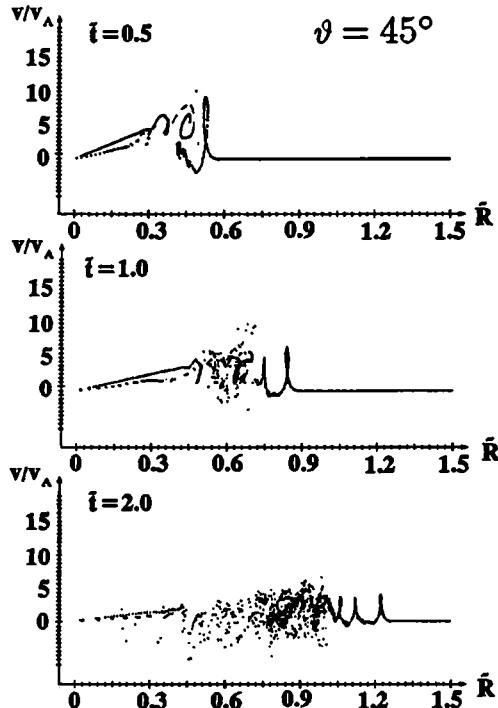


Fig. 38. Phase planes at $M_A = 10$, $\delta = 100$, $\vartheta = 45^\circ$.

The deceleration of the plasma cloud is accompanied by the generation of disturbances in the ambient plasma. In Fig. 38 the phase trajectories of cloud and background particles for polar angle $\vartheta = 45^\circ$ are represented at the successive time moments $t \approx 0.5\bar{R}/V_0$, $t \approx \bar{R}/V_0$, $t \approx 2\bar{R}/V_0$ at $\delta \approx 10^2$, $M_A = 10$. It can be seen the successive generation of disturbances in the background plasma connected with the rotation of the cloud ions. These disturbances are transformed into supercritical collisionless shock wave propagating in the background plasma.

5.5.4. Interaction of the laser impulse with plasma

As the example of computations with using of particle method in completely kinetic code the numerical simulation of the interaction between the powerful laser impulse and plasma is considered. The numerical complexity of the problem is related with the presence of some

essentially different time and space scales. In the space it is a ratio of the laser impulse wave and the space size of the problem which reaches several tens. In the time there are essentially different velocities of the ions and electrons. Besides, some real processes need the three-dimensional simulation. Therefore only recently with the advent of multiprocessor computers it became possible to solve the problems of such a complexity.

In this problem the ion and the electron components are described both by the Vlasov kinetic equations. The need of kinetic consideration of both components is caused by the substantially non-equilibrium of the ion and electron velocity distribution functions. In this case the evolution of both species are to calculate by particle method. Up to now in such a statement with the help of two-dimensional and three-dimensional codes on the basis of the PIC-model the problems of acceleration of charged particles, relativistic self-focusing of the laser impulse, formation of the quasi-static magnetic fields and others were solved in papers [94-108].

The problem is formulated in the following way. The computational domain in the form of a rectangular parallelepiped $\{0 < x < L_x, 0 < y < L_y, 0 < z < L_z\}$ is filled with stationary plasma consisting of ions and electrons. The foil target is simulated by a thin plasma layer. At the initial time moment the electric and magnetic fields within the domain are equal to zero. As a model of the laser impulse the electric and magnetic fields are excited at the left boundary of the domain for $x = 0$. Then such impulse propagates into the domain interacting with the ions and electrons.

The system of equations consists of kinetic equations for the ion and electron components:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \frac{\partial f_\alpha}{\partial \mathbf{r}} + \mathbf{F}_\alpha \cdot \frac{\partial f_\alpha}{\partial \mathbf{p}} = 0,$$

$$\mathbf{F}_\alpha = q_\alpha \left(\mathbf{E} + \frac{1}{c} [\mathbf{v}, \mathbf{B}] \right),$$

and the Maxwell equations

$$\frac{\partial \mathbf{B}}{\partial t} = -\eta \mathbf{E}, \quad (78)$$

$$\frac{\partial \mathbf{E}}{\partial t} = -\eta \mathbf{B} - 4\pi \mathbf{j}, \quad (79)$$

$$\operatorname{div} \mathbf{B} = 0, \quad (80)$$

$$\operatorname{div} \mathbf{E} = 4\pi \rho. \quad (81)$$

Here by index $\alpha = i, e$ ions and electrons are designated, $f_\alpha(\mathbf{r}, \mathbf{v}, t)$ is the distribution function of α -species, \mathbf{p}_α are the relativistic momentum of the α -particle,

$$\mathbf{p}_\alpha = m_\alpha \gamma_\alpha \mathbf{v}_\alpha, \quad \gamma_\alpha = (1 - v_\alpha^2/c^2)^{-1/2},$$

m_α is the rest-mass of the ion or electron, q_α is the ion or electron charge with value $q_\alpha = \pm e$. The current density \mathbf{j} and the charge density ρ are expressed in the form

$$\begin{aligned} \mathbf{j} &= \sum_\alpha q_\alpha \int f_\alpha \mathbf{v} d\mathbf{v}, \\ \rho &= \sum_\alpha q_\alpha \int f_\alpha d\mathbf{v}. \end{aligned}$$

From the kinetic equations as was made above for one-component problem one can obtain the relativistic equations of the motion for model ions and electrons:

$$\begin{aligned} \frac{d\mathbf{p}_\alpha}{dt} &= q_\alpha (\mathbf{E} + \frac{1}{c} [\mathbf{v}_\alpha \mathbf{B}]), \\ \frac{d\mathbf{r}_\alpha}{dt} &= \mathbf{v}_\alpha, \end{aligned} \tag{82}$$

Let us introduce dimensionless variables using as characteristic scales the following values: the wave length of the laser impulse λ ; the period of the laser impulse wave $t_0 = 2\pi/\omega_0$; the light speed c ; the electric field $E_0 = m_e c \omega_0 / 2\pi e$; the plasma density $n_0 = m_e \omega_0^2 / 16\pi^3 e^2$. Here ω_0 is a frequency of the electromagnetic impulse. Equations (82) in the dimensionless variables have the following form:

$$\begin{aligned} \frac{d\mathbf{p}_e}{dt} &= -(\mathbf{E} + [\mathbf{v}_e, \mathbf{B}]), && \text{(for electrons)} \\ \frac{d\mathbf{p}_i}{dt} &= \kappa(\mathbf{E} + [\mathbf{v}_i, \mathbf{B}]), && \text{(for ions)} \\ \frac{d\mathbf{r}_\alpha}{dt} &= \mathbf{v}_\alpha, \end{aligned}$$

where $\kappa = m_e/m_i$ is the electron-ion mass ratio, $\mathbf{p}_\alpha = \gamma_\alpha \mathbf{v}_\alpha$, $\gamma_\alpha^{-1} = \sqrt{1 - v_\alpha^2}$.

The Maxwell equations are transformed into the form:

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial t} &= \text{rot} \mathbf{B} - \mathbf{j}, \\ \frac{\partial \mathbf{B}}{\partial t} &= -\text{rot} \mathbf{E}, \\ \text{div} \mathbf{E} &= \rho, \\ \text{div} \mathbf{B} &= 0. \end{aligned}$$

The charge density ρ and the current density \mathbf{j} satisfy the continuity equation

$$\frac{\partial \rho}{\partial t} + \operatorname{div} \mathbf{j} = 0.$$

At the initial time moment $t = 0$ within the domain the electric and magnetic fields are equal to zero.

$$\mathbf{E}(x, y, z) = \mathbf{B}(x, y, z) = 0.$$

The plasma within the domain is uniform and quasi-neutral initially. It means that at $t = 0$ ion and electron densities are equal and constant:

$$n_i(x, y, z) = n_e(x, y, z) = \text{const.}$$

The initial density is set in such a way that at the initial moment the equality

$$\frac{\omega_{pe}}{\omega_0} = a,$$

was satisfied, where a is a given value, ω_{pe} is an electron plasma frequency.

The boundary conditions are set in the following way. At the boundary $x = 0$ the electric and magnetic fields are set as the functions of time and coordinates (x, y) . Depending on the choice of these functions the localized electromagnetic impulse can be set in different ways. At the opposite boundary $x = L_x$ the conditions are set that allow the electromagnetic waves to leave the computational domain freely.

At the side boundaries $y = 0, L_y$ and $z = 0, L_z$, the boundary conditions are set in such a way that they can't influence the processes within the computational domain. We used the periodic boundary conditions.

In order to obtain the electric and magnetic fields the scheme suggested by Langdon and Lazinski in 1976 [110] is usually used. In this scheme the fields are determined from the difference analogs of the Faraday law (78) and Ampere law (79):

$$\frac{\mathbf{B}^{m+1/2} - \mathbf{B}^{m-1/2}}{\tau} = -\operatorname{rot}_h \mathbf{E}^m, \quad (83)$$

$$\frac{\mathbf{E}^{m+1} - \mathbf{E}^m}{\tau} = -\mathbf{j}^{m+1/2} + \operatorname{rot}_h \mathbf{B}^{m+1/2}. \quad (84)$$

In this scheme the operators rot_h and div_h are the difference analogs of differential operators rot and div . They are determined by the formulas:

$$\begin{aligned} \operatorname{rot}_h \mathbf{B} = & \left\{ \frac{B_{z,\alpha,\beta+1/2,\gamma} - B_{z,\alpha,\beta-1/2,\gamma}}{h_y} - \frac{B_{y,\alpha,\beta,\gamma+1/2} - B_{y,\alpha,\beta,\gamma-1/2}}{h_z}, \right. \\ & \frac{B_{x,\alpha,\beta,\gamma+1/2} - B_{x,\alpha,\beta,\gamma-1/2}}{h_z} - \frac{B_{z,\alpha+1/2,\beta,\gamma} - B_{z,\alpha-1/2,\beta,\gamma}}{h_x}, \\ & \left. \frac{B_{y,\alpha+1/2,\beta,\gamma} - B_{y,\alpha-1/2,\beta,\gamma}}{h_x} - \frac{B_{x,\alpha,\beta+1/2,\gamma} - B_{x,\alpha,\beta-1/2,\gamma}}{h_y} \right\}, \end{aligned}$$

$$\begin{aligned} \operatorname{div}_h \mathbf{B} = & \frac{B_{x,\alpha+1/2,\beta,\gamma} - B_{x,\alpha-1/2,\beta,\gamma}}{h_x} + \\ & + \frac{B_{y,\alpha,\beta+1/2,\gamma} - B_{y,\alpha,\beta-1/2,\gamma}}{h_y} + \frac{B_{z,\alpha,\beta,\gamma+1/2} - B_{z,\alpha,\beta,\gamma-1/2}}{h_z}. \end{aligned}$$

The components of the electric and magnetic fields are computed in the nodes of the meshes shifted relatively to each other on time and space. This scheme has the second order of approximation on space and time and what is very important the difference analogs of the Eq. (80) and Gauss laws (81) are satisfied here under absence of charges:

$$\begin{aligned} \operatorname{div}_h \mathbf{B}^{m+1/2} &= 0, \\ \operatorname{div}_h \mathbf{E}^m &= \rho. \end{aligned} \quad (85)$$

In the case when the charge density is not equal to zero, the satisfaction depends on the consistent calculations of current and charge densities. If they are calculated so that they satisfy the difference analog of the continuity equation (see Chapter 1)

$$\frac{\rho^{m+1} - \rho^m}{\tau} + \operatorname{div}_h \mathbf{j}^{m+1/2} = 0, \quad (86)$$

then the Gauss divergence law is also fulfilled. In the case when these equations are not satisfied the electric field is improved in such a way that difference analogs of both equations could be realized. This correction was suggested by Boris in 1970 [79]. It supposes to solve the Poisson equation on every time step, but this requires too much computational time. In 1987 Marder [111] suggested to introduce into the right side of the Eq. (7) the additional term which reduces a residual in the equation (85)

$$\frac{\mathbf{E}^{m+1} - \mathbf{E}^m}{\tau} = -\mathbf{j}^{m+1/2} + \operatorname{rot}_h \mathbf{B}^{m+1/2} + d \operatorname{grad} F,$$

where

$$F = \operatorname{div}_h \mathbf{E}^m - \rho^m.$$

This method is more economical. Here we don't need to solve the Poisson equation. But the defect of this method is in the fact that on every step the error in equation (85) is not fully eliminated, and this can gradually distort the modeled physical processes.

In 1992 Villasenor and Buneman [112] suggested the calculation method of current densities by the exact calculation of plasma fluxes across the cell boundaries. It allowed to satisfy automatically the difference continuity equation (86), and, consequently, to satisfy exactly the difference Gauss law. The similar method was used in our codes, it is described in detail in p.p. 1.3.5.

Thus, the computational scheme of the problem on one step is divided into two stages. At the first, Lagrangian step, according to the scheme

$$\frac{\mathbf{p}_\alpha^{m+1/2} - \mathbf{p}_\alpha^{m-1/2}}{\tau} = q_\alpha \left(\mathbf{E}^m + \left[\frac{\mathbf{v}_\alpha^{m+1/2} + \mathbf{v}_\alpha^{m-1/2}}{2}, \mathbf{B}^m \right] \right),$$

$$\frac{\mathbf{r}_\alpha^{m+1} - \mathbf{r}_\alpha^m}{\tau} = \mathbf{v}_\alpha^{m+1/2},$$

the velocities and coordinates of the particles are calculated. Besides, here we determine the components of the current density $\mathbf{j}^{m+1/2}$ by the formulas (89) of Chapter 1.

At the second, Euler step, we solve the Maxwell equations according to the scheme (83)-(84) and determine the functions $\mathbf{B}^{m+1/2}$, \mathbf{E}^{m+1} .

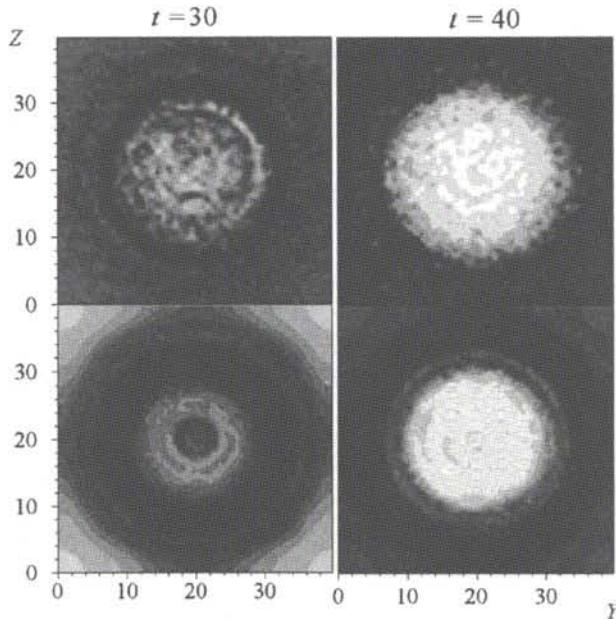


Fig. 39. The maximum density of electrons (upper) and ions (below) of the foil in the projection on the plane (y, z) at the time moments $t = 30$ and $t = 40$.

Let us consider the results of solving of the problem about the interaction of the short laser impulse with the thin foil. In Fig. 39 the interaction of the impulse with the foil is shown at $a = 32$. The other parameters are as such: length and width of the impulse are 25λ and 10λ correspondingly, the foil width is 0.375λ , the plasma density corresponds to $\omega_p/\omega = 1.1$. Here λ is a wave length of the laser impulse. From the Fig. 39 one can see that in the result of the interaction the steep leading edge of the laser impulse going through the foil is formed. The width of the leading edge is $\approx 3\lambda$. The formed impulse can now be used for the acceleration of particles up to relativistic velocities.

The calculations according to the three-dimensional program were performed at the mesh $320 \times 320 \times 320$. The total number of the model

particles was 4.915.200. The calculation up to $t = 40$ took approximately 50 min at 32 processors of CRAY T3E.

Chapter 6

Statistical particle-in-cell methods

6.1. Introduction

Statistical particle-in-cell methods are used in problems with the statement based on kinetic equations. These equations form a basis of the mathematical apparatus of the kinetic theory of gases, the kinetic theories of solid-state and dense gas plasma, and the kinetic theory of coagulating disperse systems. They are used in hydrodynamic turbulence theory, astrophysics, and other areas of fundamental sciences. A great number of independent variables and a complex structure of the kinetic equations make it very difficult to apply to them regular numerical methods based on finite-difference approximations, quadrature formulas, etc.

Statistical methods mostly developed in these areas and are based on the statistical character of kinetic equations, are a natural alternative to the above-listed methods. A discrete internal structure of the medium and individual character of the interaction of particles make it possible to construct imitation algorithms, which are, in essence, variants of particle methods.

In this chapter, statistical particle-in-cell methods are considered on the base of problems of rarefied gas dynamics (RGD). Here, they became as widely used as the particle methods in computational physics of rarefied plasma considered in Chapter 5.

The group of statistical particle-in-cell methods for RGD problems originated in papers by G. Bird described in his book [30]. In it, the main blocks of FORTRAN procedures that realize the scheme proposed by the author are presented. Besides, it provides in a compact form reference materials on the kinetic gas theory. Therefore, the book, up to now, can serve as a good guide for practical study of the method.

Among Russian specialists in computations, the greatest contribution to the justification and development of this method was made by O.M. Belotserkovsky and V.E. Yanitsky.

It should be noted that in the literature, especially in publications on RGD, these methods are often assigned to Monte Carlo methods, thereby reflecting the peculiarities of their realization. Many sources even use abbreviation DMCS (Direct Monte-Carlo Simulation). Therefore, it is important that all the above-mentioned authors successively stated that their schemes are variants of particle-in-cell methods. This reasonable methodological approach allowed from the first steps to bring out

a universal character of the methods, and then extend their application outside RGD. It is characteristic that the first advances in this extension were made just in the groups of Bird and Belotserkovsky-Yanitsky.

Thus, the description of these methods in a book devoted to particle-in-cell methods is fully justified.

Rarefied gas flows constitute a wide class of gas motions. It is assumed to classify rarefied flow regimes by Knudsen number [113]

$$Kn = \frac{\lambda}{L}.$$

Here λ is the mean free path of gas molecules, and L is the characteristic linear scale of the flow. As L one can use the middle size of a streamlined body, the diameter of a gas channel or a nozzle, the length of a sonic wave, or a space interval on which the typical difference of the macroparameters of a gas is formed.

The continual gas flow described by equations of gas dynamic (see Chapter 3), is the low limit of the regime scale at $Kn \rightarrow 0$. In this chapter, we consider flows at Knudsen numbers $Kn \simeq 10^{-1} \div 10^0$, where the gas behavior is substantially different than that in the continual state.

Such flows are realized at flows past spacecrafts in the upper atmosphere, in vacuum technological apparatuses of chemical and electronic industries, and in vacuum systems for scientific researches. Simultaneously, the flow past submicron aerosol particles and the propagation of ultrasound at atmospheric pressures also refer here.

The kinetic equations of the theory of gases are briefly considered in p. 2. Some elementary Monte Carlo procedures used in the algorithms under consideration are presented in p. 3 to elucidate the specific character of statistical simulation. The algorithms of statistical particle methods are described in p. 4. Both stages of the splitting-up problem are considered here on the basis of difference schemes, without the use of the Monte Carlo methods for integral equations [114], to emphasize their commonality with other particle-in-cell methods.

A brief sketch of applications of statistical particle-in-cell methods to the problems of gas dynamics, coagulation of disperse systems, and hydrodynamic turbulence is given in p. 5 at the end of chapter.

6.2. Kinetic equations of rarefied gas

6.2.1. Boltzmann equations

The Boltzmann kinetic equation for a single-species gas is written in the following form [113]:

$$\frac{Df}{Dt} = J(f, f) \quad (1)$$

Here $f(\mathbf{r}, \mathbf{v}, t)$ is the distribution function (DF) of gas molecules that depends on the space coordinates \mathbf{r} and molecular velocities \mathbf{v} . In the general case, these variables are determined in the direct product of spaces $R^3(\mathbf{r}) \times R^3(\mathbf{v}) \times R_+^1(t)$.

The mean macroscopic characteristics of the gas are expressed in terms of power moments DF taken over the velocity space $R^3(\mathbf{v})$. These quantities are functions of coordinates and time and are of local nature. In particular, the gas density is calculated as follows:

$$\rho(\mathbf{r}, t) = mn(\mathbf{r}, t) = m \int d\mathbf{v} f(\mathbf{r}, \mathbf{v}, t). \quad (2)$$

The macroscopic gas flow velocity is this:

$$\mathbf{U}(\mathbf{r}, t) = n^{-1} \int d\mathbf{v} \mathbf{v} f(\mathbf{r}, \mathbf{v}, t) \quad (3)$$

The gas absolute temperature at a point is given by the relation

$$T(\mathbf{r}, t) = \frac{m}{3kn} \int d\mathbf{v} (\mathbf{v} - \mathbf{U})^2 f(\mathbf{r}, \mathbf{v}, t) \quad (4)$$

Here m is the molecular mass, n is a number gas density, and k is the Boltzmann constant.

The statistical particle methods considered in this chapter are based on the concepts and terms of probability theory [34]. Therefore, to avoid misunderstanding, notice that the notation of function $f(\mathbf{r}, \mathbf{v}, t)$ by the term DF accepted in kinetic theory [115] does not coincide in meaning with the same term in probability theory. However, the function

$$F_1(\mathbf{r}, \mathbf{v}, t) = n^{-1}(\mathbf{r}, t) f(\mathbf{r}, \mathbf{v}, t) \quad (5)$$

is the probability density of gas molecule distribution in $R^3(\mathbf{v})$ at a point \mathbf{r} at a time t .

The transport operator in the left-hand side of Eq. (1) is determined as the derivative along the trajectory of a gas molecule in the six-dimensional space of coordinates and velocities

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{r}} + \frac{\mathbf{Q}}{m} \frac{\partial f}{\partial \mathbf{v}} \quad (6)$$

In contrast to the Vlasov equation (see Chapter 5), in the rarefied gas dynamics the force fields \mathbf{Q} usually depend neither on DF nor on molecular velocity. For the Boltzmann equation, there exist wide classes of problems, for instance, the RG aerodynamics problems, in which there is no field of external forces at all. All this makes it possible to assume that the operator (6) has a divergent form:

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \frac{\partial}{\partial \mathbf{r}} (\mathbf{v} f) + \frac{\partial}{\partial \mathbf{v}} \left(\frac{\mathbf{Q}}{m} f \right).$$

It is proposed in the derivation of the Boltzmann equation [113] that only binary interactions of particles considered as their collisions (scatterings) affect considerably the formation of DF. The contribution of the interactions is given by the collision operator (integral). If we restrict ourselves to a more illustrative molecular model of hard spheres, the Boltzmann collision integral can be represented in the following form:

$$\begin{aligned} J(f, f) = \frac{d_0^2}{2} \int d\mathbf{v}_1 \times \\ \times \int d\mathbf{n} |(\mathbf{g}, \mathbf{n})| (\mathbf{f}(\mathbf{v}')\mathbf{f}(\mathbf{v}'_1) - \mathbf{f}(\mathbf{v})\mathbf{f}(\mathbf{v}_1)). \end{aligned} \quad (7)$$

Here d_0 is a molecular diameter, \mathbf{n} is a unit direction vector, and $d\mathbf{n}$ is a differential surface element of a unit sphere.

The structure of Eq. (7) that is quadratic with respect f reflects the binary character of interaction. The velocity variables in $J(f, f)$ are related by the collision transformation as follows:

$$(\mathbf{v}', \mathbf{v}'_1) = S(\mathbf{v}, \mathbf{v}_1) \quad (8)$$

It is given by the following relations:

$$\begin{aligned} \mathbf{v}' = \frac{1}{2}(\mathbf{v} - \mathbf{v}_1) + \frac{1}{2}g\mathbf{n}, \quad \mathbf{v}'_1 = \frac{1}{2}(\mathbf{v} - \mathbf{v}_1) - \frac{1}{2}g\mathbf{n}, \\ g = |\mathbf{g}| = |\mathbf{v} - \mathbf{v}_1|. \end{aligned} \quad (9)$$

The vector \mathbf{n} lies in the line of centers at the moment of collision of particles. The momentum and energy conservation laws are satisfied for each pair of colliding molecules:

$$\mathbf{v} + \mathbf{v}_1 = \mathbf{v}' + \mathbf{v}'_1 \quad v^2 + v_1^2 = v'^2 + v'_1^2 \quad (10)$$

The process of particle collision is stochastic in character. The fact itself of the collision of a molecule with a velocity \mathbf{v} in the vicinity of the point \mathbf{r} of the physical space is a random event. Also, the velocity \mathbf{v}_1 of the molecule-partner in the collision is a random value. And, finally, the mutual location of the molecules at the moment of collision, which is given by the direction vector \mathbf{n} , is random too.

All these random acts are characterized by their own probabilities (density distributions). Their simulation by Monte Carlo methods forms the basis of the statistical particle methods.

It is helpful to write the integral of collisions Eq. (7) in the following structural form:

$$J(f, f) \equiv J_+(f, f) - \nu(f)f(\mathbf{r}, \mathbf{v}, t) \quad (11)$$

The terms in right side of Eq. (11) are called integrals of inverse and direct collisions correspondingly. They give a gain and loss of particles with the velocity \mathbf{v} in the vicinity of the point \mathbf{r} of the coordinate space. The function $\nu(f)$

$$\begin{aligned}\nu(f) &= \frac{d_0^2}{2} \int d\mathbf{v}_1 \int d\mathbf{n} |(\mathbf{v} - \mathbf{v}_1, \mathbf{n})| f(\mathbf{r}, \mathbf{v}, t) = \\ &= \pi d_0^2 \int d\mathbf{v}_1 |\mathbf{v} - \mathbf{v}_1| f(\mathbf{r}, \mathbf{v}, t)\end{aligned}\quad (12)$$

is the total collision number of a molecule with the velocity \mathbf{v} per unit time in a unit volume. The quantity

$$\sigma_t = \pi d_0^2 \quad (13)$$

is called the total collision cross-section for the model of hard spheres.

In the practice of calculations, the model collision integral [113]

$$J_M(f, f) = \nu(f_0 - f) \quad (14)$$

is often used instead of the Boltzmann collision operator (7) which is very complex. A comparison with Eqs. (11) and (12) shows that ν simulates the collision frequency, and the product νf_0 models the integral of inverse collisions. Usually ν is a function of the local moments (2) and (4), and does not depend on the relative velocity of colliding particles. Correspondingly,

$$f_0 = n \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} \exp \left(-\frac{m(\mathbf{v} - \mathbf{U})^2}{2kT} \right) \quad (15)$$

is a local Maxwell DF [113], whose parameters n , $\mathbf{U}(\mathbf{r}, t)$ and $T(\mathbf{r}, t)$ are determined by the relations (3) and (4) on the solution of f .

The kinetic equation with the model integral (14) is called a (model) kinetic equation of Bhatnagar-Gross-Krook (BGK) [113]. In spite of its simplicity, it turned out to be useful in many problems of the rarefied gas aerodynamics [26, 27, 116].

6.2.2. The splitting up scheme

The two physical processes are naturally distinguished in the Boltzmann kinetic equation (1). One of them is the translation of DF along the trajectories of free motion of molecules. It is described by the differential operator (6). The other process whose mathematical representation is given by the collision operator (7) is usually called the process of collisional relaxation [115]. The splitting up scheme for these processes on a time (iteration) step τ is represented in the form of the following two auxiliary problems:

$$\frac{D\varphi}{Dt} = 0, \quad \varphi(p\tau) = f(p\tau), \quad (16)$$

$$\frac{\partial \tilde{f}}{\partial t} = J(\tilde{f}, \tilde{f}), \quad \tilde{f}(p\tau) = \varphi((p+1)\tau). \quad (17)$$

Besides the initial data, the boundary conditions formulated in s.p. 2.4 are also added to the differential operator of the problem (16).

It is appropriate to remind here that splitting up schemes of the form (16), (17) were first introduced in [117] for the linear kinetic equation of transport theory.

Each of the processes defined above has its characteristic time. We can take

$$\tau_{tr} = \frac{L}{U},$$

as a time scale of free molecular transport (16). Here L is the characteristic hydrodynamic scale, for instance, the size of a middle cross section of a body streamlined by a gas. In some cases, this can be the grid spacing in the flow region. The characteristic relaxation time in Eq. (17) is expressed as a mean free time. It can be estimated in the following way [113]:

$$\tau_\lambda = \left(n \sigma_t \sqrt{\frac{2kT}{m}} \right)^{-1}. \quad (18)$$

Usually, the relation between the characteristic times

$$\frac{\tau_\lambda}{\tau_{tr}} \sim Kn, \quad (19)$$

is considered. Here Kn is the Knudsen number the value of which determines the regime of rarefied gas flow.

Statistical methods of particles are most widely used in a range $Kn \sim 10^{-1} \div 1$. It is evident that here the time step τ should be consistent with τ_λ . Therefore, as shown below, at the realization of the splitting up scheme (16), (17), the relation chosen is

$$\tau \ll \tau_\lambda.$$

Since the transport problem Eq. (16) is described by the first-order partial differential equation, its solutions $\varphi(\mathbf{r}, \mathbf{v}, t)$ are invariant along the characteristics of this equation (see p.p. 2.4).

The problem of spatially uniform relaxation Eq. (17) has five macroscopic conservation laws. Its integrals are the total number of particles n (density) (2), the mean gas velocity \mathbf{U} (3), and the temperature T (internal energy) (4). For the equation (17) with the model collision integral (14), this is established directly by the calculation of the corresponding moments of both sides of Eq. (17).

Actually, since the parameters f_0 in the form (15) are determined on the solution Eq. (17), the equalities

$$\int d\mathbf{v} f_0 = \int d\mathbf{v} f = n,$$

$$\int d\mathbf{v} \mathbf{v} f_0 = \int d\mathbf{v} \mathbf{v} f = n\mathbf{U},$$

$$\int d\mathbf{v}(\mathbf{v} - \mathbf{U})^2 f_0 = \int d\mathbf{v}(\mathbf{v} - \mathbf{U})^2 f = \frac{3kT}{m}.$$

are valid.

Hence, we see that these moments of the collision integral (14) are zero at any t . Thus, for problem Eq. (17)

$$\frac{\partial n}{\partial t} = \frac{\partial \mathbf{U}}{\partial t} = \frac{\partial T}{\partial t} = 0.$$

These conservation laws are satisfied also in the general case of the Boltzmann collision integral [113].

6.2.3. The master kinetic equation

In [28, 115], it was proposed to consider the Kac "master-equation" to construct algorithms of statistical modeling of the uniform relaxation stage, instead of problem (17). In this case, the problem of collisional relaxation is represented in the following form:

$$\begin{aligned} \frac{\partial F_N(\mathbf{V}, t)}{\partial t} &= \frac{n}{N} \frac{d_0^2}{4} \sum_{i < j}^N \int d\Omega_{ij} \times \\ &\times g_{ij}(F_N(\mathbf{V}'_{ij}, t) - F_N(\mathbf{V}, t)), \\ F_N(\mathbf{V}, t_0) &= F_N^0(\mathbf{V}). \end{aligned} \quad (20)$$

Equation (20), which we will call an M -equation, describes the evolution of a spatially uniform gas consisted of N similar molecules in a sequence of binary collisions. The integrals in the right-hand side are carried out on a unit sphere. As in the Boltzmann equation (1), (7), here we use for simplicity the molecular model of hard spheres. The function $F_N(\mathbf{V}, t)$ is the joint probability density of N molecule velocities given by the vector $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_N)$. As the density distribution, it satisfies the condition of normalization

$$\int d\mathbf{V} F_N(\mathbf{V}, t) = 1$$

The integration of $F_N(\mathbf{V}, t)$ with respect to the part of variables $\mathbf{v}_{s+1}, \dots, \mathbf{v}_N$ leads to partial (marginal [34]) densities:

$$F_s = \int d\mathbf{v}_{s+1} \dots d\mathbf{v}_N F_N(\mathbf{V}, t).$$

In particular, integration with respect to $N - 1$ variables gives the density F_1 introduced above (see (5)), where the dependence on \mathbf{r} is omitted).

The vector

$$\mathbf{V}'_{ij} = (\mathbf{v}_1, \dots, \mathbf{v}'_i, \dots, \mathbf{v}'_j, \dots, \mathbf{v}_N)$$

gives the set of post-collision velocities of N particles, in which the pair $\mathbf{v}'_i, \mathbf{v}_j$ is determined in terms of the collisional transformation (8):

$$(\mathbf{v}'_i, \mathbf{v}'_j) = S(\mathbf{v}_i, \mathbf{v}_j) \quad (21)$$

In this case, in relations (9), which give the result of the transformation, one should use the direction vector \mathbf{n}_{ij} and the modulus of relative velocity $g_{ij} = |\mathbf{v}_i - \mathbf{v}_j|$ that correspond to this collision. In each act of (21), the conservation laws (10) are satisfied.

To represent the relation between the two levels of description of uniform relaxation (see (17) and (20)), notice that if we integrate M -equation (20) with respect to the velocities $\mathbf{v}_2, \dots, \mathbf{v}_N$, after the transformations we obtain an equation of the following form:

$$\begin{aligned} \frac{\partial F_1(\mathbf{v}, t)}{\partial t} = & \frac{d_0^2}{4} n \frac{N-1}{N} \int \int d\mathbf{v}_1 d\Omega(\mathbf{g}, \mathbf{n}) \times \\ & \times (F_2(\mathbf{v}', \mathbf{v}'_1, t) - F_2(\mathbf{v}, \mathbf{v}_1, t)). \end{aligned} \quad (22)$$

An additional assumption of the statistical independence of interacting particles

$$F_2(\mathbf{v}, \mathbf{v}_1, t) = F_1(\mathbf{v}, t) F_1(\mathbf{v}_1, t)$$

and the limiting transition at $N \rightarrow \infty$ turns (22) into the spatially uniform Boltzmann equation of the form (17), but written for the probability density $F_1(\mathbf{v}, t)$. To some extent, this result explains the following asymptotic property of problem (20) as $N \rightarrow \infty$ [115, 118]. If at $t = t_0$ the initial distribution is multiplicative

$$F_N^0(\mathbf{V}) = \prod F_1^{(1)}(\mathbf{v}),$$

the solution $F_N(\mathbf{V}, t)$ remains such at all subsequent times, and the function

$$f(\mathbf{v}, t) = n F_1(\mathbf{v}, t)$$

is a solution of the spatially uniform Boltzmann equation (see (17)) with the initial function

$$f^0(\mathbf{v}) = n F_1^0(\mathbf{v}).$$

6.2.4. Initial and boundary conditions

Let us explain the formulation of boundary conditions for equation (1) by using the problem of a rarefied gas flow past a body. This is also reasonable because for these problems the statistical particle method has a clear priority [30, 28]. The calculation domain $G \subset R^3(\mathbf{r})$ with a boundary S incorporates the streamlined body, which is bounded by a surface Γ . We assume for simplicity that the external force field is absent. Then, DF is determined by using the velocity variable \mathbf{v} on all $R^3(\mathbf{v})$. It must satisfy the conditions of decrease as $|\mathbf{v}| \rightarrow \infty$ so that its first moments (2)-(4) be bounded.

The collisions operator does not affect the spatial variables DF . Therefore, the boundary conditions on the surfaces S and Γ are determined by the type of the differential transport operator (6). Its

characteristics going from the points of S form a family of straight lines as follows:

$$\mathbf{r}(t) = \mathbf{r}_s + \mathbf{v}t, \quad \mathbf{r}_s \in S. \quad (23)$$

Hence, to invert the transport operator at the internal points G adjacent to the surface S , we should specify DF on it for all directions into G :

$$f(\mathbf{r}_s, \mathbf{v}, t) = f_s(\mathbf{r}_s, \mathbf{v}, t), \quad \mathbf{v}n_s > 0, \quad (24)$$

where \mathbf{n}_s is outward unit normal of S . The characteristics (6) outgoing the surface are written similarly to (23). For an impermeable hard surface, however, the boundary condition that generalizes (24) is usually used. For DF of molecules moving along directions from the surface Γ , a relation with the DF of the molecules incident on Γ is given. The boundary condition is written in the form [113]:

$$\begin{aligned} |(\mathbf{v}, \mathbf{n}_\Gamma)|f(\mathbf{r}_\Gamma, \mathbf{v}, t) &= \int_{(\mathbf{v}_1, \mathbf{n}_\Gamma) < 0} d\mathbf{v}_1 |(\mathbf{v}_1, \mathbf{n}_\Gamma)| \times \\ &\times W(\mathbf{v}, \mathbf{v}_1) f(\mathbf{r}_\Gamma, \mathbf{v}_1, t), \\ |(\mathbf{v}, \mathbf{n}_\Gamma)| &> 0, \quad \mathbf{r}_\Gamma \in S \end{aligned} \quad (25)$$

The function $W(\mathbf{v}, \mathbf{v}_1)$ is determined by the model of the surface scattering. It has the meaning of conditional probability density of reflection with the velocity \mathbf{v} of a molecule incident on Γ with a velocity \mathbf{v}_1 . If the surface does not absorb molecules and does not have any sources, we have for \mathbf{v} and $\mathbf{v}_1 \in R^3(\mathbf{v})$

$$\int d\mathbf{v} W(\mathbf{v}, \mathbf{v}_1) = 1.$$

It should be emphasized that from the physical point of view the condition (25) is the balance of incident and reflected molecule fluxes in the vicinity of an arbitrary point $\mathbf{r}_\Gamma \in \Gamma$.

The model of mirror-diffuse reflection [30, 113] is often used in rarefied gas dynamics. For it,

$$\begin{aligned} W(\mathbf{v}, \mathbf{v}_1) &= \alpha \delta(\mathbf{v} - \mathbf{v}_1 + 2\mathbf{n}_\Gamma(\mathbf{v}_1; \mathbf{n}_\Gamma)) + \\ &+ (1 - \alpha) f_w(\mathbf{v}) |(\mathbf{v}, \mathbf{n}_\Gamma)|, \\ f_w(\mathbf{v}) &= \left(2\pi \frac{kT_w}{m}\right)^{-3/2} \exp\left(-\frac{mv^2}{2kT_w}\right). \end{aligned} \quad (26)$$

Here α and $(1 - \alpha) \in [0, 1]$ are the probabilities of mirror and diffuse reflection of a molecule and T_w is the absolute temperature of the surface Γ . The sign of the normal velocity component of incident molecule only change under mirror reflection. This is expressed in the form of the argument of the Dirac delta-function in (26). Diffusely scattered molecules have normal (Gaussian) distribution [30] with the dispersion

$$\sigma^2 = \frac{kT_w}{m}.$$

6.3. Some procedures of Monte Carlo methods

Algorithms of statistical modeling (Monte Carlo methods) form a basis of the statistical particle methods. At least a brief overview of these methods would be helpful here. It will make possible a consistent presentation of the material in this chapter. Here, we restrict ourselves to some simple computer simulation procedures of the samples of random quantities with a given distribution [114]. In accordance with the terminology accepted in Monte Carlo methods, such procedures are called *simulating* of random quantities. The simulating represents, as a rule, a certain algorithm of transformation of a random quantity uniformly distributed in interval $[0, 1]$ into a given random quantity.

In the process of computer generation, special number sequences are obtained with the help of special programs. At certain assumptions, these numbers can be considered as uniformly distributed on $[0, 1]$. In the literature, these numbers are called pseudo-random to emphasize their model character. The corresponding programs are known as random number generators, and they are included in standard software. They are often called "RAND" (RANDU) procedure.

6.3.1. The inversion method

It is supposed that the reader should be familiar with elementary concepts of probability theory [34], which are used here without comments.

Let R denote a random quantity uniformly distributed on $[0, 1]$. Its distribution function (DF) is as follows:

$$F_R(x) = P(R < x) = \int_0^x dt \cdot 1 = x \quad (27)$$

Let ξ be a random quantity with a probability density $p_\xi(x)$ being simulated, and DF is

$$F_\xi(x) = P(\xi < x) = \int_{-\infty}^x dt p_\xi(t) \quad (28)$$

To obtain a simulating formula of the form

$$\xi = \varphi(R), \quad (29)$$

we assume that φ is strictly monotone on $[0, 1]$. Substituting the expression (29) into (28) and making evident transformations, we obtain successively

$$\begin{aligned} P(\xi < x) &= P(\varphi(R) < x) = \\ &= P(R < \varphi^{-1}(x)) = \varphi^{-1}(x), \end{aligned} \quad (30)$$

where the final equality follows from (27).

From (28) and (30), we have

Procedure 1.1. The sought-for simulating formula is

$$\varphi(R) = F_{\xi}^{-1}(R). \quad (31)$$

This means that the random quantity being simulated (28) is obtained by solving of the equation

$$\int_{-\infty}^x dt p_{\xi}(t) = R \quad (32)$$

for x .

Let us give two examples of using equation (32).

Procedure 1.2. The random quantity ξ with uniform (rectangular) distribution on $[a, b]$.

$$p_{\xi}(x) = \begin{cases} \frac{1}{(b-a)}, & x \in [a, b]; \\ 0, & x \notin C[a, b]. \end{cases}$$

In accordance with (31) and (3.2), the formula

$$\xi = a + R(b - a). \quad (33)$$

holds.

Procedure 1.3. The random quantity ξ with exponential distribution

$$p_{\xi}(x) = \nu \exp(-\nu x), \quad \nu, x > 0.$$

The simulating formula has the following form:

$$\xi = -\frac{1}{\nu} \ln(1 - R). \quad (34)$$

In applications, there are often discrete random quantities with a finite set of values $\{x_i\}$. A tabular representation,

$$\left(\begin{array}{cccc} x_1 & \dots & \dots & x_m \\ q_1 & \dots & \dots & q_m \end{array} \right),$$

is used for them. It means that

$$P(\xi = x_i) = q_i, \quad \sum_{i=1}^m q_i = 1. \quad (35)$$

The set $\{q_i\}$ generates some ordered decomposition of the section $[0, 1]$ into intervals of length q_i , $i = 1 \dots M$. The probability that the random quantity R will get into the interval q_i depends only on the interval length:

$$\begin{aligned} P\left(\sum_{m=1}^{i-1} q_m < R < \sum_{m=1}^i q_m\right) &= \\ &= \sum_{m=1}^i q_m - \sum_{m=1}^{i-1} q_m = q_i. \end{aligned} \tag{36}$$

A simulation scheme of the discrete random quantity (35) by inversion method follows from this. The following procedure is used:

Procedure 1.4. Simulation is determined by the condition

$$\begin{aligned} \sum_{m=1}^{i-1} q_m < R < \sum_{m=1}^i q_m, \\ \xi = x_i. \end{aligned} \tag{37}$$

Actually, the distribution of this random quantity with allowance for (36) is as follows:

$$P(\xi = x_i) = P\left(\sum_{m=1}^{i-1} q_m < R < \sum_{m=1}^i q_m\right) = q_i.$$

The sought-for inequality in the condition (37) can be tested by successive summation of q_i .

Further, the following example would be helpful.

Procedure 1.5. The integer random value ξ with the distribution

$$P(\xi = k) = \frac{1}{N}, \quad k = 1, \dots, N.$$

In accordance with (11), the simulating formula is represented in the following form:

$$k = E[NR] + 1, \tag{38}$$

where $E(x)$ is the integer part of the quantity x .

6.3.2. Simulation of joint distributions

If the density of a joint distribution is represented in a multiplicative form (independent random quantities)

$$p_{\xi_1, \dots, \xi_n}(x_1, \dots, x_n) = \prod_{i=1}^n p_{\xi_i}(x_i),$$

then simulation is reduced to the simulating of one-dimensional distributions.

For two-dimensional random quantities with a continuous density, the relations

$$p_{\xi\eta}(x, y) = p_\xi(x/y)p_\eta(y), \tag{39}$$

$$p_\eta(y) = \int_{-\infty}^{\infty} dx p_{\xi\eta}(x, y). \quad (40)$$

hold [34]. Using them, the simulation of a two-dimensional random quantity is performed in two stages.

Procedure 2.1. The next value of the random quantity $\eta = y_i$ is sampled with the use of the density (40). Then the value $\xi = x_i$ is simulated on the basis of the conditional density $p_\xi(x/y_i)$. By virtue of the relations (39) and (40), the pairs (x_i, y_i) will be a sample from the joint distribution (39).

For a greater number of variables, the formulas that generalize (39) and (40) are used. This also reduces the sampling to successive simulation of one-dimensional conditional distributions [114].

It should be noted that for the random vector $\bar{\xi} = (\xi_1, \dots, \xi_n)$ with uniform distribution in $D \subset R^n$, the density is expressed by the relation

$$p_{\xi_1, \dots, \xi_n}(x_1, \dots, x_n) = \frac{\chi_D}{mes D}. \quad (41)$$

Here χ_D is the characteristic function (indicator) of domain D and the denominator is the Lebesgue measure of the domain.

Sometimes the replacement of the variables in the initial distributions makes it possible to obtain simple simulating formulas. Let us consider, as an example, the sampling of a pair of standard normally distributed random quantities $N(0, 1)$. One usually starts from differential probabilities (probability elements) [34]. In this case, they are described in the following form:

$$p_\xi(x)dx p_\eta(y)dy = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})dx \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2})dy.$$

Making successive replacements,

$$x = r \cos \theta, \quad y = r \sin \theta, \quad z = \frac{r^2}{2}, \quad (42)$$

we obtain

$$p_\xi(x)p_\eta(y)dxdy = \exp(-z)dz \frac{1}{2\pi} d\theta$$

Hence, there follows

Procedure 2.2. In accordance with procedures 1.2 and 1.3 (see (33) and (34)), one can use for θ and z the simulating relations

$$\theta = 2\pi R_1, \quad z = -\ln(1 - R_2).$$

Their substitution into (42) yields the following simulating formulas for the pair of normal quantities:

$$\begin{aligned} x &= \sqrt{-2 \ln(1 - R_2)} \cos 2\pi R_1, \\ y &= \sqrt{-2 \ln(1 - R_2)} \sin 2\pi R_2. \end{aligned} \quad (43)$$

The simulation of the joint distributions of random quantities forms the basis of the methods described in the next subpoints.

6.3.3. The rejection method (by J. Neumann)

This algorithm is greatly universal. It is in the following. Let a continuous function, $f(x) \geq 0$, be given for $x \in [a, b]$. It is necessary to sample a random quantity ξ with the probability density

$$p_\xi(x) = f(x) / \int_a^b dx f(x). \quad (44)$$

It follows from the general formula (41) that the density of uniform distribution of a two-dimensional random quantity (ξ, η) in the domain

$$D = \{x \in [a, b], 0 \leq y \leq f(x)\}$$

is

$$p_{\xi\eta}(x, y) = \chi_D / \int_a^b dx f(x) \quad (45)$$

In this case, it follows from the formula (40) that

$$\begin{aligned} p_\xi(x) &= \int_{-\infty}^{\infty} dt p_{\xi\eta}(x, t) = \\ &= \int_0^{f(x)} dt / \int_a^b dx f(x) = f(x) / \int_a^b dx f(x). \end{aligned} \quad (46)$$

This justifies

Procedure 3.1. The pairs (ξ, η) with a uniform distribution in D are simulated. Here, the random quantities $\{\xi\}$ have the required distribution (44). The simplest way to construct the sought-for random quantity is as follows. Points with the coordinates

$$x = a + R_1(b - a), \quad y + R_2 \max_{x \in [a, b]} f(x),$$

uniformly distributed in a minimal rectangle containing domain D are simulated by using the formula (33). If $y < f(x)$, then $\xi = x$. Otherwise, the point is rejected.

The Neumann method can be effectively used to simulate discrete random numbers with the distribution (35). Let us consider the example of an integer random quantity:

$$P(\xi = m) = g_m / \sum_{l=1}^L g_l, \quad \forall g_l > 0.$$

It is encountered in schemes of the statistical particle method (see p. 4).

Procedure 3.2. By using simulating formula (38), we find the number

$$m = E[LR_1] + 1.$$

If

$$\eta = R_2 < \frac{g_m}{G}, \quad G = \max_l g_l,$$

then $\xi = m$; otherwise, the pair (ξ, η) is rejected.

6.3.4. The superposition method

The representation for partial probability density

$$p_\xi(x) = \int_{-\infty}^{\infty} dy p_\xi(x/y) p_\eta(y), \quad (47)$$

obtained from formula (39) is the basis of the method for continuous random quantities. If η is an integer random quantity with the distribution law

$$P(\eta = i) = q_i, \quad i = 1, \dots, M.$$

there exists, instead of (47), the formula

$$p_\xi(x) = \sum_{i=1}^M q_i p_i(x). \quad (48)$$

In practice, the method is often used to simulate random quantities with a distribution of the form (48). Thus, is evident

Procedure 4.1. The random quantity with the number $i = m$ is simulated with the help of Procedure 1.4 or 3.2. Then, the random quantity ξ is simulated by using the density $p_m(x)$. Formula (47) determines a similar procedure.

6.4. Statistical particle methods

6.4.1. General characteristics

An Eulerian mesh G_h that can be non-uniform is introduced in the calculation flow domain G . The spatial changes of the DF $f(\mathbf{r}, \mathbf{v}, t)$ at each Eulerian cell are assumed to be small. Usually, the mean free path of particles

$$\lambda \cong (n\sigma_t)^{-1} \quad (49)$$

can be taken as the scale of the spatial variation of DF in a rarefied gas.

Therefore, the Eulerian mesh spacing h should be chosen from the condition

$$h < \lambda.$$

Since we use the splitting-up scheme (16)-(17) in which the process of uniform collisional relaxation with a typical time τ_λ distinguished, the requirement that the change in DF on the time step τ be small imposes the following restriction on it:

$$\tau \ll \tau_\lambda. \quad (50)$$

A characteristic peculiarity of the statistical particle methods is as follows: no extrapolation of the features carried by particles (mass, momentum, etc.) to the nodes G_h is made (cf. the previous chapters). Instead of this, on the Euler step the process of collisional relaxation

(17) of the DF is reproduced within each spatial cell by binary collisions of model particles.

Collisions are simulated by Monte Carlo methods, with particles of only one cell as collision partners. In this case, the locations of model particles within one cell do not differ. The spatial coordinates of all particles in a given cell are considered to coincide with its center.

This approach is consistent with the statistical character of the collision integral (7), (14) and the determination of all DF incoming in it at one spatial point \mathbf{r} (the short-range interaction of gas molecules). Thus, in this method the Eulerian mesh G_h is auxiliary, supporting the splitting up scheme and specifying the discreteness of calculation of DF and its moments (2)-(4).

Since the method is imitational in character, the particles must reproduce the kinetics of gas molecules and the macrocharacteristics of gas, as in plasma (Chapter 5). Even in a very rarefied gas, however, the molecule number density is high ($n \sim 10^{15} \text{ cm}^{-3}$). It is clear that even modern supercomputers cannot process such volumes of data. There occurs a typical situation when

$$\frac{n}{n_M} = \alpha \gg 1, \quad (51)$$

Here n_M is the number of model particles per unit volume.

For the "gas" of model particles some relations follow from the conditions of adequate simulation and the last inequality. A part of them is essentially used in the computational process. Others were mistakenly considered in the literature as arguments against the simulation of RGD problems by statistical particle methods. Therefore, at least a brief analysis of these relations is necessary.

The ratio between the mean free path and the characteristic scale of the flow domain is determined by the Knudsen number (see (19)):

$$Kn = \frac{\lambda}{L}.$$

It is a similarity criterion [113], which value should be the same in the natural and "model" gas. If we assume for simplicity that the characteristic size in both cases is unity, the conservation of the number Kn requires that the mean free paths be equal:

$$\lambda = \lambda_M \quad (52)$$

It follows from it, with allowance for (49) and (51), that

$$\frac{(\sigma_t)_M}{\sigma_t} = \alpha \gg 1 \quad (53)$$

That is, model particles have a large scattering cross-section. It is known [113] that the condition of ideal gas is the possibility to neglect by the volume of its molecules. This condition can be written in the following form:

$$n\sigma_t^{\frac{3}{2}} \ll 1.$$

The corresponding quantity for the model gas is expressed as follows:

$$n_M(\sigma_{tM})^{\frac{3}{2}} = n\sigma_t^{\frac{3}{2}}\alpha^{\frac{1}{2}} \quad (54)$$

It is easy to see that the condition of perfection of the model gas is often not satisfied or close to a critical limit.

Setting equal the mesh densities of gases, we obtain the following relation for the mass of model particles:

$$\frac{m_M}{m} = \alpha \gg 1 \quad (55)$$

To simulate the relaxation stage correctly, it is important to use a single time scale given by the mean free time τ_λ (18). The required equality can be written in the following form:

$$n\sigma_t C = n_M(\sigma_t)_M C_M, \quad (56)$$

Here C and $(C)_M$ are root mean square deviations (the square root of dispersion) in the real and model velocity distributions of the molecules.

If we take into account the condition (52), the relation (56) is reduced to the requirement that the dispersions of DF of the real and model gases be equal or the same

$$C = C_M. \quad (57)$$

A correct interpretation of the model ensemble of particles is as follows. The entire ensemble of particles in the calculation domain G_h forms a representative sample of the DF being simulated. Each model particle can be considered as a set of a great number of real molecules with close velocities and coordinates. This view corresponds to the grouping procedure of sampled data known in mathematical statistics [34].

The collisions of particles are reproduced by means of statistical simulation of the scattering problem. Below we shall see that the simulation formulas include only the quantities that are present in the conditions (52) and (56). The sizes themselves of the "model" molecules associated with the formally unacceptable relations (53) and (54) are not used in the simulation in any way. Formulas (9) for the velocities of molecules after the collision and the conservation laws (10) do not contain the molecular mass. Therefore, in spite of (55), the dynamics of model particles does not change. This is also valid for the scattering of particles on the hard surface S (see (25) and (26)).

6.4.2. The scheme and calculation cycle for the Boltzmann equation

Let us consider the scheme proposed in [119, 120]. It makes it possible to use a universal approach to the introduction of the Lagrangian mesh of model particles, which is employed in all chapters here.

1. The Lagrangian step

We consider the DF at the Lagrangian step in the following form:

$$\varphi(\mathbf{r}, \mathbf{v}, t) = \sum_{i=1}^M \delta(\mathbf{r} - \mathbf{r}_i(t)) \delta(\mathbf{v} - \mathbf{v}_i(t)), \quad (58)$$

Here M is the total number of model particles in the domain G_h at the start time of the step, δ is three-dimension Dirac delta-function. The use of singular (point) particles is typical for statistical RGD methods. In this case, we do not need to smooth the interaction between particles at small distances, because it is not considered directly, but simulate statistically by using the scattering problem. Also, it is not necessary to increase artificially the radius of spatial influence of a particle, as in the methods of particles for media with long-range action (see Chapters 4 and 5). There remains, however, the problem of fluctuations under small numbers of particles in cells, which is common for all particle methods. As above, the fluctuations particles can be reduced by use of finite kernels. However, here the number of particles in a cell is bounded from below rather strongly by the requirement of adequate simulation of collision rate.

Let us substitute DF (58) into equation (16) of Lagrangian step, multiply it by the smooth finite function $\psi(\mathbf{r}, \mathbf{v})$ and integrate over $R^3(\mathbf{r}) \times R^3(\mathbf{v})$:

$$\begin{aligned} & \sum_{i=1}^M \left(\frac{\partial}{\partial t} \psi(\mathbf{r}_i(t), \mathbf{v}_i(t)) + \right. \\ & + \int d\mathbf{r} \psi(\mathbf{r}_i(t), \mathbf{v}_i(t)) \mathbf{v}_i(t) \frac{\partial}{\partial \mathbf{r}} \delta(\mathbf{r} - \mathbf{r}_i(t)) + \\ & \left. + \frac{\mathbf{Q}(\mathbf{r}_i)}{m} \int d\mathbf{v} \psi(\mathbf{r}_i(t), \mathbf{v}) \frac{\partial}{\partial \mathbf{v}} \delta(\mathbf{v} - \mathbf{v}_i(t)) \right) = 0. \end{aligned}$$

Calculating the derivatives of generalized functions, we derive the following relation:

$$\begin{aligned} & \sum_{i=1}^M \left(\left(\frac{d\mathbf{r}_i}{dt} - \mathbf{v}_i \right) \frac{\partial \psi(\mathbf{r}_i, \mathbf{v}_i)}{\partial \mathbf{r}_i} + \right. \\ & \left. + \left(\frac{d\mathbf{v}_i}{dt} - \frac{\mathbf{Q}(\mathbf{r}_i)}{m} \right) \frac{\partial \psi(\mathbf{r}_i, \mathbf{v}_i)}{\partial \mathbf{v}_i} \right) = 0. \end{aligned}$$

Hence, by virtue of the arbitrary character of the functions $\psi(\mathbf{r}, \mathbf{v})$, one can obtain a system of equations for the coordinates and velocities of model molecules. It describes their motion at the transport (Lagrangian) step

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \quad \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{Q}(\mathbf{r}_i)}{m}, \quad i = 1, \dots, M. \quad (59)$$

As noted in s.p. 1.1, the force field $Q(\mathbf{r}_i)$ is assumed to be a given function and does not depend on DF.

Usually the integration step τ is determined by the condition (50) for the Eulerian step and is sufficiently small. Therefore, the simplest explicit schemes are used for numerical solving of the system (59) in RGD problems. We shall not consider them here, since equations similar to Eq. (59), but for a more complex case of plasma dynamics, are described in detail in Chapter 5 (see also the program in Supplement C).

If the displacement of the i -th particle in accordance with equations (11) is not outside the Eulerian cell, its coordinate \mathbf{r} remains the same:

$$\mathbf{r}_i^p = \mathbf{r}_i^{p+1}.$$

After the trajectory of the particle crosses the external (permeable) boundary of the calculation domain S_h , it leaves the model ensemble, and it can no longer be traced.

When, at some $\tau_* < \tau$, the trajectory of a model molecule reaches the hard surface Γ , the boundary condition (25) is simulated by Monte Carlo methods. In particular, the superposition method considered in s.p. 3.4 is used for simulation of the mirror-diffuse scattering (26). In accordance with Procedure 4.1, if $R_1 < \alpha$, a particle imitates mirror reflection. The projection of the velocity vector on the local normal \mathbf{n}_Γ changes its sign. At $R_1 > \alpha$, diffuse scattering into half-space is simulated in accordance with Procedure 2.2. The molecular features, i.e. momentum components, transmitted energy, etc., are fixed in the surface cell on Γ .

After the "collision" of a particle with Γ , its motion on the time interval $\tau - \tau_*$ is traced. Its velocity and new location in G_h in the $(p+1)$ -th time layer are found from equations (59).

At the Lagrangian step, the condition (24) on the permeable surface S is also simulated. For this, new particles are introduced into the boundary cells of the calculation domain G_h . The locally Maxwell DF of the form (15) whose parameters depend on the coordinate \mathbf{r}_s is often specified as a boundary function $f_s(\mathbf{r}_s, \mathbf{v})$. It is convenient to choose a Cartesian system of coordinates connected with the local normal \mathbf{n}_s so that the vector $\mathbf{U}(\mathbf{r}_s)$ lie in one of the coordinate planes. Then the two velocity components of a new particle in the coordinate plane tangent to S are sampled by the Procedure 2.2. After that, the projection of the average velocity $\mathbf{U}(\mathbf{r}_s)$ to the given axis is added to one of them:

$$v = v' + |\mathbf{U}| \cos \theta_s,$$

where

$$\cos \theta_s = \frac{(\mathbf{U}(\mathbf{r}_s) \mathbf{n}_s)}{|\mathbf{U}(\mathbf{r}_s)|}.$$

The normal velocity components are simulated by the Neumann rejection method considered in s.p. 6.3.3. The explicit form of the procedure can be found in [30].

For simplicity, particles are sometimes injected only in those cells for which $\cos \theta_s > 0$. The error admitted in this case is the smaller the greater the relation $|U|/(2kT_s/m)^{1/2}$.

An additional requirement is that the relations between the number of particles injected in individual sections S_i of the boundary S be proportional to the corresponding number fluxes

$$Q_i = \int_{S_i} d\mathbf{s} \int_{(\mathbf{v}, \mathbf{n}_s) > 0} d\mathbf{v} (\mathbf{v}, \mathbf{n}_s) f_s(\mathbf{r}_s, \mathbf{v}).$$

On the other hand, the number of injected particles is limited by the tendency to preserve their total number in the model ensemble.

The Lagrangian step is finished after calculation of the new coordinates and velocities of all particles at the $(p+1)$ -th time moment.

2. The Eulerian step

After the Lagrangian step, each Eulerian cell, with the center at the j -th node into which the domain G_h is divided, contains N_j^{p+1} particles with a set of velocities $\mathbf{V}_j^{p+1} = (\mathbf{v}_1^{p+1}, \dots, \mathbf{v}_N^{p+1})$. The cell center determines their common spatial coordinate. Since the state of all cells changes in accordance with the same algorithm, the subscript that denotes the cell index will be omitted. Therefore, it should be noted at the consideration of the algorithms of simulation of collisional relaxation described below that they are used to process the entire array of Eulerian cells.

A distinctive feature of the methods described in this chapter is in the following. A discrete representation of DF by model particles is also used at the Eulerian step instead of defining of mesh functions. This is caused by the fact that statistical simulation of the evolution of DF for a relatively small sample of particles turns out to be much more economical than the use in the problem (18) of the finite difference approximation in combination with some classical quadrature formula for the multidimensional integral in $J(f, f)$. Besides, at this approach so much computer time would be required that is not realistic for a medium-power computer.

For simplicity, let us restrict ourselves to the case when equation (17) contains the model collision integral (14). Let us approximate the BGK-equation by the following explicit difference scheme:

$$\frac{\tilde{f}^{p+1} - f^p}{\tau} = \nu^p (f_0^p - f^p).$$

Taking into account the fact that the initial data for the problem (17) are taken from the upper layer of the Lagrangian step, the solution at the $(p+1)$ -th time moment can be represented as follows:

$$\tilde{f}^{p+1} = (1 - \tau\nu^p)\varphi^{p+1} + \tau\nu^p f_0^p, \quad (60)$$

Here ν^p and f_0^p are determined by the parameters n^{p+1} , \mathbf{U}^{p+1} and T^{p+1} . They are calculated by formulas (2)-(4) as the moments of DF φ^{p+1} .

It is convenient to represent the sample DF φ^{p+1} within an Eulerian cell in the form

$$\varphi^{p+1} = \frac{n^{p+1}(\mathbf{r})}{N} \sum_{i=1}^N \delta(\mathbf{v} - \mathbf{v}_i^{p+1}) \quad (61)$$

Here N is the total number of model molecules in an Eulerian cell at a given time moment; $\{\mathbf{v}_i^{p+1}\}$ is a set of velocities of model particles; δ is multidimensional Dirac delta function. The numerical particle density is determined as

$$n(\mathbf{r}) = \frac{N}{\Delta},$$

where Δ is the cell volume.

Substitution of the sample DF (61) into the integrals (3) and (4) makes it possible to find the mean macroscopic velocity \mathbf{U}^{p+1} and the gas temperature T^{p+1} as the sample mean of \mathbf{V}^{p+1} ,

$$\mathbf{U}^{p+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i^{p+1}, \quad (62)$$

and sample dispersion

$$3\frac{k}{m}T^{p+1} = \frac{1}{N} \sum_{i=1}^N (\mathbf{v}_i^{p+1})^2 - (\mathbf{U}^{p+1})^2 \quad (63)$$

On the other hand, the local Maxwell function f_0^{p+1} can be represented similarly to (61) as a sample DF:

$$f_0^{p+1} = \frac{n^{p+1}}{N} \sum_{i=1}^N \delta(\mathbf{v} - \mathbf{v}'_i), \quad (64)$$

Here the set $\mathbf{V}' = (\mathbf{v}'_1, \dots, \mathbf{v}'_N)$ is a sample of the volume N from the distribution f_0^{p+1} with the parameters n^{p+1} , \mathbf{U}^{p+1} , and T^{p+1} . It is clear that the following relations are satisfied for the sample \mathbf{V}' :

$$\mathbf{U}^{p+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}'_i, \quad (65)$$

$$3\frac{k}{m}T^{p+1} = \frac{1}{N} \sum_{i=1}^N (\mathbf{v}'_i)^2 - (\mathbf{U}^{p+1})^2 \quad (66)$$

Substitution of (61) and (64) into relation (60) leads to the following expression of the solution

$$\begin{aligned} \frac{1}{n^{p+1}} \tilde{f}^{p+1}(\mathbf{v}) &= \frac{1}{N} \times \\ &\times \sum_{i=1}^N ((1-q)\delta(\mathbf{v} - \mathbf{v}_i^{p+1}) + q\delta(\mathbf{v} - \mathbf{v}'_i)). \end{aligned} \quad (67)$$

In the right-hand side of (67), we introduce

$$q = \tau\nu^p.$$

Recall that the function ν^p has the meaning (see (12) and (14)) of the molecular collision frequency. In this case,

$$\nu^p \sim \frac{1}{\tau_\lambda}$$

and, by virtue of the limitation (50) on the step τ , the quantity

$$1 - q \simeq 1 - \frac{\tau}{\tau_\lambda} > 0 \quad (68)$$

is positive.

Integrating the equality (67) over $R^3(\mathbf{v})$, we obtain

$$\int d\mathbf{v} \frac{1}{N} \sum_{i=1}^N ((1-q)\delta(\mathbf{v} - \mathbf{v}_i^{p+1}) + q\delta(\mathbf{v} - \mathbf{v}'_i)) = 1 \quad (69)$$

The last two relations allow us to consider the right-hand and, hence, the left-hand side of Eq. (67) as a generalized (in the meaning of the theory of generalized functions [121]) probability density distribution defined on samples of the size N .

This leads to the algorithm of transformation of the initial sample φ^{p+1} (61) into the sample \tilde{f}^{p+1} by the Monte Carlo method. For this, we should use the superposition method from p. 2. The following steps are performed.

1. With the help of Procedure 1.5, the model particle index is chosen from the uniform distribution

$$P(\xi_1 = i) = \frac{1}{N}. \quad (70)$$

2. Then the discrete random quantity ξ_2 with the distribution

$$\begin{pmatrix} \xi_2^1 & \xi_2^2 \\ 1-q & q \end{pmatrix} \quad (71)$$

is simulated by using Procedure 1.4. If $\xi_2 = \xi_2^1$, the model particle with the index i considered does not change its velocity \mathbf{v}_i^{p+1} . If, however, the value $\xi_2 = \xi_2^2$ is taken, the velocity of this particle should be replaced by the velocity \mathbf{v}'_i . It is obtained by the simulation with the density distribution

$$\frac{1}{n^{p+1}} f_0^{p+1} = \left(\frac{m}{2\pi k T^{p+1}} \right)^{\frac{3}{2}} \exp \left(-\frac{m(\mathbf{v}' - \mathbf{U}^{p+1})^2}{2kT^{p+1}} \right) \quad (72),$$

which is carried out with the help of Procedures 2.2 and 3.1 (see also [30]).

Since as a result of the simulation the sample of size N again goes to the sample of the same size, it is sufficient to perform successive item-by-item examination of all particles instead of a random choice of the particle index from the distribution (70).

It is seen that the amount of calculations in the algorithm is linearly dependent on N . Therefore, in this case the procedure of the Eulerian step is economical.

Let us show that the algorithm is conservative, in the sense of conserving the number density n^{p+1} and mean sample values (62) and (63) at the transition from DF φ^{p+1} to DF \tilde{f}^{p+1} . The conservation of n^{p+1} is an evident consequence of the invariability of the sample size N and the volume of the cell Δ in the process of simulation. Formally, the same conservation law can be obtained by integrating expressions (61) and (67) over $R^3(\mathbf{v})$ (see also (69)).

The conservation of the velocity \mathbf{U}^{p+1} is proved by the integration of expression (67) with respect to $R^3(\mathbf{v})$:

$$\begin{aligned} & \frac{1}{n^{p+1}} \int d\mathbf{v} u \tilde{f}^{p+1}(\mathbf{v}) = \\ & = (1-q)\mathbf{U}^{p+1} + q\mathbf{U}^{p+1} = \mathbf{U}^{p+1}, \end{aligned}$$

where relations (62) and (65) are taken into account in the transformations of the right-hand side.

The conservation of the temperature (internal energy) is proved in a similar way:

$$\begin{aligned} & \frac{1}{n^{p+1}} \int d\mathbf{v} \mathbf{v}^2 \tilde{f}^{p+1}(\mathbf{v}) = \\ & = (1-q)\frac{3kT^{p+1}}{m} + q\frac{3kT^{p+1}}{m} = \frac{3kT^{p+1}}{m}. \end{aligned}$$

The final expression is obtained here with the use of (63) and (66) as well as the conservation laws established above.

It should be noted that the quantity

$$q = \tau\nu \sim \frac{\tau}{\tau_\lambda}$$

can be interpreted as the probability of collision of a molecule on a small time interval τ . Indeed, the BGK-equation does not contain collision acts, and their result, as we see, is simulated by the density distribution (72).

An analogous simulation algorithm of the Eulerian step is constructed for the Boltzmann equation with the exact collision integral [119].

Its derivation, however, would require rather long transformations of the integral of collisions $J(f, f)$ (7). Besides, this algorithm is not perfect from the point of view of calculations [122]. In particular, it is not economical, because in this case the amount of calculations for the transition on the step τ in the Eulerian cell is proportional to N^2 [119].

6.4.3. The scheme and calculation cycle for the master kinetic equation

Modern algorithms for the Eulerian step in the statistical method of particles are usually constructed on the basis of the M -equation (20). Its connection with the Boltzmann equation at the multiplicative initial data was shown in s.p. 2.3. A peculiarity of these algorithms is the use of the scheme of fractional steps. The fractional steps τ_k are considered as random quantities varying within the time step τ at which the splitting-up scheme (16), (21) is constructed. It is clear that only a satisfactory approximation from below can be achieved instead of the accurate equality

$$\sum_{k=1}^n \tau_k = \tau$$

required. From here, there follows a new factor that determines the additional approximation error at the stage of collisional relaxation.

Let us present the derivation of the simulation process for the equation (20), which was first considered in [28]. For this, we approximate the M -equation on the fractional steps τ_k by the explicit difference scheme

$$\begin{aligned} \frac{F_N^{k+1} - F_N^k}{\tau_k} &= \frac{n}{N} \frac{d_0^2}{4} \times \\ &\times \sum_{i<j}^N \int d\Omega_{ij} g_{ij} (F_N^k(\mathbf{V}'_{ij}) - F_N^k(\mathbf{V})). \end{aligned} \quad (73)$$

Superscript k is used to distinguish fractional steps, and superscript p denotes time moments in the splitting scheme.

As in s.p. 4.2, at the next fractional step τ_k we consider the sample of size N as initial data, but represent it in the form of a sample from the multidimensional joint distribution F_N :

$$F_N^k = \delta(\mathbf{V} - \mathbf{V}_k) = \prod_{i=1}^N \delta(\mathbf{v}_i - \mathbf{v}_i^k) \quad (74)$$

At $k = 1$, the corresponding set of velocities is the result of the preceding Lagrangian step. It is seen from Eq. (74) that F_N^k satisfies the condition of normalization and has a multiplicative form (see s.p. 2.3).

Solving (73), we write the solution at the $(k + 1)$ -th fractional step as follows:

$$F_N^{k+1} = (1 - \tau_k \nu^k) F_N^k + \tau_k \nu^k \Psi_N^k \quad (75)$$

Here, we introduce the notation

$$\nu^k = \frac{n}{N} \frac{d_0^2}{4} \sum_{i < j}^N g_{ij}^k \int d\Omega_{ij} = \frac{n}{N} \sigma_t \sum_{i < j}^N g_{ij}^k, \quad (76)$$

$$\Psi_N^k = \frac{1}{\nu^k} \frac{n}{N} \frac{d_0^2}{4} \sum_{i < j}^N g_{ij}^k \int d\Omega_{ij} F_N^k(\mathbf{V}_{ij}^k) \quad (77)$$

The total collision cross-section σ_t for the model of hard spheres was determined in (13). Since (see (61))

$$\frac{n}{N} = \frac{1}{\Delta},$$

the quantity ν^k is the number of molecular collisions per unit time in a unit volume of an Eulerian cell (collision rate).

It is seen that the structure of Eq. (75) is similar to that of Eq. (60), but its probabilistic interpretation is somewhat more complex.

The fact of collision of a pair of model particles on a time interval τ_k will be considered as a probabilistic event. We characterize it as follows. The probability q of its realization on a small time interval dt is proportional to the length of the interval

$$q = \nu_k dt.$$

The probability of realization of two or more events (collisions) on the same small interval $(t, t + dt)$ is equal to zero for any t . These properties show unambiguously that the time intervals between two successive collisions (the fractional steps τ_k) are distributed in accordance with the Poisson law [34] with the density

$$p(\tau_k) = \nu_k \exp(-\nu_k \tau_k). \quad (78)$$

Let us calculate the mathematical expectation of the random variable τ_k

$$M\tau_k = \int_0^\infty dt t \nu_k \exp(-\nu_k t) = \frac{1}{\nu_k},$$

Here M is the symbol of mathematical expectation. That is, the mean time between the collisions is expressed as the reverse quantity ν_k , which coincides with the physical meaning of this function. It can be noted that ν_k is the rate of the Poisson event flow, which is here a succession of collisions.

The use of the Poisson distribution makes it possible to obtain correct collisions statistics in a cell, without its simulation by small steps, as in the previous algorithm (see (71)).

Now, it should be demonstrated that the function Ψ_N^k can be considered as a certain density distribution of the multidimensional random

variable $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_N)$. First, it is seen from the definition of Ψ_N^k that it is nonnegative. Therefore, it remains to prove that the normalization condition

$$\int d\mathbf{V} \Psi_N^k(\mathbf{V}) = 1. \quad (79)$$

is satisfied.

After the substitution of expressions (74) and (76) into Eq. (77), we have

$$\Psi_N^k(\mathbf{V}) = \sum_{i < j}^N \frac{g_{ij}^k}{\sum_{i < j}^N g_{ij}^k} \prod_{m \neq i, j}^N \delta(\mathbf{v}_m - \mathbf{v}_m^k) \psi_{ij}(\mathbf{v}_i, \mathbf{v}_j), \quad (80)$$

$$\psi_{ij}(\mathbf{v}_i, \mathbf{v}_j) = \frac{1}{4\pi} \int d\Omega_{ij} \delta(\mathbf{v}'_i - \mathbf{v}_i^k) \delta(\mathbf{v}'_j - \mathbf{v}_j^k). \quad (81)$$

As in the initial form of the M -equation (20), here subscripts i and j denote the pair of collision partners. We write their velocities connected by the collision transform (21) with the help of relations (9) in the following form:

$$\mathbf{v}'_i = \mathbf{R}_{ij} + \frac{1}{2} g_{ij} \Omega_{ij}, \quad \mathbf{v}'_j = \mathbf{R}_{ij} - \frac{1}{2} g_{ij} \Omega_{ij}, \quad (82)$$

$$\mathbf{v}_i^k = \mathbf{R}_{ij}^k + \frac{1}{2} g_{ij}^k \Omega_{ij}^k, \quad \mathbf{v}_j^k = \mathbf{R}_{ij}^k - \frac{1}{2} g_{ij}^k \Omega_{ij}^k.$$

The unit vector Ω_{ij} determines the direction of the relative velocity of particles after the collision. We introduce the notations

$$\begin{aligned} \mathbf{R}_{ij} &= \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j), & \mathbf{g}_{ij} &= \mathbf{v}_i - \mathbf{v}_j, \\ g_{ij} &= |\mathbf{g}_{ij}| \end{aligned} \quad (83)$$

for center-mass-velocities \mathbf{R}_{ij} and relative velocities \mathbf{g}_{ij} of collided particles.

At the k -th moment, the corresponding values for the velocities of particles before the collision are determined similarly.

To transform the integrand in (81), we use the representation of the Dirac delta function in terms of the Fourier transform of the unity [121]:

$$\delta(\mathbf{x}) = \frac{1}{(2\pi)^3} \int dk e^{i\mathbf{k}\mathbf{x}} 1. \quad (84)$$

With the help of the relations (82) and (84), one can write

$$\begin{aligned} \delta(\mathbf{v}'_i - \mathbf{v}_i^k) \delta(\mathbf{v}'_j - \mathbf{v}_j^k) &= \int \int dk_1 dk_2 \times \\ &\times \exp(i(\mathbf{k}_1 + \mathbf{k}_2)(\mathbf{R}_{ij} - \mathbf{R}_{ij}^k)) \times \\ &\times \exp(i(\mathbf{k}_2 - \mathbf{k}_1) \frac{1}{2}(g_{ij} \Omega_{ij} - g_{ij}^k \Omega_{ij}^k)). \end{aligned}$$

Let us perform an orthogonal replacement of the integration variables by using the following formulas:

$$\mathbf{p} = \mathbf{k}_1 + \mathbf{k}_2, \quad \mathbf{q} = \frac{1}{2}(\mathbf{k}_2 - \mathbf{k}_1).$$

Then, the following sequence of transformations is carried out :

$$\begin{aligned} \delta(\mathbf{v}'_i - \mathbf{v}'_i^k) \delta(\mathbf{v}'_j - \mathbf{v}'_j^k) &= \int d\mathbf{p} \exp(i\mathbf{p} \times \\ &\times (\mathbf{R}_{ij} - \mathbf{R}_{ij}^k)) \int d\mathbf{q} \exp(i\mathbf{q}(g_{ij}\Omega_{ij}) - g_{ij}^k \times \\ &\times \Omega_{ij}^k) = \delta(\mathbf{R}_{ij} - \mathbf{R}_{ij}^k) \delta(g_{ij}\Omega_{ij} - g_{ij}^k \Omega_{ij}^k) \\ \delta(g_{ij}\Omega_{ij} - g_{ij}^k \Omega_{ij}^k) &= \delta(g_{ij} - g_{ij}^k) \delta(\Omega_{ij} - \Omega_{ij}^k). \end{aligned} \quad (85)$$

If, at the integration with respect to the angular variables Ω_{ij} , we choose the polar axis collinearly to the vector Ω_{ij}^k , the generalized delta function concentrated on the unit sphere can be rewritten in the following form:

$$\delta(\Omega_{ij} - \Omega_{ij}^k) = \delta(\Omega_{ij}).$$

With allowance for the expression obtained in Eq. (85) and after subsequent transformations, let us consider the integral of the function $\psi_{ij}(\mathbf{v}_i, \mathbf{v}_j)$:

$$\begin{aligned} \iint d\mathbf{v}_i d\mathbf{v}_j \psi_{ij}(\mathbf{v}_i, \mathbf{v}_j) &= \frac{1}{4\pi} \times \\ &\times \iint d\mathbf{v}_i d\mathbf{v}_j \delta(\mathbf{R}_{ij} - \mathbf{R}_{ij}^k) \delta(g_{ij} - g_{ij}^k) \int d\Omega_{ij} \delta(\Omega_{ij}). \end{aligned}$$

By virtue of the definition of the δ -function on a unit sphere, the last integral is equal to unity. In the integral that remained, we can make an orthogonal replacement of variables, performing the transition to \mathbf{R}_{ij} and \mathbf{g}_{ij} by formulas (83).

As a result, we obtain

$$\iint d\mathbf{v}_i d\mathbf{v}_j \psi_{ij}(\mathbf{v}_i, \mathbf{v}_j) = 1 \quad (86)$$

Now, the function $\psi_{ij}(\mathbf{v}_i, \mathbf{v}_j)$ can be considered as the probability density distribution, which can be conveniently represented in the following multiplicative form:

$$\begin{aligned} \psi_{ij}(\mathbf{v}_i, \mathbf{v}_j) d\mathbf{v}_i d\mathbf{v}_j &= \delta(\mathbf{R}_{ij} - \mathbf{R}_{ij}^k) d\mathbf{R}_{ij} \times \\ &\times \delta(g_{ij} - g_{ij}^k) dg_{ij} \frac{1}{4\pi} d\Omega_{ij}. \end{aligned} \quad (87)$$

The product of the generalized densities in (87) implies that at a collision with probability equaled unity, the center-of-mass velocity of the pair of molecules \mathbf{R}_{ij}^k and the modulus of their relative velocity g_{ij}^k are conserved. The invariance of these quantities under transformation (21) is a direct consequence of the conservation laws (10).

The last multiplier in (87) is the uniform density distribution on a unit sphere, which can be transformed as follows:

$$\frac{d\Omega_{ij}}{4\pi} = p(\chi, \varphi) d\chi d\varphi = \frac{\sin \chi}{2} d\chi \frac{1}{2\pi} d\varphi, \quad (88)$$

Here χ and φ are the polar and azimuthal angles. Since the normalization condition (79) follows directly from (80) and (85), the function Ψ_N^k is actually the joint density distribution of the random quantity \mathbf{V} .

Hence, we obtained a full probabilistic interpretation of the right-hand side of equation (75). In accordance with it, the algorithm for the construction of a sample from F_N^{k+1} , at the next fractional step within the Eulerian step, is in the following.

1. The collision frequency

$$\nu^k = \frac{n}{N} \pi d_0^2 \sum_{i < j}^N g_{ij}^k. \quad (89)$$

is calculated by using the set of velocities $\mathbf{V}^k = (\mathbf{v}_1^k, \dots, \mathbf{v}_N^k)$.

2. With the use of the density distribution (78)

$$p(t) = \nu^k \exp(-\nu^k t),$$

and with the help of Procedure 1.3 of the inversion method, the time interval τ_k (fractional step) up to the next collision in a given Eulerian cell is sampled.

3. A pair of collision partners determined by the multi-subscript $l = (i, j)$ is chosen. For this, l is represented as a discrete random variable with the following probability distribution:

$$P(l = m) = \frac{g_m^k}{\sum_{l=1}^L g_l^k}, \quad (90)$$

$$L = \frac{1}{2} N(N - 1),$$

Here L is the total number of all kinds of pairs in the cell. This operation is prompted by the multiplier of evident sense in the right-hand side of the density distribution (80). The simulation of Eq. (90) is realized by Procedure 1.4 or Procedure 3.2 presented in p. 3.

4. Independent random values of the angles χ and φ that determine the vector Ω'_{ij} are simulated for the pair of particles (i, j) sampled. The quantity χ is obtained with the help of Procedure 1.1. It follows from (88) that the angle φ is uniformly distributed in the interval $[0, 2\pi]$. Its value is found from the simulating formula (33).

5. The velocities of the sampled pair $(\mathbf{v}_i^k, \mathbf{v}_j^k)$ are replaced by the velocities

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \mathbf{v}'_i = \mathbf{R}_{ij}^k + \frac{1}{2} g_{ij}^k \Omega'_{ij}, \\ \mathbf{v}_j^{k+1} &= \mathbf{v}'_j = \mathbf{R}_{ij}^k - \frac{1}{2} g_{ij}^k \Omega'_{ij}. \end{aligned} \quad (91)$$

Hence, the conservation laws (10) are exactly satisfied at each collision act. This means that at the Eulerian step the momentum and energy conservation laws are exactly satisfied for the whole system of model particles.

As a result of the next fractional step, the sample

$$\mathbf{V}^{k+1} = (\mathbf{v}_1^k, \dots, \mathbf{v}_i^{k+1}, \dots, \mathbf{v}_j^{k+1}, \dots, \mathbf{v}_N^k). \quad (92)$$

is obtained in the Eulerian cell.

6. The value of the fulfilled fractional step τ_k is added to the time counter S_n :

$$S_{k+1} = S_k + \tau_k, \quad S_0 = 0.$$

If the inequality

$$S_k \leq \tau \leq S_{k+1}, \quad (93)$$

is valid, the Eulerian step in this cell is completed. Its result is the sample (92). Otherwise, the cycle described is repeated until the inequality (93) will be satisfied.

It should be emphasized that the frequency of collisions ν^k in this algorithm is recalculated at each fractional step. It is evident from Eq. (89) that approximately $N^2/2$ operations are necessary even for a single recalculation of ν^k . Since correct simulation is provided by equality (56), the total number of collisions in a time of order of τ_λ is proportional to $n_m \sim N/\Delta$. Therefore, in this algorithm the amount of calculations for the transition on the step τ has the order $O(N^3)$. Thus, the algorithm described is far from economical.

Nevertheless, the simulation process considered is useful as a standard. A comparison with it makes it possible to analyze errors of various heuristic schemes for the Eulerian step. Correcting individual operations in it, one can purposefully construct approximate economical algorithms providing a satisfactory approximation. First constructions of such a type were carried out in [28].

6.4.4. The Bird algorithm

An economical algorithm for the Eulerian step was proposed by G. Bird, in whose papers the statistical particle-in-cell method for rarefied gas was initiated. It can be said with confidence that the success of the method was associated with a reasonable computational approach, which was successively followed in these papers.

A great amount of calculations in the exact algorithm for the M -equation is due to the need to recalculate the collision frequency ν_k in each cycle (76). Then it is used as a parameter in the Poisson law (78) for the sample of the next fractional step τ_k . To linearize the sampling procedure of τ_k with respect to N , Bird decided not to simulate the exact distribution (78). Instead of this, he proposed to consider fractional steps as the realization of a discrete random variable with the following values:

$$\tau_k(m) = \frac{2}{(N-1)n g_m^k \pi d_0^2}, \quad m = 1, \dots, L \quad (94)$$

For (94), the probability distribution is given in the following form (see also (90)):

$$P(\tau_k = \tau_k(m)) = P(l = m) = \frac{g_m^k}{\sum_{l=1}^L g_l^k} \quad (95)$$

The meaning of the relation (95) can be explained as follows. If a pair with the multisubscript m is sampled from the distribution (90) in a given cell, then the next time step (the time between the previous collision and this collision) takes the value (94). Here, to emphasize the new detail, the dependence on m is specially introduced into the notation of the step.

Notice that the Bird algorithm with discrete time steps is interpreted on the basis of the difference scheme (75) in a more natural way than the exact algorithm with the Poisson distribution. At such sampling of fractional steps (times between the collisions), the algorithm is clearly approximate. To understand its relation to the exact scheme presented in s.p. 4.3, we find the mathematical expectation of the random variable (94) with the distribution (95):

$$\begin{aligned} M\tau_k(m) &= \sum_{m=1}^L \tau_k(m) P(l = m) = \\ &= \frac{N}{n} (\sigma_t \sum_{m=1}^L g_m^k)^{-1} = \frac{1}{\nu^k}, \end{aligned} \quad (96)$$

where the latter equality is obtained with allowance for (76).

On the other hand, the following was obtained in s.p. 4.3 for the fractional steps in the exact algorithm continuously distributed by the Poisson law (78):

$$M\tau_k = \frac{1}{\nu^k}.$$

Thus, the mean values of fractional steps in the both algorithms coincide:

$$M\tau_k(m) = M\tau_k \quad (97)$$

Let us assume for simplicity that the transition on the splitting-up step τ in the cell is realized exactly, but not in the sense of satisfying the inequality (93). We represent the content of the time counter in the exact algorithm as follows:

$$S_P = \sum_{k=1}^n \tau_k = \tau.$$

Correspondingly, we have for the Bird scheme:

$$S_B = \sum_{k=1}^{n^*} \tau_k(m) = \tau.$$

These are two sums of independent random values. In S_P , each term is distributed with the probability density (78), where the parameter ν^k changes after each step. In S_B , the terms are discrete random quantities with a set of probabilities (95), which also change from step to step. In both cases, these changes are caused by the replacement of the velocities of collided particles in accordance with formulas (91).

Taking into account the above and relations (96) and (97), the equality

$$MS_P = MS_B.$$

holds. Since the equalities (97) for all terms are satisfied, it is clear that the number of fractional steps in the both algorithms is, "on average", the same:

$$Mn = Mn^*.$$

It is seen from these relations that the vicinity of both schemes should be understood as the coincidence, "on average", of the values of fractional steps and the collision numbers in a cell at one splitting up step τ . Besides, the error of the Bird algorithm decreases proportionally to N^{-1} [30]. There is, it is true, a certain contradiction here. That is, as we pass over to the algorithm linear over N , we have to increase the number of particles N in order to achieve the accuracy required. However, in practice, this contradiction is usually resolved at an acceptable level.

In his desire to derive an economical algorithm, to simulate the distribution (95) G. Bird used Procedure 3.2 of the Neumann method for the discrete random quantity. He employed it instead of Procedure 1.4 of the inversion method.

As a result, the cycle in the Bird algorithm is represented as follows:

1. A pair of collision partners is sampled in accordance with the discrete probability distribution (90). For this, the Neumann rejection method (Procedure 3.2) is used.
2. The value of the next fractional step is calculated by formula (94).

Then, there follow items 4 and 5 of the exact algorithm for the M -equation.

6. The value of the fractional step $\tau_k(m)$ (94) is brought into the time counter S_k , and the inequality (93) is verified.

The cycle is repeated until (93) is satisfied.

6.5. Examples of the application

The computational opportunities of statistical particle-in-cell methods (SPM) are determined by the accuracy, speed of convergence, and

other properties of the procedures of Monte Carlo methods. In particular, the accuracy typical for SPM is the same as that for statistical estimation, whose usual order lies within $\epsilon = 0.02 \div 0.05$ [114]. This means that for a DF with an exponential behavior of the form (15), a representative sample of the form (13) can be obtained in the following range:

$$|U| \pm 2\sqrt{\frac{2kT}{m}}$$

Hence, it is reasonable to use SPM for the problems with the main contribution to their solution of the thermal energy molecules. In such problems, it is required to calculate only the behavior of gas dynamic parameters, such as mass density, velocity, and temperature (see (2)-(4)) determined by the DF variation in this energy interval.

This class includes a wide range of problems of spacecraft aerodynamics and internal rarefied gas flows in a range of Knudsen numbers $Kn = 10^{-1} \div 10^0$. As mentioned above, the SPC originated on this basis in the 60ies and 70ies.

Useful properties of the methods became evident in the process of their development. They include the stability and gentle requirements to the Eulerian mesh, the possibility of construction of economical algorithms, and a rather fast convergence of iterations in the calculations by the establishing method.

This is the reason of a considerable progress in SPM in the last two decades. At the present time, they occupy an important place in the computational aerodynamics of rarefied gas due to a great volume of calculations performed with their use. From this point of view, they can be compared with the particle methods in plasma physics (see Chapter 5). It should be noted that these methods formed a basis for the calculation of complex applied problems of RGD.

We can take, as an example, the calculations of the aerodynamics of a multi-use spacecraft of the "Shuttle" type made by G. Bird in late 70ies. Some of the results obtained were published in [123]. A fairly realistic approximation of the spacecraft geometry by surfaces and curves, to the second order inclusive, was used. The number of cells in the flow field was of the order of 10^3 , and the number of model particles reached 10^4 . The calculations were carried out at the following Knudsen numbers: $Kn = 10^0 - 10$. Gas emission from the surfaces due to their degassing, catalytic reactions on them, and jets of the control engines were taken into account as additional factors.

Now, the development of SPM is continued: new economical modifications are created. The medium under simulation becomes more complex. Since elementary kinetic acts are simulated, as a rule, as imitations, SPM are easily generalized to multi-component mixtures and molecular gases with internal degrees of freedom. Chemical reactions, dissociation and recombination, as well as other kinetic processes, are

taken into account in the same, relatively easy, way. In this case, however, the requirements to the computational resources increase, although they remain within the limits of capabilities of modern super computers. The results obtained in recent years on the basis of SPM in the dynamics of rarefied gas can be found in the materials of international symposia on RGD.

In addition to the traditional range of RGD problems, SPM is used successfully also in other areas of mechanics. In particular, their extrapolation to the limit $Kn \rightarrow 0$, which corresponds to the ideal gas model, was natural [31]. It is known [113] that ideal gas flows under approximation by Euler gas dynamics equations (see Chapter 3) are described at the molecular level by the local Maxwell DF (15).

It is assumed in the algorithm proposed in [31] that DF in each Eulerian cell has the form (15). It is necessary to calculate the gas dynamic quantities ρ , \mathbf{U} , and T , which change from cell to cell. The general SPM scheme (see s.p. 4.3) is modified as follows. The realization of the Lagrangian stage remains almost unchanged. A purely mirror reflection is specified on impermeable surfaces, which corresponds to $\alpha = 1$ in (26). In this way, the macroscopic impenetrability condition is simulated at the molecular level. In this case, it allows to avoid the formation of boundary layers. The velocity values of model particles in Eulerian cells are not stored. Instead of this, the total number of particles N_j^{p+1} , the total momentum

$$\mathbf{P}_j^{p+1} = m \sum_{i=1}^{N_j} \mathbf{v}_i^{p+1}$$

and the total energy

$$\begin{aligned} E_j^{p+1} &= \frac{m}{2} \sum_{i=1}^{N_j} (\mathbf{v}_i^{p+1} - \mathbf{U}_j^{p+1})^2, \\ \mathbf{U}_j^{p+1} &= \frac{\mathbf{P}_j^{p+1}}{m N_j^{p+1}}. \end{aligned}$$

are summed up for each j -th cell. The local gas mass density ρ_j^{p+1} and the temperature T_j^{p+1} are determined from these quantities. The pressure is calculated from the state equation of an ideal gas

$$p = \rho R T$$

At the Eulerian step, the velocities of all particles are sampled by using the equilibrium DF with the parameters n_j^{p+1} , \mathbf{U}_j^{p+1} , and T_j^{p+1} , instead of simulating the sequence of collisions for each cell. The step τ is chosen so that the displacements of particles at this set of velocities are of the order of the typical spacing of the Eulerian cell.

The algorithm is conservative with respect to the mass, momentum, and energy fluxes. It does not have mechanisms of artificial dissipation. Actually, this modification of SPM approaches the Harlow method (Chapter 3).

Good results for problems with inviscid interaction of shock waves have been obtained with the use of this algorithm. The shock waves had a finite thickness. It is, however, less than the "smearing" of shock waves by finite-difference schemes. Notice that the algorithm makes it possible to consider boundaries of rather complex shapes. It is easily generalized to flows of gas mixtures, in particular, with equilibrium chemical kinetics. The operating speed of this method, however, is much lower than that of finite-difference schemes.

The simulation of coagulation processes in disperse systems is another application area of SPM. They are described by the Smolukhovsky kinetic equation [124] as follows:

$$\frac{\partial f}{\partial t} + \mathbf{U} \frac{\partial f}{\partial \mathbf{r}} = \frac{1}{2} \int_0^v dv_1 \beta(v - v_1, v_1) \times \\ \times f(v - v_1) f(v_1) - f(v) \int_0^\infty dv_1 \beta(v, v_1) f(v_1).$$

Here $f(\mathbf{r}, v, t)$ is the DF of the disperse phase on the volumes (sizes) of particles, $\beta(v, v_1)$ is the cross-section of the coagulation process, and \mathbf{U} is the given velocity field of the carrier phase. The structure of the equation is close to that of the Boltzmann equation (1). The convective operator differs in that the transport is realized by the external velocity field \mathbf{U} . The structure of the collision operator reflects a binary collision of particles. Here, however, the act of interaction consists in the merging (coalescence) of two particles with the formation of one particle with the total volume. In this case, there appears a factor that is new for SPM, which is associated with the change in the number of particles at the Eulerian step. Another difficulty is that there is not the M -equation similar to (20) for rarefied gas in the kinetic theory of disperse systems [124].

An approach that eliminates these difficulties and is effective from point of view of computations was proposed in [125]. An additional variable, $w \in [0, 1]$, is introduced for model particles. It is called the particle weight. With its help, the formal interaction of particles is determined by the scheme

$$(v_i, w_i) + (v_j, w_j) \rightarrow (v_i + v_j, w_i) + (v_j, w_j - w_i),$$

$$w_j > w_i.$$

Here, the number of particles remains unchanged, although the weights of formally conserved particles decrease monotonically. This approach makes it possible to describe the process in the cell by the Chapman-Kolmogorov equation for systems with binary interaction [34], which is similar to the M -equation (20).

As a result, we obtain the algorithm for the processing of the Eulerian cell, which is similar to that described in s.p. 4.3. Then, various approximate methods can be used to increase its effectiveness.

The Smolukhovsky equation is rather universal. In particular, it is used to describe many processes in the atmosphere. Here refer the formation of clouds and fogs, as well as phenomena associated with atmospheric electricity, precipitation, and transfer of natural and anthropogenic aerosols. Thus, for SPM there open up possibilities of application to problems of meteorology, climate, and ecology.

In [32], it was proposed to use SPM to simulate hydrodynamic turbulence. The Lundgren kinetic equation [126] for the DF of turbulent pulsations, $f(\mathbf{r}, \mathbf{v}', t)$, was considered as a basis. This equation has the form of the BGK-equation with a collision term of the form (14). The M -equation of the following form was used to describe the process in the Eulerian cell:

$$\frac{\partial F_N(\tilde{\mathbf{U}}), t}{\partial t} = \nu \left(\frac{1}{S_N} - F_N(\tilde{\mathbf{U}}, t) \right).$$

Here $\tilde{\mathbf{U}}$ is a $3N$ -dimensional vector of pulsation velocities, S_N is the surface of a $3N$ -dimensional sphere of constant energy of turbulent pulsations. The corresponding term in the right-hand side has the meaning of the uniform density distribution on it (see (15)).

In this equation, the processing of the sample $\tilde{\mathbf{U}}$ in the Eulerian cell becomes much simpler. The time interval $\tau_k < \tau$ before the interaction is simulated with the use of the distribution (29). Then a new vector, $\tilde{\mathbf{U}}$, is sampled by using the uniform distribution on S_N . Since further transitions on S_N are equiprobable, fractional steps are not used. The process in the cell is completed. It is seen that the use of the Lundgren equation has made it possible to avoid the determination of the physical individuality of turbulent particles ("moles") and the concrete definition of their interaction law.

Until the method has not been widely used. This is most likely explained by the fact that the Lundgren equation is a rather crude approximation for the description of the distribution of turbulent pulsations.

Supplements

The programs of some universal blocks, which are widely used in the practice of particle-in-cell methods, are presented here. They are written in FORTRAN-77, and can be used directly. They can also be used in the generation of more complex programs based on the particle methods.

A. Subroutine of initial data preparation

In the particles method for collisionless plasma, mesh functions are usually specified on a mesh at the initial time. Besides, particles must be distributed in cells, with initial velocities assigned to them in accordance with the initial conditions for the particles distribution function.

The preparation algorithm of the initial distribution of model particles consists in the following. Let the calculation domain in Cartesian coordinates (x, y) have the form of a rectangle $0 \leq x \leq x_{max}$, $0 \leq y \leq y_{max}$. At the initial time, the domain is filled with a substance of density $\rho_0(x, y)$ and temperature $T = \text{sigma}$. A certain number of particles of the same mass is placed in each cell. The number of the particles is proportional to the density in this cell. The mass of a particle is equal to the mass of the substance in the domain divided by the total number of model particles. The velocity of a particle is taken equal to the sum of a mean velocity of the substance at the point in which the particle is placed and the random value of the thermal velocity corresponding to the substance temperature at the same point. Two external functions, `rand` and `rand2`, are used in the subroutine [114]. The first function generates random numbers uniformly distributed in the interval $[0, 1]$. The other function gives random numbers with a normal distribution with the parameters $[0, \text{sigma}]$.

The following variables are used in the subroutine:

- `jm` — total number of model particles;
- `x(jm)`, `y(jm)` — arrays of the coordinates of particles;
- `vx(jm)`, `vy(jm)`, `vz(jm)` — arrays of the velocity components of particles;
- `ma(jm)` — array of masses of model particles;
- `xm`, `ym` — dimensions of the calculation domain on the x and y coordinates;
- `rho(im+1,km+1)` — density of the substance given at mesh nodes;
- `ux(im+1,km+1),uy(im+1,km+1),uz(im+1,km+1)` — velocity components of the substance;

sigma — dispersion of the particle velocity distribution;
n — the number of model particles in a cell;
im, km — the number of cells on the *x* and *y* coordinates.

```

subroutine start(xm,ym,vm,sigma,n,im,km,jm,
*                  rand,rand2,rho,ux,uy,uz,x,y,vx,vy,vz,ma)
  real*8 x(jm),y(jm),vx(jm),vy(jm),vz(jm),ma(jm)
  real*8 rho(im+1,km+1),ux(im+1,km+1),
*uy(im+1,km+1),uz(im+1,km+1)
  integer im,km,j,n,i,k,j1
  real*8 xm,ym,vm,hx,hy,sigma,s,s1,s2
  hx=xm/im
  hy=ym/km
c
c — Distribution of particles —
c
j=0
do 1 i=1,im
  do 1 k=1,km
    do 1 j1=1,n
      j=j+1
      s1=hx*rand
      s2=hy*rand
      x(j)=(i-1.)*hx+s1
      y(j)=(k-1.)*hy+s2
      s=(1.-s1)*((1.-s2)*ux(i,k)+s2*ux(i,k+1))+*
      *s1*((1.-s2)*ux(i+1,k)+s2*ux(i+1,k+1))
      vx(j)=s+rand2(sigma)
      s=(1.-s1)*((1.-s2)*uy(i,k)+s2*uy(i,k+1))+*
      *s1*((1.-s2)*uy(i+1,k)+s2*uy(i+1,k+1))
      vy(j)=s+rand2(sigma)
      s=(1.-s1)*((1.-s2)*uz(i,k)+s2*uz(i,k+1))+*
      *s1*((1.-s2)*uz(i+1,k)+s2*uz(i+1,k+1))
      vz(j)=s+rand2(sigma)
      s=(1.-s1)*((1.-s2)*rho(i,k)+s2*rho(i,k+1))+*
      *s1*((1.-s2)*rho(i+1,k)+s2*rho(i+1,k+1))
      ma(j)=s*hx*hy/n
1 continue
  return
end

```

B. Subroutines of interpolation between the Lagrangian and Eulerian meshes

The subroutines of interpolation between the Lagrangian and Eulerian meshes typical for particle-in-cell methods are presented in this supplement. A rectangular mesh domain with regular nodes $\{x_i = h_x i, y_k = h_y k, i = 0, \dots, I, k = 0, \dots, K\}$ is considered. Here h_x and h_y denote the mesh steps along the x - and y -axes.

B1. Interpolation of the mesh vector-function to the Lagrangian mesh of particles

The present subroutines realize the interpolation scheme with a mesh kernel of the type (1.32) generalized to the two-dimensional case (see [22]). Values of the mesh vector-function \mathbf{F}_h (such as velocity, force, and momentum) are interpolated to the location of the particle centered in (x, y) -point.

The case of integer nodes $\mathbf{F}_{i,k} = \mathbf{F}(x_i, y_k)$.

```
function inpol1(x,y,hx,hy,f,im,km)
real f(im,km),x,y,hx,hy,s1,s2
integer im,km,i,k
s1=x/hx
i=int(s1+1)
s1=s1-i+1
s2=y/hy
k=int(s2+1)
s2=s2-k+1
inpol1=(1.-s1)*((1.-s2)*f(i,k)+s2*f(i,k+1))+*
s1*((1.-s2)*f(i+1,k)+s2*f(i+1,k+1))
return
end
```

The case of half-integer nodes (centers of cells) $\mathbf{F}_{i-1/2,k-1/2} = \mathbf{F}(h_x(i-1/2), h_y(k-1/2))$.

```
function inpol2(x,y,hx,hy,f,im,km)
real f(im,km),x,y,hx,hy,s1,s2
integer im,km,i,k
s1=x/hx
i=int(s1+1.5)
s1=s1-i+1.5
s2=y/hy
k=int(s2+1.5)
```

```

s2=s2-k+1.5
inpol1=(1.-s1)*((1.-s2)*f(i,k)+s2*f(i,k+1))+  

* s1*((1.-s2)*f(i+1,k)+s2*f(i+1,k+1))
return
end

```

B2. Interpolation of the scalar function from the Lagrangian mesh of particles to nodes of the Eulerian mesh

The subroutine realizes the interpolation scheme for particles with the kernel of the type (1.29) generalized to the two-dimensional case ([22]). The values of the grid function ρ (such as charge or mass density) are calculated with the use of the relevant scalar features of individual particles located at the points (x_j, y_j) of the calculation domain.

The Neumann boundary condition or periodicity conditions can be specified at the boundary nodes of the mesh domain. The type of the boundary conditions is determined by the values of the parameters $ng1$ and $ng2$.

$$ng1 = \begin{cases} 1, & \frac{\partial \rho(0, y_k)}{\partial x} = \frac{\partial \rho(x_I, y_k)}{\partial x} = 0, \\ 2, & \rho(0, y_k) = \rho(x_I, y_k). \end{cases}$$

$$ng2 = \begin{cases} 1, & \frac{\partial \rho(x_i, 0)}{\partial y} = \frac{\partial \rho(x_i, y_K)}{\partial y} = 0, \\ 2, & \rho(x_i, 0) = \rho(x_i, y_K). \end{cases}$$

The case of integer nodes $\rho_{i,k} = \rho(x_i, y_k)$.

```

subroutine density1(x,y,mas,jm,ro,im,km,hx,hy,ng1,ng2)
integer jm,im,km,ng1,ng2,j,i,k
real x(jm),y(jm),mas(jm)
real ro(im+1,km+1)
real hx,hy,s1,s2
do 1 i=1,im+1
do 1 k=1,km+1
ro(i,k)=0.
1 continue
do 2 j=1,jm
s1=x(j)/hx
i=int(s1+1)
s1=s1-i+1
s2=y(j)/hy
k=int(s2+1)

```

```

s2=s2-k+1
ro(i,k)=ro(i,k)+mas(j)*(1.-s1)*(1.-s2)
ro(i,k+1)=ro(i,k+1)+mas(j)*(1.-s1)*s2
ro(i+1,k)=ro(i+1,k)+mas(j)*s1*(1.-s2)
ro(i+1,k+1)=ro(i+1,k+1)+mas(j)*s1*s2
2 continue
  if(ng1.eq.1) then
    do 3 k=1,km+1
      ro(1,k)=2.*ro(1,k)
      ro(im+1,k)=2.*ro(im+1,k)
3 continue
  endif
  if(ng1.eq.2) then
    do 4 k=1,km+1
      ro(1,k)=ro(1,k)+ro(im+1,k)
      ro(im+1,k)=ro(1,k)
4 continue
  endif
  if(ng2.eq.1) then
    do 5 i=1,im+1
      ro(i,1)=2.*ro(i,1)
      ro(i,km+1)=2.*ro(i,km+1)
5 continue
  endif
  if(ng2.eq.2) then
    do 6 i=1,im+1
      ro(i,1)=ro(i,1)+ro(i,km+1)
      ro(i,km+1)=ro(i,1)
6 continue
  endif
  do 7 i=1,im+1
    do 7 k=1,km+1
      ro(i,k)=ro(i,k)/(hx*hy)
7 continue
  return
end

```

The case of half-integer nodes (centers of cells) $\rho_{i-1/2,k-1/2} = \rho(h_x(i-1/2), h_y(k-1/2))$.

```

subroutine density2(x,y,mas,jm,ro,im,km,hx,hy,ng1,ng2)
integer jm,im,km,ng1,ng2,j,i,k
real x(jm),y(jm),mas(jm)
real ro(im+2,km+2)
real hx,hy,s1,s2

```

```

do 1 i=1,im+2
do 1 k=1,km+2
ro(i,k)=0.
1 continue
do 2 j=1,jm
s1=x(j)/hx
i=int(s1+1.5)
s1=s1-i+1.5
s2=y(j)/hy
k=int(s2+1.5)
s2=s2-k+1.5
ro(i,k)=ro(i,k)+mas(j)*(1.-s1)*(1.-s2)
ro(i,k+1)=ro(i,k+1)+mas(j)*(1.-s1)*s2
ro(i+1,k)=ro(i+1,k)+mas(j)*s1*(1.-s2)
ro(i+1,k+1)=ro(i+1,k+1)+mas(j)*s1*s2
2 continue
if(ng1.eq.1) then
do 3 k=1,km+2
ro(2,k)=ro(1,k)+ro(2,k)
ro(1,k)=ro(1,k)
ro(im+1,k)=ro(im+1,k)+ro(im+2,k)
ro(im+2,k)=ro(im+1,k)
3 continue
endif
if(ng1.eq.2) then
do 4 k=1,km+2
ro(2,k)=ro(2,k)+ro(im+2,k)
ro(im+2,k)=ro(2,k)
ro(im+1,k)=ro(im+1,k)+ro(1,k)
ro(1,k)=ro(im+1,k)
4 continue
endif
if(ng2.eq.1) then
do 5 i=1,im+2
ro(i,2)=ro(i,1)+ro(i,2)
ro(i,1)=ro(i,2)
ro(i,km+1)=ro(i,km+1)+ro(i,km+2)
ro(i,km+2)=ro(i,km+1)
5 continue
endif
if(ng2.eq.2) then
do 6 i=1,im+2
ro(i,2)=ro(i,2)+ro(i,km+2)
ro(i,km+2)=ro(i,2)
ro(i,km+1)=ro(i,km+1)+ro(i,1)
ro(i,1)=ro(i,km+1)

```

```

6 continue
endif
do 7 i=1,im+2
do 7 k=1,km+2
ro(i,k)=ro(i,k)/(hx*hy)
7 continue
return
end

```

B3. The subroutine of interpolation of generalized fields to the particle location on unstructured grids

The following variables are used in the subroutine:

Input:

melem — maximum number of elements
mpoin — maximum number of nodes
ndim — $ndim = 2$ or $ndim = 3$ – the space dimension
grid(mpoin,ndim) — array of coordinates of each vertex
ntetra(melem,ndim+1) — array of numbers of vertices of each element
e1nod,e2nod,e3nod(mpoin) — arrays of the electric fields in the nodes
b1nod,b2nod,b3nod(mpoin) — arrays of the magnetic fields in the nodes
xj — x-coordinate of the particle with number j
yj — y-coordinate of the particle with number j
nposj — the number of triangle in which particle with number j is located

Output:

sh(ndim+1) — Array of the shape functions of the triangle in which the particle with number j is located
e1pp, e2pp, e3pp — electric fields in the particle position
b1pp, b2pp, b3pp — magnetic fields in the particle position

```

subroutine pipol(melem,mpoin,ndim,ntetra,grid,
*                  e1nod,e2nod,e3nod,b1nod,b2nod,b3nod,
*                  xj,yj,nposj,sh,e1pp,e2pp,e3pp,b1pp,b2pp,b3pp,delta)
implicit none
integer melem,ndim,mpoin
integer nposj,i,ntetra(melem,ndim+1)
real*8 grid(mpoin,ndim),cbace(melem,ndim)
real*8 e1nod(mpoin),e2nod(mpoin),e3nod(mpoin)

```

```

real*8 b1nod(mpoin),b2nod(mpoin),b3nod(mpoin)
real*8 xj,yj
real*8 sh(ndim+1)
real*8 e1pp,e2pp,e3pp
real*8 b1pp,b2pp,b3pp
call shap2d(melem,mpoin,ndim,ntetra,grid,
*x           xj,yj,nposj,sh)
e1pp=0.
do i=1,ndim+1
e1pp=e1pp + e1nod(ntetra(nposj,i))*sh(i)
enddo
e2pp=0.
do i=1,ndim+1
e2pp=e2pp + e2nod(ntetra(nposj,i))*sh(i)
enddo
e3pp=0.
do i=1,ndim+1
e3pp=e3pp + e3nod(ntetra(nposj,i))*sh(i)
enddo
b1pp=0.
do i=1,ndim+1
b1pp=b1pp + b1nod(ntetra(nposj,i))*sh(i)
enddo
b2pp=0.
do i=1,ndim+1
b2pp=b2pp + b2nod(ntetra(nposj,i))*sh(i)
enddo
b3pp=0.
do i=1,ndim+1
b3pp=b3pp + b3nod(ntetra(nposj,i))*sh(i)
enddo
return
end

```

B4. The subroutine for assignment of the particle charge on unstructured grids

The following variables are used in the subroutine:

Input:

melem — maximum number of elements

ntetra(melem,ndim+1) — array of numbers of vertices of each element

nposj — the number of an element in which particle is located

ndim — $\text{ndim}=2$ or $\text{ndim}=3$

mptyp — maximum number of particle types

npt — type of the particle

sh(ndim+1) — the array of shape functions of the particle with index

j

weight — the weight of the particle

Output:

array(mpoin,mptyp) — the array of the "effective" charge or current density in the vertices of elements

```

subroutine lgssd (array,mpoin,melem,ntetra,nposj,
*                  ndim,mptyp,npt,sh,weight)
implicit none
integer mpoin,melem,ndim,npt,mptyp,nposj
integer ntetra(melem,ndim+1),ki(ndim+1)
integer i
real*8 sh(ndim+1),weight
real*8 array(mpoin,mptyp)
do i=1,ndim+1
  ki(i)=ntetra(nposj,i)
  array(ki(i),npt) = array(ki(i),npt) + sh(i)*weight
end do
return
end

```

B5. The subroutine for the determination of the scalar density in the nodes of unstructured grids

The following variables are used in the subroutine:

Input:

melem — maximum number of elements

nelem — real number of elements

mpoin — maximum number of nodes

npoin — real number of nodes

ndim — $\text{ndim}=2$, or $\text{ndim}=3$

mptyp — maximum number of particle types

nptyp — real number of particle types

ntetra(melem,ndim+1) — array of numbers of vertices of each element

voli(mpoin) — The array of "inverse" volumes

Output:

array(mpoin,mptyp) — The array of the charge or current density

```

subroutine lgssd1(melem,nelem,mpoin,npoin,ndim,
*                  mptyp,nptyp,ntetra,voli,array)
implicit none
integer mpoin,npoin,melem,nelem
integer ndim,mptyp,nptyp
integer ntetra(melem,ndim+1),ki(ndim+1)
integer i,l,npt
real*8 voli(mpoin)
real*8 array(mpoin,mptyp)

c
c Determination charge density in the nodes
c
do npt=1,nptyp
do i=1,npoin
array(i,npt)=array(i,npt)*voli(i)
enddo
enddo
return
end

```

C. Subroutine for the particle dynamics

C1. Subroutine for calculation of the particles dynamics in fields of mass forces

The subroutine realizes the difference scheme of form (5.16) for the dynamics of particles in the force (electric) field in a two-dimensional rectangular domain. The calculation is limited by only one time step. Simultaneously, the mesh density of the scalar feature (charge) carried by particles is calculated. Periodic boundary conditions are set at the boundary of the domain for the density (charge).

The following variables are used in the subroutine:

xm — size of the solution domain in the *x*-axis;
ym — size of the solution domain in the *y*-axis;
h1 — grid spacing along the *x*-axis;
h2 — grid spacing along the *y*-axis;
tau — time step;
alp — ratio of ion and electron masses;
ex(im,kmp-1) — array of *x*-components of electric field (force);

ey(imp-1,kmp) — array of *y*-components of electric field (force) intensity;

x(jmp) — array of *x*-coordinates of particles;

y(jmp) — array of *y*-coordinates of particles;

vx(jmp) — array of *x*-components of particle velocities;

vy(jmp) — array of *y*-components of particle velocities;

e(jmp) — array of scalar features (charges) of particles;

rr(jmp) — array of density (charge).

```

subroutine move2(xm,ym,h1,h2,tau,alp,
*                  ex,ey,x,y,vx,vy,e,rr)
integer imp,kmp,jmp,im1,km1,i,k,i1,k1,j
real*8 xm,ym,h1,h2,tau,alp,tau1,tau2,tau3,
* s1,s2,s3,s4,sx,sy,u,v,x1,y1
parameter(imp=18)
parameter(kmp=18)
parameter(jmp=6000)
real*8 ex(imp,kmp-1),ey(imp-1,kmp),rr(imp,kmp)
real*8 x(jmp),y(jmp),vx(jmp),vy(jmp),e(jmp)
tau2=-tau*0.5
tau3=tau*0.5/alp
im1=imp-1
km1=kmp-1
do 1 i=1,imp
do 1 k=1,kmp
rr(i,k)=0.
1 continue
do 2 j=1,jmp
if(e(j).gt.0) then
tau1=tau3
else
tau1=tau2
endif
c
c - Interpolation of mesh intensity
c - of the electric field to the particle location
c
s2=x(j)/h1
i=int(s2+1.)
i1=int(s2+1.5)
s1=i-s2
s2=i1-0.5-s2
s4=y(j)/h2
k=int(s4+1.)
k1=int(s4+1.5)

```

```

s3=k-s4
s4=k1-0.5-s4
sx=tau1*(s1*(s4*ex(i,k1)+(1.-s4)*ex(i,k1+1))+  

+(1.-s1)*(s4*ex(i+1,k1)+(1.-s4)*ex(i+1,k1+1)))
sy=tau1*(s2*(s3*ey(i1,k)+(1.-s3)*ey(i1,k+1))+  

+(1.-s2)*(s3*ey(i1+1,k)+(1.-s3)*ey(i1+1,k+1)))
c
c - Calculation of new velocities and coordinates of particles
c
u=vx(j)+sx
v=vy(j)+sy
x1=x+tau*u
y1=y+tau*v
c
c - Test: is the particle left the domain
c - and it is outside the domain boundaries
c
if(x1.lt.0.) x1=x1+xm
if(x1.gt.xm) x1=x1-xm
if(y1.lt.0.) y1=y1+ym
if(y1.gt.ym) y1=y1-ym
c
c - calculation of the charge density
c - by using new coordinates of particles
c
s2=x1/h1
i=int(s2+1.5)
s2=s2-i+1.5
s4=y1/h2
k=int(s4+1.5)
s4=s4-k+1.5
rr(i,k)=rr(i,k)+(1.-s2)*(1.-s4)*a1
rr(i+1,k)=rr(i+1,k)+s2*(1.-s4)*a1
rr(i,k+1)=rr(i,k+1)+(1.-s2)*s4*a1
rr(i+1,k+1)=rr(i+1,k+1)+s2*s4*a1
x(j)=x1
y(j)=y1
vx(j)=u
vy(j)=v
2 continue
c
c - Periodic boundary conditions for the density
c
do 3 k=1,kmp
rr(2,k)=rr(2,k)+rr(kmp,k)
rr(im1,k)=rr(im1,k)+rr(1,k)

```

```

rr(1,k)=rr(im1,k)
rr(imp,k)=rr(2,k)
3 continue
do 4 i=1,imp
rr(i,2)=rr(i,2)+rr(i,kmp)
rr(i,km1)=rr(i,km1)+rr(i,1)
rr(i,1)=rr(i,km1)
rr(i,kmp)=rr(i,2)
4 continue
return
end

```

C2. The subroutine for relativistic particle pusher according to Boris [79]

The following variables are used in the subroutine:

Input:

u — particle momentum in x-direction
v — particle momentum in y-direction
w — particle momentum in z-direction
x — x-coordinate of particle
y — y-coordinate of particle
z — z-coordinate of particle
qpart — charge of particle in the units of elementary charge (electron charge=-1)
pmass — mass of particle in the units of electron mass
exj — electric field in x-direction
eyj — electric field in y-direction
ezj — electric field in z-direction
bxj — magnetic field in x-direction
byj — magnetic field in y-direction
bzj — magnetic field in z-direction
t — time step

Output:

x,y,z — on the new time level
u,v,w — on the new time level

```

subroutine ppushr(u,v,w,x,y,z,qpart,pmass,
*                  exj,eyj,ezj,bxj,byj,bzj,t)
implicit none
real*8 qpart,pmass,exj,eyj,ezj,bxj,byj,bzj,u,v,w,x,y,z,t
real*8 th,thq,tx,ty,tz,

```

```

*ts,ts2,sx,sy,sz,up,vp,wp,
*umi,vmi,wmi,ua,va,wa,gith,git
thq=0.5*t*qpart
umi=u+exj*thq
vmi=v+eyj*thq
wmi=w+ezj*thq
gith=thq/sqrt(pmass**2+umi**2+vmi**2+wmi**2)
tx=bxj*gith
ty=byj*gith
tz=bzj*gith
ts=tx**2+ty**2+tz**2
ts2=2/(1.0+ts)
sx=tx*ts2
sy=ty*ts2
sz=tz*ts2
up=umi+vmi*tz-wmi*ty
vp=vmi+wmi*tx-umi*tz
wp=wmi+umi*ty-vmi*tx
ua=umi+vp*sz-wp*sy
va=vmi+wp*sx-up*sz
wa=wmi+up*sy-vp*sx
u=ua+exj*thq
v=va+eyj*thq
w=wa+ezj*thq
git=t/sqrt(pmass**2+u**2+v**2+w**2)
x=x+u*git
y=y+v*git
z=z+w*git
return
end

```

D. The subroutines of a localization of particles on the unstructured grid

The following variables are used in the subroutine:

Input:

mpoin — maximum number of vertices
melem — maximum number of elements
ndim — **ndim**=2
grid(mpoin,ndim) — array of coordinates of each vertex
ntetra(melem,ndim+1) — array of numbers of vertices of each element
nate(melem,ndim+1) — array of numbers of adjacent triangles

xj — x-coordinate of the particle with number j
yj — y-coordinate of the particle with number j

Output:

nposj — the number of triangle in which particle with number j is located
sh(ndim+1) — Array of the shape functions of triangle in which the particle with number j is located

D1. The subroutines of particle localization on two-dimensional triangular grid (Löhner's algorithm [55])

```

subroutine plocg2(mpoin,melem,ndim,xj,yj,
*                  nposj,sh,grid,ntetra,nate)
implicit none
integer melem,mpoin,ndim
integer nposj,ntetra(melem,ndim+1),nate(melem,ndim+1)
integer i,j,k1,nc,index
real*8 grid(mpoin,ndim)
real*8 xj,yj
real*8 sh(ndim+1)
real*8 dish,vmin
4 continue
c
c   If nposj=0 the particle is outside the computational
c   domain. In this case we must be use of the boundary conditions
c
if(nposj.eq.0) then
  return
endif
nc=nposj
index=1
c
c   call subroutine for determination of shape-functions
c
call shap2d(melem,mpoin,ndim,ntetra,grid,xj,yj,nposj,sh)
vmin=0.0 d-0
do i=1,ndim+1
  dish=1-sh(i)
  if(dmin1(sh(i),dish).lt.-1.0 d-15) then
    index=-1
  endif
  if(sh(i).le.vmin) then
    vmin=sh(i)
  endif
enddo

```

```

k1=i
endif
enddo
if(index.eq.1) then
go to 6
endif
if(index.eq.-1) then

nposj=nate(nc,k1)
go to 4
end if
6 nposj=nc
5 continue
return
end

```

D2. The subroutines of particle localization on three-dimensional tetrahedrons grid (Assous algorithm [49])

The following variables are used in the subroutine:

Input:

mpart — maximum number of particles
npart — real number of particles
mpoin — maximum number of vertices
melem — maximum number of elements
grid(mpoin,3) — array of coordinates of each vertex
ntetra(melem,4) — array of numbers of vertices of each element
nate(melem,4) — array of numbers of adjacent tetrahedrons
x(mpart) — array of x-coordinates of particles
y(mpart) — array of y-coordinates of particles
z(mpart) — array of z-coordinates of particles
u — particle momentum on x-axis
v — particle momentum on y-axis
w — particle momentum on z-axis

Note:

In the non-relativistic case u,v,w are the velocities of the particles.

Output:

npos(mpart) — array of numbers of tetrahedrons in which particles

are located

```

subroutine plocg3(mpart,npart,mpoin,melem,
*                  x,y,z,u,v,w,npos,grid,ntetra,
*                  nate,vec12,vec32,vec42,numk2)
implicit none
integer mpart,npart,melem,mpoin
integer j,i,k1,k2,k3,k4,nc
integer nsumt,itet,index
integer ntetra(melem,4),nate(melem,4)
integer npos(mpart),numk2(4),ntmin(4)
real*8 x(mpart),y(mpart),z(mpart),
* u(mpart),v(mpart),w(mpart)
real*8 vec12(melem,4,3),vec32(melem,4,3),
* vec42(melem,4,3)
real*8 grid(mpoin,3)
real*8 ve32(3),ve42(3),pok2(3)
real*8 ves(3),det(4)
real*8 sc1,sc2
do 5 j=1,npart
6 continue
  nc=npos(j)
c
c  If npos(j)=0 the particle is outside the computational
c  domain. In this case we must be
c  use of the boundary conditions.
c
      if(npos(j).eq.0) go to 5
      k1=1
      k2=2
      k3=3
      k4=4
      nsumt=0
c
c  nsumt — variable that determines the number
c  of the negative determinants
c
      do i=1,4
        ntmin(i)=0
      enddo
c
c  ntmin(i) — integer array that determines the numbers
c  of the negative determinants
c
      itet=0
      do k1=1,4

```

```

c
c   calculation of det(k1), k1=1,2,3,4
c
c   ves(1)=vec32(nc,k1,2)*vec42(nc,k1,3)-
* vec32(nc,k1,3)*vec42(nc,k1,2)
  ves(2)=vec32(nc,k1,3)*vec42(nc,k1,1)-
* vec32(nc,k1,1)*vec42(nc,k1,3)
  ves(3)=vec32(nc,k1,1)*vec42(nc,k1,2)-
* vec32(nc,k1,2)*vec42(nc,k1,1)
  sc1=vec12(nc,k1,1)*ves(1)+  

* vec12(nc,k1,2)*ves(2)+  

* vec12(nc,k1,3)*ves(3)
  sc2=(x(j)-grid(ntetra(nc,k2),1))*ves(1)+  

* (y(j)-grid(ntetra(nc,k2),2))*ves(2)+  

* (z(j)-grid(ntetra(nc,k2),3))*ves(3)
  det(k1)=sc1*sc2

c
c   det(k1) — array of values of determinants
c
c   if(det(k1).le.-1.0 d-15) then
  nsumt=nsumt+1
  itet=itet+1
  ntmin(itet)=k1
  end if
  k2=k3
  k3=k4
  k4=k1
  enddo
  if(nsumt.eq.0) then
  go to 5
  endif

c
c   determination of the face (k2,k3,k4) which
c   is intersected by the particle trajectory
c
c   if(nsumt.eq.1) then
c
c   if nsumt=1 then only one det(k1) < 0
c
c   k1=ntmin(1)
  endif
  if(nsumt.eq.2) then
c
c   if nsumt=2 than two det(k1) < 0
c
  k1=ntmin(1)

```

```

k2=numk2(k1)
call equal(mpoin,melem,grid,ntetra,
*           vec12,vec32,vec42,nc,k1,k2,
*           ve32,ve42,pok2)
call cross(ve32,ve42,pok2,x(j),y(j),z(j),
*           u(j),v(j),w(j),index)
if(index.eq.-1) then
k1=ntmin(2)
endif
endif
if(nsumt.eq.3) then
c
c   if nsumt=3 than three det(k1) < 0
c
k1=ntmin(1)
k2=numk2(k1)
call equal(mpoin,melem,grid,ntetra,
*           vec12,vec32,vec42,nc,k1,k2,
*           ve32,ve42,pok2)
call cross(ve32,ve42,pok2,x(j),y(j),z(j),
*           u(j),v(j),w(j),index)
if(index.eq.-1) then
k1=ntmin(2)
k2=numk2(k1)
call equal(mpoin,melem,grid,ntetra,
*           vec12,vec32,vec42,nc,k1,k2,
*           ve32,ve42,pok2)
call cross(ve32,ve42,pok2,x(j),y(j),z(j),
*           u(j),v(j),w(j),index)
if(index.eq.-1) then
k1=ntmin(3)
endif
endif
npos(j)=nate(nc,k1)
go to 6
5 continue
return
end

```

E. The subroutines for calculation of linear shape-functions on unstructured grids

E1. The subroutine for calculation shape-functions with respect to the particle locations in two-dimensional case

The following variables are used in the subroutine:

Input:

melem — maximum number of elements
mpoin — maximum number of nodes
ndim — **ndim** = 2
grid(mpoin,ndim) — array of coordinates of each vertex
ntetra(melem,ndim+1) — array of numbers of the vertices of each element

Output:

sh(ndim+1) — Array of the shape functions of the triangle in which the particle with number j is located

```

subroutine shap2d(melem,mpoin,ndim,ntetra,
*                  grid,xj,yj,nposj,sh)
implicit none
integer melem,ndim,mpoin
integer nposj,ntetra(melem,ndim+1)
integer k,i
real*8 grid(mpoin,ndim)
real*8 xj,yj
real*8 a(ndim+1,ndim+1),sh(ndim+1)
real*8 deti
do i=1,ndim+1
do k=1,ndim+1
a(i,k)=grid(ntetra(nposj,k),i)-grid(ntetra(nposj,3),i)
enddo
enddo
a(1,3)=xj-grid(ntetra(nposj,3),1)
a(2,3)=yj-grid(ntetra(nposj,3),2)

c
c determination of the shape-functions of triangles
c with respect to the particle location
c
deti=1.0 d-0/(a(1,1)*a(2,2)-a(1,2)*a(2,1))
sh(1)=(a(1,3)*a(2,2)-a(1,2)*a(2,3))*deti
sh(2)=(a(1,1)*a(2,3)-a(1,3)*a(2,1))*deti
sh(3)=1.0 d-0-sh(1)-sh(2)

```

```
return
end
```

E2. The subroutine of calculation of the shape-functions with respect to the particle locations for three-dimensional case

The following variables are used in the subroutine:

Input:

`melem` — maximum number of elements
`mpoin` — maximum number of nodes
`ndim` — `ndim = 3`
`grid(mpoin,ndim)` — array of coordinates of each vertex
`ntetra(melem,ndim+1)` — array of numbers of vertices of each element

Output:

`sh(ndim+1)` — Array of the shape-functions of tetrahedron in which the particle with number j is located

```
subroutine shap3d(melem,mpoin,ndim,
*                  ntetra,grid,xj,yj,zj,nposj,sh)
implicit none
integer melem,ndim,mpoin
integer nposj,ntetra(melem,ndim+1)
integer k,i
real*8 grid(mpoin,ndim)
real*8 xj,yj,zj
real*8 a(20,21),sh(20)
real*8 det,s
do i=1,ndim
do k=1,ndim
a(i,k)=grid(ntetra(nposj,k),i)-grid(ntetra(nposj,4),i)
enddo
enddo
c
c determination of the shape-functions of tetrahedrons
c with respect to the particle location
c
a(1,4)=xj-grid(ntetra(nposj,4),1)
a(2,4)=yj-grid(ntetra(nposj,4),2)
a(3,4)=zj-grid(ntetra(nposj,4),3)
call gauss(3,a,sh,s,det)
```

```

sh(4)=1-sh(1)-sh(2)-sh(3)
return
end

```

F. The auxiliary subroutines

F1. The subroutine of determination of the local coordinates of point r (r -vector)

This subroutine is used in subroutine ploc3.

```

subroutine cross(ve32,ve42,pok2,x,y,z,u,v,w,index)
implicit none
integer index
real*8 ve32(3),ve42(3),pok2(3)
real*8 x,y,z,u,v,w
real*8 sidel,sider
real*8 alf,beta
sidel=(w*ve32(2)-v*ve32(3))*(v*ve42(1)-u*ve42(2))-
*(v*ve32(1)-u*ve32(2))*(w*ve42(2)-v*ve42(3))
sider=(y-z-pok2(2)+pok2(3))*(v*ve42(1)-u*ve42(2))-
*(x-y-pok2(1)+pok2(2))*(w*ve42(2)-v*ve42(3))
alf=sider/sidel
sidel=v*ve42(1)-u*ve42(2)
sider=x-y-pok2(1)+pok2(2)-alf*(v*ve32(1)-u*ve32(2))
beta=sider/sidel
index=-1
if(alf.ge.0.and.beta.ge.0.
and.alf+beta.le.1.0 d-0) then
index=1
end if
return
end

```

F2. The subroutine for determination of auxiliary vectors of tetrahedrons with nodes (k1,k2,k3,k4)

This subroutine is used in subroutine ploc3.

```

subroutine vecaux(mpoin,melem,grid,ntetra,
*                  numk2,vec12,vec32,vec42)
implicit none

```

```

integer mpoin,melem
integer numk2(4),ntetra(melem,4)
integer i,k1,k2,k3,k4,n
real*8 vec12(melem,4,3),
* vec32(melem,4,3),vec42(melem,4,3)
real*8 grid(mpoin,3)
do 3 n=1,melem
k1=1
k2=2
k3=3
k4=4
2 continue
  do 4 i=1,3
    vec32(n,k1,i)=grid(ntetra(n,k3),i)-
* grid(ntetra(n,k2),i)
    vec42(n,k1,i)=grid(ntetra(n,k4),i)-
* grid(ntetra(n,k2),i)
    vec12(n,k1,i)=grid(ntetra(n,k1),i)-
* grid(ntetra(n,k2),i)
4 continue
  if(k1.eq.4) go to 3
  k2=k3
  k3=k4
  k4=k1
  k1=k1+1
  go to 2
3 continue
c
c   determination k2(k1)
c
  numk2(1)=2
  numk2(2)=3
  numk2(3)=4
  numk2(4)=1
  return
end

```

F3. The subroutine used in subroutine ploc3

```

subroutine equal(mpoin,melem,grid,ntetra,
*                 vec12,vec32,vec42,
*                 nc,k1,k2,ve32,ve42,pok2)
implicit real*8(a-h,o-z)
integer mpoin,melem

```

```

integer ntetra(melem,4)
integer i,nc,k1,k2
real*8 vec12(melem,4,3),
* vec32(melem,4,3),vec42(melem,4,3)
real*8 ve32(3),ve42(3),pok2(3)
real*8 grid(mpoin,3)
do i=1,3
ve32(i)=vec32(nc,k1,i)
ve42(i)=vec42(nc,k1,i)
pok2(i)=grid(ntetra(nc,k2),i)
enddo
return
end

```

F4. The subroutine for Gauss elimination

This subroutine is used in subroutine shap3d.

```

subroutine gauss(n,a,x,s,det)
implicit none
real*8 a(20,21),x(20)
real*8 det,r,s
integer n,n1,k,k1,i,j
n1=n+1
det=1
do 25 k=1,n
k1=k+1
s=a(k,k)
j=k
do 21 i=k1,n
r=a(i,k)
if(dabs(r).le.dabs(s)) go to 21
s=r
j=i
21 continue
if(s.eq.0) return
if(j.eq.k) go to 23
do 22 i=k,n1
r=a(k,i)
a(k,i)=a(j,i)
22 a(j,i)=r
23 do 24 j=k1,n1
24 a(k,j)=a(k,j)/s
do 25 i=k1,n

```

```

r=a(i,k)
do 25 j=k1,n1
25 a(i,j)=a(i,j)-a(k,j)*r
    do 28 i=1,n
28 det=det*a(i,i)
    x(n)=a(n,n1)
    do 27 i=n-1,1,-1
    s=a(i,n1)
    do 26 j=i+1,n
26 s=s-a(i,j)*x(j)
27 x(i)=s
      return
      end

```

G. Subroutines for the solution of the Poisson equation (Poisson solvers)

The subroutines realize two economical algorithms of numerical solution of the Poisson equation in Cartesian coordinates on the plane

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = \rho,$$

often used in a practice of particle methods. Here the function φ can be a scalar potential of the electric or gravitational field, a stream function, or a projection of the vector-potential of the magnetic field. Hence, the function ρ is the density of the electric charge, gravitational mass, the vorticity projection, or the current density projection onto normal to a calculation plane.

G1. Direct method

The Poisson equation in a rectangular mesh domain with a regular node grid $\{x_i = h_x i, y_k = h_y k, i = 0, \dots, I, k = 0, \dots, K\}$ is approximated by the following second-order scheme:

$$\frac{\varphi_{i+1,k} - 2\varphi_{i,k} + \varphi_{i-1,k}}{h_x^2} + \frac{\varphi_{i,k+1} - 2\varphi_{i,k} + \varphi_{i,k-1}}{h_y^2} = \rho_{i,k}$$

with the periodic boundary conditions $\{\varphi_{0,k} = \{\varphi_{I,k}, \varphi_{i,0} = \{\varphi_{i,K}\}\}$. A double Fourier transform realized by the numerical procedure of the fast Fourier transform [73] is used.

The following variables are used in the subroutine:

im — number of mesh nodes along the *x*-axis (im must be the integer power of 2);

km — number of mesh nodes along the *y*-axis (km must be the integer power of 2);

hx — grid step along the *x*-axis;

hy — grid step along the *y*-axis;

phi(im,km) — array of values for the mesh solution of the Poisson equation;

rho(im,km) — array of values of the grid right-hand side function of the Poisson equation;

pp(im,km),a(im),b(km) — operating arrays.

```

subroutine poisson1(nx,ny,im,km,hx,hy,
*                                rho,phi,pp,a,b)
real*8 phi(im,km),rho(im,km)
complex*16 pp(im,km),a(im),b(km)
integer i,k
real*8 hx,hy,pi,s
pi=3.14159265358979d0
do 3 k=1,km
do 1 i=1,im
a(i)=rho(i,k)
1 continue
call fftc(a,nx,1,im)
do 2 i=1,im
pp(i,k)=a(i)
2 continue
3 continue
do 6 i=1,im
do 4 k=1,km
b(k)=pp(i,k)
4 continue
call fftc(b,ny,1,km)
do 5 k=1,km
pp(i,k)=b(k)
5 continue
6 continue
do 7 i=1,im
do 7 k=1,km
if(i+k.eq.2) then
pp(1,1)=0.
else
s=(dsin(pi*(i-1)/im)/hx)**2+(dsin(pi*(k-1)/km)/hy)**2
pp(i,k)=-0.25d0*pp(i,k)/s
endif
7 continue

```

```

do 10 i=1,im
do 8 k=1,km
b(k)=pp(i,k)
8 continue
call fftc(b,ny,-1,km)
do 9 k=1,km
pp(i,k)=b(k)
9 continue
10 continue
do 13 k=1,km
do 11 i=1,im
a(i)=pp(i,k)
11 continue
call fftc(a,nx,-1,im)
do 12 i=1,im
phi(i,k)=real(a(i))
12 continue
13 continue
end
subroutine fftc(a,n,isi,np)
implicit real*8(a-h,o-z)
complex*16 a(np),t,w,w1
do 1 i=1,np
if(isi.gt.0) a(i)=a(i)/np
1 continue
pi2=8.d-0*datan(1.d-0)
nn=np
j=1
do 4 i=1,nn
if(i.ge.j) go to 2
t=a(j)
a(j)=a(i)
a(i)=t
2 m=nn/2
3 if(j.le.m) go to 4
j=j-m
m=m/2
if(m.ge.1) go to 3
4 j=j+m
mm=1
5 if(mm.ge.nn) return
ii=2*mm
th= pi2/isign(ii,n*isi)
w1=dcmplx(-2.0 d-0*dsin(th/2)**2,dsin(th))
w=1
do 7 m=1,mm

```

```

do 6 i=m,nn,ii
t=w*a(i+mm)
a(i+mm)=a(i)-t
a(i)=a(i)+t
6 continue
w=w1*w+w
7 continue
mm=ii
go to 5
end

```

G2. Combined iteration method

Here, the method of successive upper relaxation in the x -axis is combined with the sweep method along the y -axis. An advantage of such a combination is that iterations in the method are performed for one coordinate only. This accelerates considerably its convergence in comparison to the standard method of upper relaxation [22]. Here, we used the Dirichlet boundary conditions on all boundaries. The value of the parameter `eps` determines the accuracy of the solution obtained, and provides the condition of termination of iterations. The iteration parameter ω (in the program called `omega`) can be within $0 < \omega < 2$. Its optimal value is determined experimentally by the number of iterations that are necessary to achieve a given accuracy. The scheme has the following form:

$$\begin{aligned}
& \frac{1}{h_x^2} \varphi_{i+1,k}^{(m+1)} - 2 \left(\frac{1}{h_x^2} + \frac{1}{h_y^2} \right) \varphi_{i,k}^{(m+1)} + \frac{1}{h_x^2} \varphi_{i-1,k}^{(m+1)} = \\
& = \omega \rho_{i,k} - \frac{\omega}{h_y^2} \left(\varphi_{i,k-1}^{(m+1)} + \varphi_{i,k+1}^{(m)} \right) - \\
& - (\omega - 1) \left[\frac{1}{h_x^2} \varphi_{i+1,k}^{(m)} - 2 \left(\frac{1}{h_x^2} + \frac{1}{h_y^2} \right) \varphi_{i,k}^{(m)} + \frac{1}{h_x^2} \varphi_{i-1,k}^{(m)} \right],
\end{aligned}$$

where m is the iteration number.

The following variables are used in the subroutine:

- `im1` — number of grid nodes along the x -axis;
- `km1` — number of grid nodes along the y -axis;
- `hx` — grid step along the x -axis;
- `hy` — grid step along the y -axis;
- `rho(im1,km1)` — array of values of the grid right-hand side function of the Poisson equation;

phi(im1,km1) — array of values of the grid solution of the Poisson equation;

omega — iteration parameter;

eps — accuracy of the solution;

alf(im1),bet(im1) — operating arrays.

```

subroutine poisson2(im1,km1,hx,hy,rho,phi,
*                                omega,eps,alf,bet)
implicit none
integer im1,km1
real*8 phi(im1,km1),rho(im1,km1)
real*8 alf(im1),bet(im1)
integer i,k,im,km
real*8 hx,hy,omega,eps,c1,c2,a,b,d,f,s,s1,s2
im=im-1
km=km-1
c1=1./hx**2
c2=1./hy**2
a=c1
b=c1
d=2.*c2+a+b
1 s1=0.
do 4 k=2,km
alf(2)=0.
bet(2)=phi(1,k)
do 2 i=2,im
f=(omega-1.)*(c1*(phi(i-1,k)+phi(i+1,k))-
*2.*(c1+c2)*phi(i,k))+
*omega*(c2*(phi(i,k+1)+phi(i,k-1))-rho(i,k))
s=d-a*alf(i)
alf(i+1)=b/s
bet(i+1)=(a*bet(i)+f)/s
2 continue
do 3 i=im,2,-1
s=alf(i+1)*phi(i+1,k)+bet(i+1)
s2=dabs(s-phi(i,k))
if(s2.gt.s1) s1=s2
phi(i,k)=s
3 continue
4 continue
if(s1.gt.eps) goto 1
return
end

```

H. Subroutine of numerical integration of the full system of Maxwell equations

The subroutine realizes a numerical solution of Maxwell equations:

$$\text{rot} \mathbf{B} = \frac{4\pi}{c} \mathbf{j} + \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t},$$

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

$$\text{div} \mathbf{E} = 4\pi\rho,$$

$$\text{div} \mathbf{B} = 0$$

in the two-dimensional rectangular domain $0 < x < x_{max}$, $0 < y < y_{max}$ in one time step. The case when plasma is absent, i.e., $\rho = 0$ and $\mathbf{j} = 0$, is considered. Periodic boundary conditions are specified on the boundary $y = 0$ and $y = y_{max}$ for all functions: $f(x, 0, t) = f(x, y_{max}, t)$. On the boundary $x = 0$, the pulse of electric field intensity is given in the following form:

$$E_x(0, y, t) = 0,$$

$$E_y(0, y, t) = A \exp \left(-\frac{(y - y_0)^2}{2r_y^2} - \frac{(x_0 - t)^2}{2r_x^2} \right) \cos(\omega t),$$

$$E_z(0, y, t) = A \exp \left(-\frac{(y - y_0)^2}{2r_y^2} - \frac{(x_0 - t)^2}{2r_x^2} \right) \sin(\omega t).$$

Here A is the amplitude of the pulse, x_0 and y_0 denote the location of the pulse center at the initial time, r_x and r_y are root-mean-square dimensions of the pulse along the x - and y -axes, and ω is the frequency of electromagnetic vibrations. The following condition is set on the boundary $x = x_{max}$:

$$\mathbf{E}_{I,l} = \left(\mathbf{E}_{I,l} + \frac{\tau}{h_x} \mathbf{E}_{I-1,l} \right) / \left(1 + \frac{\tau}{h_x} \right).$$

It allows the pulse to leave the domain without reflection. A uniform rectangular grid with steps h_x and h_y in the directions of x - and y -axes, respectively, is used. The subroutine realizes the leapfrog scheme [38] with the time step τ :

$$\frac{H_{z,i-1/2,l}^{m+1/2} - H_{z,i-1/2,l}^{m-1/2}}{\tau} = -\frac{E_{z,i-1/2,l+1/2}^m - E_{z,i-1/2,l-1/2}^m}{h_y},$$

$$\frac{H_{y,i-1/2,l}^{m+1/2} - H_{y,i-1/2,l}^{m-1/2}}{\tau} = \frac{E_{z,i+1/2,l-1/2}^m - E_{z,i-1/2,l-1/2}^m}{h_x},$$

$$\frac{H_{z,i,l}^{m+1/2} - H_{z,i,l}^{m-1/2}}{\tau} = \frac{E_{x,i,l+1/2}^m - E_{x,i,l-1/2}^m}{h_y} - \frac{E_{y,i+1/2,l}^m - E_{y,i-1/2,l}^m}{h_x},$$

$$\begin{aligned} \frac{E_{x,i,l-1/2}^{m+1} - E_{x,i,l-1/2}^m}{\tau} &= \frac{H_{z,i,l}^m - H_{z,i,l-1}^m}{h_y}, \\ \frac{E_{y,i-1/2,l}^{m+1} - E_{y,i-1/2,l}^m}{\tau} &= -\frac{H_{z,i,l}^m - H_{z,i-1,l}^m}{h_x}, \\ \frac{E_{z,i-1/2,l-1/2}^{m+1} - E_{z,i-1/2,l-1/2}^m}{\tau} &= \frac{H_{y,i,l-1/2}^m - H_{y,i-1,l-1/2}^m}{h_x} - \\ &\quad - \frac{H_{x,i-1/2,l}^m - H_{x,i-1/2,l-1}^m}{h_y}. \end{aligned}$$

Only the first two equations of the Maxwell system are used in this scheme. In this case, the difference analog of the last equation of the system

$$\frac{H_{x,i+1/2,l}^{m-1/2} - H_{x,i-1/2,l}^{m-1/2}}{h_x} + \frac{H_{y,i,l+1/2}^{m-1/2} - H_{y,i,l-1/2}^{m-1/2}}{h_y} = 0$$

is satisfied identically, and the error in the difference equation for the next-to-last equation of the system

$$\frac{E_{x,i,l-1/2}^m - E_{x,i-1,l-1/2}^m}{h_x} + \frac{E_{y,i-1/2,l}^m - E_{y,i-1/2,l-1}^m}{h_y} = \rho_{i-1/2,l-1/2}^m$$

depends on the consistency between the calculation of the density of current j and the density of plasma ρ . In the absence of plasma, this equation is also satisfied identically.

The following variables are used in the subroutine:

$hx(\text{imp}, \text{lmp})$, $hy(\text{imp}, \text{lmp})$, $hz(\text{imp}, \text{lmp})$ — arrays of the x -, y -, and z -components of the magnetic field;

$ex(\text{imp}, \text{lmp})$, $ey(\text{imp}, \text{lmp})$, $ez(\text{imp}, \text{lmp})$ — arrays of the x -, y -, and z -components of the electric field;

$jx(\text{imp}, \text{lmp})$, $jy(\text{imp}, \text{lmp})$, $jz(\text{imp}, \text{lmp})$ — arrays of the x -, y -, and z -components of the current density;

tx — time;

$x0$, $y0$ — initial location of the center of the pulse;

$r0$ — root-mean-square half-width of the pulse along the y -axis;

rl — root-mean-square half-width of the pulse along the x -axis;

$b0$ — amplitude of the pulse;

wk — frequency of the pulse;

tau — time step;

$h1$ — grid step along the x -axis;

$h2$ — grid step along the y -axis.

```

subroutine emh(hx,hy,hz,ex,ey,ez,jx,jy,jz,
              tx,x0,y0,r0,rl,b0,wk,
              tau,h1,h2)
parameter(imp=102,lmp=52)
real*8 hx(imp,lmp),hy(imp,lmp),hz(imp,lmp),
=ex(imp,lmp),ey(imp,lmp),ez(imp,lmp),
=jx(imp,lmp),jy(imp,lmp),jz(imp,lmp)
im1=imp-1
lm1=lmp-1
c1=tau/h1
c2=tau/h2
c
c — Calculation of magnetic fields —
c
do 1 i=1,imp
do 1 l=1,lm1
hx(i,l)=hx(i,l)-c2*(ez(i,l+1)-ez(i,l))
1 continue
do 2 i=1,im1
do 2 l=1,lmp
hy(i,l)=hy(i,l)+c1*(ez(i+1,l)-ez(i,l))
2 continue
do 3 i=1,im1
do 3 l=1,lm1
hz(i,l)=hz(i,l)+c2*(ex(i,l+1)-ex(i,l))-
-c1*(ey(i+1,l)-ey(i,l))
3 continue
c
c — Calculation of electric fields —
c
do 4 i=1,im1
do 4 l=2,lm1
ex(i,l)=ex(i,l)+c2*(hz(i,l)-hz(i,l-1))-tau*jx(i,l)
4 continue
do 5 i=2,im1
do 5 l=1,lm1
ey(i,l)=ey(i,l)-c1*(hz(i,l)-hz(i-1,l))-tau*jy(i,l)
5 continue
do 6 i=2,im1
do 6 l=2,lm1
ez(i,l)=ez(i,l)+c1*(hy(i,l)-hy(i-1,l))-
*c2*(hx(i,l)-hx(i,l-1))-tau*jz(i,l)
6 continue
c
c — Periodic boundary conditions —
c

```

```

do 7 i=1,im1
ex(i,1)=ex(i,lm1)
ex(i,lm1)=ex(i,2)
7 continue
do 8 i=2,im1
ez(i,1)=ez(i,lm1)
ez(i,lm1)=ez(i,2)
8 continue
c
c - Boundary conditions simulating the pulse at x=0
c - and boundary conditions of free output at x=xm.
c
do 9 l=1,lm2
ex(1,l)=0.
ex(imp,l)=(ex(imp,l)+c1*ex(im1,l))/(1.+c1)
9 continue
do 10 l=1,lm1
x=-h1/2.
y=(l-1.)*h2-y0
r=(y/r0)**2+((x+x0-tx)/rl)**2
b=b0*exp(-r/2.)
s=wk*(tx-x)
ey(1,l)=b*cos(s)
ey(imp,l)=(ey(imp,l)+c1*ey(im1,l))/(1.+c1)
10 continue
do 11 l=1,lm1
x=-h1/2.
y=(l-1.5)*h2-y0
r=(y/r0)**2+((x+x0-tx)/rl)**2
b=b0*exp(-r/2.)
s=wk*(tx-x)
ez(1,l)=b*sin(s)
ez(imp,l)=(ez(imp,l)+c1*ez(im1,l))/(1.+c1)
11 continue
return
end

```

Bibliography

- [1] Kochin N.E., Kibel I.A., Roze N.V. Theoretical hydromechanics. P.I. M: FM, 1963, 583p (in Russian).
- [2] Leonard A. Vortex methods for flow simulation. J. Comp. Phys., 1980, V.37, p.289-335.
- [3] Vshivkov V.A., Kraeva M.A., Malyshkin V.E. Parallel Implementation of the Particle-in-Cell Method // Programming and Computer Software, 1997, V.23, N.2, p.87-97.
- [4] Kraeva M.A., Malyshkin V.E. Implementation of PIC Method on MIMD Multicomputers with Assembly Technology. In Proceed. of H-PCN Europe 1997, LNCS, V.1255, Springer Verlag, p. 541-549.
- [5] Kraeva M.A., Malyshkin V.E. A Dynamical balance of assembly in the realization PIC - method on MIMD multicomputers. Programmirovanie. 1999, N.1 (in Russian).
- [6] Kraeva M.A., Malyshkin V.E. Assembly Technology for Parallel Realization of Numerical Models on MIMD-Multicomputers. In the special issue of the International Journal on Future Generation Computer Systems (Elsevier), devoted to Parallel Computing Technologies. 2001, V.17, p.755-765.
- [7] Hockney R.W., Eastwood J.W. Computer simulation using particles. McGraw-Hill Inc. 1981, p.523.
- [8] Christiansen J.P. Vortex. Two-dimensional hydrodynamics simulation code. / Culham. Lab. Rep., CLM-106. HMSO. London: 1970.
- [9] Harlow F.H., Dickman D.O., Harris D.E., Martin R.E. Two dimensional hydrodynamic calculations. / Los Alamos Scientific Lab. Rep. NLA-2301, 1959.
- [10] Harlow F.H. The particle-in-cell computing method in fluid dynamics. Meth. Comput. Phys. 1964, V.3, p.319-343.
- [11] Theoretical foundation and construction of the numerical algorithms for the problems of mathematical physics. / Ed. by K.I. Babenko. Moscow: Nauka, 1979, p.295 (in Russian).
- [12] Yanenko N.N., Anuchina N.N., Petrenko V.E., Shokin Yu.I. On a calculation methods for gasdynamic problems with large deformations. Chislennye metody mehaniki sploshnoi sredy. 1970, V.1, p.40-62 (in Russian).
- [13] Belotserkovsky O.M., Davyudov Yu.M. A coarse particle method in gas dynamic. Moscow: Nauka, 1982, p.392 (in Russian).
- [14] Batchelor J. Introduction to the fluid dynamics. Moscow: Mir, 1973, 757p (in Russian).

- [15] Lamb H. Hydrodynamics. Dover publications Inc., New York, 1954.
- [16] Rosenhead L. The formation of vortices from a surface of discontinuity. Proc. Roy. Soc. London A, 1931, v.134, p. 170-192.
- [17] Chorin A.J. Numerical study of slightly viscous flow. J. Fluid. Mech., 1973, v.57, p.785-796.
- [18] Baker G.R. The "Cloud in Cell" Technique Applied to the Roll Up of Vortex Sheets. J. Comp. Phys., 1979, V.31, N.1, p.76-95.
- [19] Christiansen J.P. Numerical simulation of Hydrodynamics by the Method of Point Vortices. // J. Comp. Phys., 1973, V.13, N.3, p.363-379.
- [20] Methods in computational physics. Plasma physics / Ed. by Alder B., Fernbach S., Rotenberg M. New York, Academic Press, 1970, V.9.
- [21] Buneman O. Dissipation of currents in ionized media. Phys. Rev., 1959, V. 115, N.3, p. 503-519.
- [22] Berezin Yu.A., Vshivkov V.A. Particle-in-cell method in rarefied plasma dynamics. Novosibirsk: Nauka, 1980, 94p (In Russian).
- [23] Kertis D. Monte Carlo method for iteration of linear operators. Uspehi matematicheskix nauk. 1957, V.12, p.149-174 (in Russian).
- [24] Haviland J.K. The solution of two molecular flow problems by the Monte-Karlo method. Methods in Comput.Phys., Adv. in Research and Appl. in Hydrodinamics. N.Y.: 1965, V.4, p.109-209.
- [25] Grigoryev Yu.N., Ivanov M.S., Kharitonova N.I. On a question about solving of nonlinear kinetic equations of the rarefied gas dynamic by the Monte Carlo method. Chislennye metody mehaniki sploshnoi sredy. 1971, V.2. p.101-107 (in Russian).
- [26] Grigoryev Yu.N., Yanenko N.N., Ivanov M.S. Numerical simulation of the rarefied gas dynamics problems. Lect. Notes Phys.,141, Berlin-Heidelberg-N.-Y.-Tokyo, Spr.-Verlag, 1981, p.454-460.
- [27] Yanenko N.N., Grigoryev Yu.N., et al. Methods of statistical modeling and direct numerical integration of kinetic equations of gas theory. Development and application to the problems of rarefied gas dynamics. Proc. 13th Int. Symp. on RGD, V.1, N.-Y.-London: Plenum Press, 1985, p.371-382.
- [28] Belotserkovsky O.M., Yanitsky V.E. The statistical particle-in -cell method for solving rarefied gas dynamics problems. J. vychislitelnoi matematiki i matematicheskoi fiziki. 1975, V.15, N.5, p.1195-1208 (in Russian).
- [29] Belotserkovsky O.M. Numerical modeling in the mechanics of continuous media. Moscow: Nauka, 1984, 514p (in Russian).

- [30] Bird G.A. Molecular Gas Dynamics. Oxford: Clarendon Press, 1976.
- [31] Pullin D.I. Direct simulation Methods for Compressible Inviscid Ideal-Gas Flow. J. Comp. Phys., 1980, V.34, N 2. p.231-244.
- [32] Yanitsky V.E. Statistical particle method for solving of some problems of kinetic gas theory and turbulence. Dis. doct. fiz.-mat. nauk. Moscow: 1984, 271p. (in Russian).
- [33] Zenkevich O., Morgan K. Finite elements and approximations. Moscow: Mir, 1986 (in Russian).
- [34] G.A.Korn, T.M. Korn. Mathematical Handbook for scientists and engineers. N.-Y.: McGraw-Hill Book Co., 1968.
- [35] Hockney R.W. A computer experiment of anomalous diffusion. Phys. Fluids. 1966, V.9, N.9, p.1826-1835.
- [36] Burger P., Dunn D.A., Halstead A.S. Computer experiments on the randomization of electrons in a collisionless plasma. Phys. Fluids. 1965, V.8, N.12, p.2263-2272.
- [37] Birdsall C.K., Fuss D. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. J. Comp. Phys. 1969, V.3, N.4, p. 494-511.
- [38] Birdsall Ch.K., Langdon A.B. Plasma physics via computer simulation. McGraw-Hill, 1985.
- [39] Vshivkov V.A., Snytnikov V.N. On the particle method for solving of the Vlasov kinetic equations. J. vychislitelnoi matematiki i matematicheskoi fiziki. 1998, V.38, N.11, p.1877-1883 (in Russian).
- [40] Vshivkov V.A., Romanov D.V., Snytnikov V.N. A problem of self-heating of model plasma in the particle method. Vychislitelnye technologii. 1999, V.4, N.3, p.62-72 (in Russian).
- [41] Vshivkov V.A. The approximation properties of the particle-in-cell method. Comp. Maths. Math. Phys. 1996, V.36, N.4, p.509-515.
- [42] Potter D. An introduction to computational physics. John Wiley and Sons, Ltd., London, 1973.
- [43] Bluhm H., Hoppe P., Bachmann H. et al. Progress in the development of a high power focusing B-applied extractortype ion diode for 1.5 TW pulse generator KALIF. Proc. 8th Internat. Conf. High-Power Particle Beams, Novosibirsk, 2-5 July 1990. World Scient. Singapore, p.451-456, 1990.
- [44] Matsumoto M., Kawata S. TRIPIC: triangular-mesh PIC code. J. Comput. Phys. 1990, V.87, p.488-493.
- [45] Thompson J.F., Warsi Z.U.A., Mastin C.W. Boundary-fitted coordinate systems for numerical solution of partial differential equations – a review. J. Comput. Phys. 1982, V.47, p.1-108.

- [46] Jones M.E. Electromagnetic PIC codes with body-fitted coordinates. Proc. 12th Internat. Conf. Numerical Simulation of Plasmas, San Francisco, 20–24 September 1984, IM3, 1987.
- [47] Westermann T. Numerical modelling of the stationary Maxwell – Lorentz system in technical devices. Intern. J. Num. Mod.: Electronic Network, Devices and Fields. 1994, V.7, p.43-67.
- [48] Halter E., Krauss M., C.-D. Munz et al. A concept for numerical solution of the Maxwell–Vlasov system. Forshungszentrum: Karlsruhe – Umwelt und Technik, FZKA 5564, 1995.
- [49] Assous F., Degond P., Segre J. A particle-tracking method for 3D electromagnetic PIC codes on unstructured meshes. Comput. Phys. Commun. 1992, V.72, p.105-114.
- [50] Sonnedrucker E., Ambrosiano J.J., Brandon S.T. A finite element formulation of the Darwin PIC model on unstructured grids. J. Comput. Phys., 1995, V.121, p.281-297.
- [51] Kruglyakova L.V., Neledov A.V., Tishkin V.F., Filatov A.Yu. Unstructured adaptive grids for problems of mathematical physics (review). Matematicheskoe modelirovanie. 1998, V.10, p.93-116 (in Russian).
- [52] Boldyrev A.S., Gasilov V.A., Olxovskaya V.G. On solving hyperbolic equations on unstructured grids. Matematicheskoe modelirovanie. 1996, V.8, p.51-78 (in Russian).
- [53] Lohner R., Ambrosiano J. A vectorized particle tracer for unstructured grids. J. Comput. Phys. 1990, V.91, p.22-31.
- [54] Westermann T. Localization shemes in 2D boundary-fitted grids. J. Comput. Phys. 1992, V.101, p.307-313.
- [55] Lohner R. Robust, vectorized search algorithms for interpolation on unstructured grids. J. Comput. Phys. 1995, V.118, p.380-387.
- [56] Fedoruk M., C.-D. Munz, Omnes P., Schneider R. A Maxwell–Lorentz solver for self-consistent particle-field simulations on unstructured grids. Karlsruhe, FZKA 6115, p.1-81, 1998.
- [57] Dudnikova G.I., Romanov D.V., Fedoruk M.P. On the particle model on the unstructured grids. Vychislitelnye technologii. 1998, V.3, p.30-46 (in Russian).
- [58] Dudnikova G.I., Romanov D.V., Fedoruk M.P. On algorithms implementing the particle method on unstructured grids. Computational mathematics and mathematical physics. 2000, V.40, p.147-158.
- [59] Landau L.D., Lifshits E.M. Mechanics. Moscow: Nauka, 1988 (in Russian).
- [60] Assous F., Degond P., Heintze E. et al. On a finite element method for solving the three-dimensional Maxwell equations. J. Comput. Phys. 1993, V.109, p.222-237.

- [61] Ambrosiano J.J., Brandon S.T., Lohner R., DeVore C.R. Electromagnetics via the Taylor-Galerkin finite element method on unstructured grids. *J. Comput. Phys.* 1994, V.110, p.310-319.
- [62] Hermeline F. Two Coupled Particle-Finite Volume Methods Using Delaunay-Voronoi Meshes for the Approximation of Vlasov-Poisson and Vlasov-Maxwell Equations. *J. Comput. Phys.* 1993, V. 106, p.1-18.
- [63] J.-P. Cioni, L. Fezoui, D. Issautier. High order upwind schemes for solving time-domain Maxwell equations. *La Recherche Aérospatiale.* 1994, N.5, p.319-328.
- [64] Rich M.A. A Method for Eulerian Fluid Dynamics. *Los Alamos Sci. Lab.*, 1963, Rep. No LAMS-2826.
- [65] Gentry R.A., Martin R.E. and Daly B.J. An Eulerian Differencing Method for Unsteady Compressible Flow Problems. *J. Comp.Phys.* 1966, V.1, p.87-118.
- [66] Belotserkovsky O.M., Davydov Yu.M. A non-stationary "coarse-particle" method for gasdynamical computations. *J. vychislitelnoi matematiki i matematicheskoi fiziki.* 1971, V.11, N.1, p.182-207 (in Russian).
- [67] Sapojnikov G.A. A combine method of fluid fluxes and particle-in-cell for calculations of the gasdynamical flows / Sb. *Voprosy razrabotki i ekspluatacji paketov prikladnyx programm.* Novosibirsk: ITPM SO AN SSSR, 1981. p.89-97 (in Russian).
- [68] Godunov S.K., Ryabenkii V.S. Finite-difference schemes (Introduction to the theory). Moscow: Nauka, 1973, p.400 (in Russian).
- [69] Anuchina N.N. On the calculation methods for flows of compressible fluid with large deformations. *Chislennye metody mehaniki sploshnoi sredy.* 1970, V. 1, N. 4, p.3-84 (in Russian).
- [70] Petrils T. An approach to the inviscid rotational compressible flows. / Prepr. B. Rol. Univ. Fac. Math. Res. Seminar. 1982, N.1. p.44-63.
- [71] Chorin A.J. Vortex sheet approximation of boundary layers. *J. Comp. Phys.* 1978, V.27, N.3 p.428-445.
- [72] Chorin A.J. Lectures on Turbulence Theory. Boston.: Publish/Perish, 1976. 159p.
- [73] Roshal A.S. The fast Fourier transform in computational physics. *Izv. Vuzov. Radiofizika,* 1976, V.XIX, N.10, p.1425-1454 (in Russian).
- [74] Yanenko N.N., Veretentsev A.N., Grigoryev Yu.N. A Hamiltonian formalism for a spatial system of small vortices in ideal fluid. *Chislennye metody mehaniki sploshnoi sredyu.* 1979, V.10, N.5, p.144-149 (in Russian).

- [75] Grigoryev Yu.N., Levinsky V.B., Yanenko N.N. Modeling of turbulence by ensembles of vortices with inviscid interaction. Arch. Mech. 1984, V.36, N.2, p.279-292.
- [76] Williamson J.H. Initial particle distributions for simulated plasma. J. Comp. Phys. 1971, V.8, p.258-267.
- [77] Byers J.A., Grewal M. Perpendicularly propagating plasma cyclotron instabilities simulated with one-dimensional computer model. Phys. Fluids, 1970, V.13, p.1819-1830.
- [78] Matsuda Y., Crawford F.W. Computational study of nonlinear plasma waves. Phys. Fluids, 1975, V.18, p.1336-1353.
- [79] Boris J.P. Relativistic plasma simulation – optimization of a hybrid code coordinates. Proceedings 4th International Conference on the Numerical Simulation of Plasmas, Washington, 20–24 September 1970, P.3-67.
- [80] Langdon A.B. Effects of the spatial grid in simulation plasma. J. Comp. Phys. 1970, V.6, p.247-267.
- [81] Langdon A.B. "Energy-conserving" plasma simulation algorithms. J. Comp. Phys. 1973, V.12, p.247-268.
- [82] White R.B., Fried B.D., Coroniti F.V. Structure of ion acoustic solitons and shock waves in a two-component plasma. Phys. Fluids. 1972, V.15, p.1484-1490.
- [83] Sagdeev R.Z. Cooperative processes and shock waves in a rarefied plasmas / Voprosy teorii plasmy. Moscow: Atomizdat, 1964, Vyp.4, p.20-80 (in Russian).
- [84] Bernhardt P.A., Roussel-Dupre R.A., Haerendel G. Observations and theory of the AMPTE magnetotail barium released. J.Geophys.Res. 1987, V.92, p.5777-5794.
- [85] Berezin Yu.A., Vshivkov V.A. Shock waves of arbitrary amplitude in rarefied plasma with magnetic field. Fizika plasmy. 1977, V.3, p.365-370 (in Russian).
- [86] Vshivkov V.A., Fedoruk M.P. On algorithm for computation of shock waves in rarefied plasma propagating at arbitrary angle with respect to a magnetic field. Chislennye metody mehaniki sploshnoi sredyu. 1985, V.16, N.5, p.33-50 (in Russian).
- [87] Operation "Argus". Moscow: Atomizdat, 1960 (in Russian).
- [88] Operation "Star fish". Moscow: Atomizdat, 1964 (in Russian).
- [89] Dudnikova G.I., Orishich A.M., Ponomarenko A.G., Zakharov Yu.P., Vshivkov V.A. Laboratory and computer simulation of generation magnetosonic disturbances in magneto-spheric plasma In "Proc. XX ICPIG", Piza, 1991.

- [90] Borovsky J.E., Pongratz M.B., Roussel-Dupre R.A., Tan T. The laboratory simulation of unmagnetized supernova remnants: absence of a blast wave. *Astrophys. J.* 1984, V.280, p.802-808.
- [91] Golubev A.I., Solovyev A.A., Terekhin V.A. On a collisionless expansion of an ionized cloud in a homogeneous magnetized plasma. *J. Prikl. mexan. and tex. fiz.* 1978, N.5, p.33-42 (in Russian).
- [92] Bashurin V.P., Golubev A.I., Terekhin V.A. On a collisionless deceleration of an ionized cloud expanding into a homogeneous magnetized plasma. 1983. *J. Prikl. mex. and tex. fiz.* 1983, N.5, p.10-17 (in Russian).
- [93] Berezin Yu.A., Vshivkov V.A., Dudnikova G.I., Fedoruk M.P. Collisionless slowing down of a plasma cloud in nonuniform magnetized background. *Sov. J. Plasma Phys.* 1992, V.18, p.812-815.
- [94] Berezin Yu.A., Dudnikova G.I., Fedoruk M.P., Vshivkov V.A. Explosion phenomena in collisionless plasmas at super-alfvenic speed. *Intern. J. Comput. Fluids. Dyn.* 1998, V.10, p.117-126.
- [95] Bulanov S.V., Vshivkov V.A., Dudnikova G.I., Naumova N.M., Pgoraro F., Pogorelsky I.V. Laser acceleration of charged particles in nonuniform plasmas. *Plasma Physics Reports.* 1997, N.23, N.4, p. 259-269.
- [96] Askar'yan G.A., Bulanov S.V., Dudnikova G.I., Esirkepov T.Zh., Lontano M., Meyer-ter-Vehn J., Pegoraro F., Pukhov A.M., Vshivkov V.A. Magnetic interaction of ultrashort high-intensity laser pulses in plasmas. *Plasma Physics and Controlled Fusion*, 1997, V.39, N. 5A, p. 137-144.
- [97] Pegoraro F., Bulanov S.V., Califano F., Esirkepov T.Zh., Lontano M., Meyer-ter-Vehn J., Naumova N.M., Pukhov A.M., Vshivkov V.A. Magnetic fields from high-intensity laser pulses in plasmas. *Plasma Phys. Control. Fusion.* 1997, V.39, p.B261-B272.
- [98] Vshivkov V.A., Bulanov S.V., Naumova N.M., Pegoraro F. Nonlinear electrodynamics of the interaction of ultra-intense laser pulses with a thin foil. *Physics of Plasmas*, 1998, V.5, N.7.
- [99] Bulanov S.V., Esirkepov T.Zh., Naumova N.M., Pegoraro F., Vshivkov V.A. Solitonlike electromagnetic waves behind a superintense laser pulse in a plasma. *Physical Review Letters*, 1999, V. 82, N.17, p.3440-3443.
- [100] Bulanov S.V., Califano F., Dudnikova G.I., Liseikina T.V., Naumova N.M., Pegoraro F., Vshivkov V.A., Sakai J.-I., Sakharov A.S. Laser acceleration of charged particles in inhomogeneous plasmas. II: Particle injection into acceleration phase due to nonlinear wake wave-breaking. *Plasma Physics Reports*, 1999, V.25, N.6, p.468-481.
- [101] Bulanov S.V., Vshivkov V.A., Dudnikova G.I., Esirkepov T.Zh., Califano F., Kamenets F.F., Liseikina T.V., Naumova N.M., Pegoraro F.

- Interaction of petawatt laser pulses with underdense plasmas. *Plasma Physics Reports*, 1999, V.25, N.9, p.701-714.
- [102] Esirkepov T.Zh., Sentoku Y., Mima K., Nishihara K., Califano F., Pegoraro F., Naumova N.M., Bulanov S.V., Ueshima Y., Liseikina T.V., Vshivkov V.A., Kato Y. Ion acceleration by superintense laser pulses in plasmas. *JETP Letters*, 1999, V.70, N.2, p.82-89.
 - [103] Liseikina T.V., Califano F., Vshivkov V.A., Pegoraro F., Bulanov S.V. Small-scale electron density and magnetic-field structures in the wake of an ultraintense laser pulse. *Physical Review E*, 1999, V.60, N.5, p.5991-5997.
 - [104] Pegoraro F., Bulanov S.V., Califano F., Esirkepov T.Zh., Liseikina T.V., Lontano M., Naumova N.M., Ruhl H., Sakharov A.S., Vshivkov V.A. Coherent electromagnetic structures in relativistic plasmas. *Physica Scripta*, 2000, V.T84, p.89-93.
 - [105] Pegoraro F., Bulanov S.V., Califano F., Esirkepov T.Zh., Liseikina T.V., Naumova N.M., Ruhl H., Vshivkov V.A. Ion acceleration regimes in underdense plasmas. *IEEE Transaction on Plasma Science*, 2000, V.28, N.4, p.1177-1183.
 - [106] Sentoku Y., Liseikina T.V., Esirkepov T.Zh., Califano F., Naumova N.M., Ueshima Y., Vshivkov V.A., Kato Y., Mima K., Nishihara K., Pegoraro F., Bulanov S.V. High Density Collimated Beams of Relativistic Ions Produced by Petawatt Laser Pulses in Plasmas. *Phys.Rev.E*, 2000, V.62, N.5, p.7271-7281.
 - [107] Vshivkov V.A., Dudnikova G.I. Algorithms for solution of the problem of interaction laser pulse with plasma. *Vychislitelnye tekhnologii*, 2001, V.6, N.2, p.47-63 (in Russian).
 - [108] Bulanov S.V., Califano F., Dudnikova G.I., Esirkepov T.Zh., Inovenkov I.N., Kamenets F.F., Liseikina T.V., Lontano M., Mima K., Naumova N.M., Nishihara K., Pegoraro F., Ruhl H., Sakharov A.S., Sentoku Y., Vshivkov V.A., Zhakhovskii V.V. Relativistic Interaction of Laser Pulses with Plasmas / In: *Reviews of Plasma Physics*. New York: Kluwer Academic, 2001, V.21, p. 227-335.
 - [109] Macchi A., Cornolti F., Pegoraro F., Liseikina T., Ruhl H., Vshivkov V. Surface oscillations in overdense plasmas irradiated by ultrashort laser pulses. *Phys. Rev. Lett.*, 2001, V.87, N.20.
 - [110] Langdon A.B., Lasinski B.F. Electromagnetic and relativistic plasma simulation models. *Meth. Comput. Phys.*, 1976, V.16, p.327-366.
 - [111] Marder B. A method for incorporating Gauss' law into electromagnetic PIC codes. *J. Comput. Phys.*, 1987, V.68, p.48-55.
 - [112] Villasenor J., Buneman O. Rigorous charge conservation for local electromagnetic field solver. *Computer Phys. Comm.*, 1992, V.69, p. 306-316.

- [113] Cogan M.N. Introduction in the rarefied gas dynamics. Moscow: Nauka, 1967. p.440 (in Russian)
- [114] Mikhailov G.A. Some problems of Monte Carlo method theory. Novosibirsk: Nauka, 1974, p.144 (in Russian).
- [115] Kach M. Several probability problems of physics and mathematics. Moscow: Nauka, 1967. p.176 (in Russian).
- [116] Shakhov E.M. Kinetic model equations and numerical results / Rar. Gas Dyn.: Proc. 14th Symp. Rar. Gas Dyn. v.1. Tokyo, 1984, p.137-148.
- [117] Marchuk G.I., Yanenko N.N. A solution of multi-dimensional kinetic equations by the splitting method. Doklady AN SSSR. 1964, V.157, N.6 p.1241-1242 (in Russian).
- [118] McKean H.P. Speed of approach to equilibrium for Kac's caricature of a Maxwellian gas. Arc Rat. Mech and Anal. 1966, V.21, N.5, p.343-367.
- [119] Nanbu K. Direct simulation scheme derived from the Boltzmann equation I. Monocomponent gases. J. Phys. Soc. Jap. 1980, V.49, N.5. p.2042-2049.
- [120] Nanbu K. On the simulation method for BGK-equation. J. Phys. Soc. Jap. 1981, V.50, N.9. p.3154-3188.
- [121] Vladimirov V.G. Generalized functions in mathematical physics. Moscow: Nauka, 1979, 320p (in Russian).
- [122] Boyd I.D., Stark G.P.W. A comparison of the implementation and performance of the Nanbu and Bird direct simulation Monte-Carlo Methods. J. Phys. Fl. 1987, V.30, N.12. p.3661-3668.
- [123] Bird G.A. Simulation of Multi-dimensional and chemically Reacting Flows. Rar. Gas Dyn. V.1., Paris: Comm. L'energie Atom, 1979, p.365-388.
- [124] Voloschuk V.M., Sedunov Yu.S. Coagulation processes in the dispersion systems. Leningrad: Gidrometeoizdat, 1975, 320p (in Russian).
- [125] Rogazinskii S.V. Monte Carlo method for solving of non-linear coagulation equation / Teoriya i prilozheniya statisticheskogo modelirovaniya, Novosibirsk: VC AN SSSR, 1985, p.137-147 (in Russian).
- [126] Lundgren T.S. Model equation for nonhomogeneous turbulence. J. Phys. Fl. 1979, V.12, N.3, p.485-497.

