tive apparatus will consist of two beams of straights with intrinsic centers S_1 and S_2. Each point of an original will lie on one forming line of a conic surface, points ∞ and –∞ will lie on the same forming line of a conic surface.

b) An original of a projective straight can lie on a cylindrical surface with a directing projective straight and projection apparatus that consists of two beams of straights with centers S_1^∞ and S_2^∞ in nonintrinsic points.

References

1. Kantor G. A study of multiplicities. – 1914. – 189 p.

2. Klein F. Elementary mathematics from the position of the higher mathematics. – T.I. – M., Science, 1987. – 431 p. – V.II – M., Science, 1987. – 416 p.

3. Klein F. Higher geometry. – M., 1939. – 399 p.

4. Vertinskaya N.D. Mathematical modeling according to constructive descriptive geometry. – Irkutsk, 1992. – 130 p.

5. Vertinskaya N.D. Mathematic modeling of multi-factor and multi-parameter processes in multi-component systems according to constructive geometry V.I of lecture course. – Irkutsk, 2009. – 229 p.

6. Vertinskaya N.D. Mathematic modeling of multi-factor and multi-parameter processes in multi-component systems Irkutsk. – Irkutsk state technical university, 2009. – 229 p.

7. Klein F. Non-euclidean geometry. – Moscow-Leningrad, USSR, 1936. – 355 p.

8. Andreyev A. Analytic geometry. – M., 1905. – 610 p.

9. Aleksandrov P.S. Lectures on analytic geometry. – M., 1968. – 911 p.

The work was submitted to the International Scientific Conference «Production technologies», Italy (Rome – Florence), September, 7–14, 2013, came to the editorial office on 14.08.2013.

THE DISTRIBUTED CALCULATORS MODEL FOR MOLECULAR-DYNAMIC SIMULATION OF STRONG INTERACTION SYSTEMS

[1,2]Voronova L.I., [1]Trunov A.S., [1,2]Voronov V.I.

[1]Russian State University for the Humanities, Moscow;
[2]National Research University Higher School of Economics, Moscow, e-mail: voronova.lilia@ya.ru

The model of distributed calculators makes it possible a parallel calculation of the correlated N-particle system with a complex multi-particle interaction (long-range ionic and short-range repulsive, two- and three-particle covalent interactions) with MPI and CUDA technologies. The computational model is based on the mathematical model of heterogeneous descriptors developed by the authors, that allows shift the focus from the describing the physical interactions in the system to the description of data flow between the descriptors. The results of computer experiments, which compare the time of the simulation on the cluster of 16 calculators and GPU NVIDIA are given. The model of distributed calculators was being tested with the software package of RIS «MD-SLAG-MELT»[1].

Nowadays, computer modeling (CM) is widely used in various fields of modern science. In particular, in physical chemistry we can study the properties and structure of materials and their relationships.

Molecular-dynamic (MD), Monte-Carlo and quantum-chemical methods are applied there and allow to define different sets of properties. The molecular-dynamics method allows to define the whole complex of properties (structural, thermodynamic, transport) and to investigate the interrelations of nanostructure and physical-chemical properties [1–5].

The size of the simulated system for MD modeling is extremely important. A significant increase in the size of the system provides the practical relevance of the results. The calculation of the systems with 10^5–10^7 particles requires a large amount of time and computer resources and it makes impossible carrying out CM without high-performance computing [6, 7]. To solve this problem, the authors have developed a model of distributed calculators based on distributed computing methods for correlated N-particle system [8–10].

Physical phenomena that are adequately described by the classical and quasiclassical theory can be simulated (using *models of particles)* by molecular dynamics method. The term «*model of particles*» is the general one for a class of computing models in which the discrete description of the physical phenomena includes cooperating particles. Each modeling particle has a set of constant and variable attributes.

In this case molecular-dynamic simulation represents the numerical solution of the Cauchy’s boundary task, which means that the initial system state in a bounded region of space (calculation area) is specified at the time $t = 0$ and the boundary conditions are reserved on it. Modeling is tracking the time evolution of this configuration. The main part of calculation is the cycle on a time step in which the state of physical system changes on time for a small step Δt.

The current condition of the physical system is defined by the attributes of the final ensemble of particles, and the evolution of the system is defined by the interaction laws of these particles. The most of the molecular-dynamics systems relates to the class of long-range potentials, or considering only short-range covalent interactions.

The subject of this work is an investigation of the polymerizing systems with multi-particle interactions which means uniting some types of interactions – two-particle contribution (long-range ionic and short-range repulsive) and multi-particle ones (two and three-particle covalent interactions). The description of this class of models is given in Table 1 [10].

The ionic model is a part of ionic-covalent model though for modeling ionic connections it can be used only independently. In the ionic model (IM) potential functions are built for the ion system. The

IM includes models of short-range repulsive and long-range Coulomb interaction. The last one is calculated by the Ewald method or the fast multipole method.

The ionic-covalent model is more realistic to describe the interactions in the polymerizing systems. The model is based on the polymer theory of the slags, which suggests the presence of small non-destructive structural groups (elementary structure groups – ESG), which form the structural grid in the polymerizing melts.

Table 1

The interparticle interaction models are used in the information research system «MD-Slag-Melt» [10]

The Ionic model (IM)	
The Short-range model	The Long-range model
The Ionic-covalent model (ICM)	
– ICM of ESG (elementary structure group)	
– ICM of BI (bridge bonds)	
– ICME (environmental effect)	
The techniques covered net of covalent bonds	

The ICM unites the ionic model, the models of two- and three-particle covalent interaction in the elementary structure groups, the three-particle covalent interaction for bridge bonds, the environmental effects for the network forming atoms-oxygen bounds inside the ESG from the nearest neighbors outside ESG.

The features of the above mentioned physicochemical models require the development of a mathematical model that allows realizing high-performance computing for correlated systems with complex interactions containing 10^5–10^7 particles.

For solution the authors have developed a model of heterogeneous descriptors for distributed MD simulation of the correlated system of N-particles.

The basic elements of the model are the object and the descriptor, which provide the possibility of calculations distribution without detail specification of all interactions between the particles

We define *an object* as a set of particles descriptions and relations between them for the system in the initial state, which is built by specific rules and provides the possibility of the system decomposition for distributing and parallel calculations.

The objects are initialized by means of heterogeneous descriptors which contain polytypic elements of the description of the allocated object which is necessary for the distribution of calculations.

Thus, the system is a set of the objects characterized by heterogeneous descriptors the calculation of which can be distributed on separate calculators, combining results on a certain scheme.

On the basis of physicochemical models and the analysis of the program code of a local MD application, the authors have constructed a set of descriptors allowing to share the application code into independent blocks between calculators. The descriptors can be divided into two classes: one-particle descriptors and aggregators (two – and three-particle descriptors).

Both classes of descriptors include the possibility of parallel computing on different calculators. However, if the one-particle descriptors can be distributed on the calculators independently, then the aggregators (containing the elements describing the cross relations of different orders between one-particle descriptors and/or aggregators) are «dependant» on the one-particle descriptor. The aggregators are computed on the same computer with a «parental» one-particle descriptor.

The notation of local molecular-dynamic model in model of descriptors for distribution MD simulating is given in Table 2.

Thus, the set of objects and their relationship between them are devided into different types of subset according to heterogeneous descriptor.

More detailed characteristic of descriptors elements are given below.

{D1(i)} – the class of one-particle descriptors assuming distribution. It is expedient to divide this descriptor into two subset.

{D1s(i)} – a subset of the descriptors, containing constant scalar attributes of particles. Here D1s(i) = <I, σ q, m, tip,>, where i – particle number, tip – particle type, m – weight, q – charge, σ – rigid sphere radius.

{D1v(i)} – a subset of the descriptors containing variable vector attributes of particles. Here

$$D1v(i)=\langle i, \bar{r}, \bar{v}, \bar{a}, \bar{F}\rangle,$$

where $\bar{r}$ – radius vector; $\bar{v}$ – speed; $\bar{a}$ – acceleration; $\bar{F}$ – full force effecting on an i-particle at a given timestep.

{D∑(i)} – the class of heterogeneous descriptors providing creation of different orders relations between descriptors of different types. Some elements of the descriptors in the class are the storage, so these descriptors are called aggregators.

Three types of subset aggregators are defined by the authors.

1. {D∑2(i)} – a subset of the two-particle aggregators containing scalar and vector elements. They provide the calculation of cross relations, on the basis of one-particle descriptors, and also contain

scalar and vector stores. The aggregators depend on one-particle descriptors (called «brunches»). Data for calculation are received from «parental» descriptor. The aggregator realizes consequent calculation of all pair relations between fixed i- object and others to form storages.

Table 2

Distributed descriptor model of molecular-dynamic modeling N-particle correlation system

Classes of descriptors and different types of descriptors in the class	Type of descriptors	Type of elements	distribution
$D1(i)$	one-particle :	attributes of particles :	independent
$D1s(i) = <i, tip, m, q, \sigma>$	scalar	constant	
$D1v(i)=\langle i, \bar{r}, \bar{v}, \bar{a}, \bar{F}\rangle,$	vector	variable	
$D\sum(i)$	aggregators :	cross relationship between descriptors	depends on «parental» descriptor
$D\Sigma 2(i)=\langle i, j, \overline{f2}, \Sigma\overline{f2}, \varphi 2, \Sigma\varphi 2...\rangle$	two-particle	{D1(i)}∩{D1(j)}	
$D\Sigma 3(i)=\langle i, j, k, \overline{f3}, \Sigma\overline{f3}, \varphi 3, \Sigma\varphi 3\rangle$	three-particle	{D1(i)}∩{D1(j)} {D1(k)}	
$D(Property) = <T, P, E, C, \gamma_V, \alpha_P \beta_T, \chi_a, msd_a, \eta, \xi, l>$	multi-particle (macroscopic properties)	{D1(i)}∩ D∑2(i)∩ D∑3(i)	not distributed

Here

$$D\Sigma 2(i)=\left\langle i, j, \overline{f2}, \Sigma\overline{f2}, \varphi 2, \Sigma\varphi 2...\right\rangle,$$

where i, j – particles numbers; $\overline{f2}$ – force acting between i and j particles; φ2 – two-particle potential; $\Sigma\overline{f2}$ – the vector storage including different types of contributions to two-particle force effecting on an i-particle; Σφ2 – the scalar storage containing values of different power contributions to total two-particle energy.

2. {D∑3(i)} – a subset of three-particle aggregators which build third order cross relations on the basis of one-particle descriptors. Each aggregator contains the variable vector and scalar elements which are calculated on special algorithms, providing formation of limited selections of particles (elementary groups) from all range of numbers. The aggregators calculate the relations between triplets of particles within these selections. Like two-particle aggregators it contains scalar and vector stores being brunches in relation to a pair of one-partial descriptors.

Here

$$D\Sigma 3(i)=\left\langle i, j, k, \overline{f3}, \Sigma\overline{f3}, \varphi 3, \Sigma\varphi 3\right\rangle,$$

where i, j, k – triplets of particles from elementary groups; $\overline{f3}$ – three-particle interaction; φ3 – three-particle potential; $\Sigma\overline{f3}$ – the vector storage including different types of contributions to the three- particle force acting on an i-particle; Σφ3 – the scalar storage containing values of different power contributions to total three- particle energy.

3. {D(Property)} – aggregators subset containing elements, describing macroscopic properties of the system, based on two- particle and three- particle aggregators. Here

$$\mathrm{D\,(Property)} = <T, P, E, C, \gamma_V, \alpha_P, \beta_T, \chi_a, msd_a, \eta, \xi, l>,$$

where α – type of particles; T – temperature; P – pressure; the C – heat capacity; E – energy; γ_V – coefficient of thermal pressure; α_P – coefficient of thermal expansion, β_T – isothermal compressibility; χ_a – velocity autocorrelation function; msd_a – mean square displacement; D_a – diffusion coefficient; η, ξ, l – viscosity factor, coefficient of conductivity and thermal conductivity. These aggregators cannot be distributed.

The method of heterogeneous descriptors make it possible to transfer the accent from physical interactional in the system to information description of data flows redistribution between descriptors and use this approach to form computing model of redistributed calculation on the bases of heterogeneous descriptors.

The model of distributed calculators is based on a master-slave methodology. The master controls the slaves, separates a set of particles into subsets which are transferred to the slaves for calculation. The master receives data from the slaves and processes the results. The slaves calculate the N-particle system on the CPU or the GPU, using MPI and CUDA technologies.

The model of distributed calculators is given in Fig. 1.

The model assumes the implementation of distributed computing a heterogeneous computing environment. The calculators have different performance and type, such as the GPU and CPU [11].

On the first step of modeling the all set of particles is transmitted to every calculator. The master generates a subset of particles that will be transferred to the calculators for processing. At run time of work, each slave receives data from the master about the calculated subset of the particle. The slave provides interactions calculation describing descriptors *D*∑2(i) и D∑3(i), associated with the *i* particle, according to a given scheme. When each slave completes the calculation, it passes the results to master which gathers and processes data.

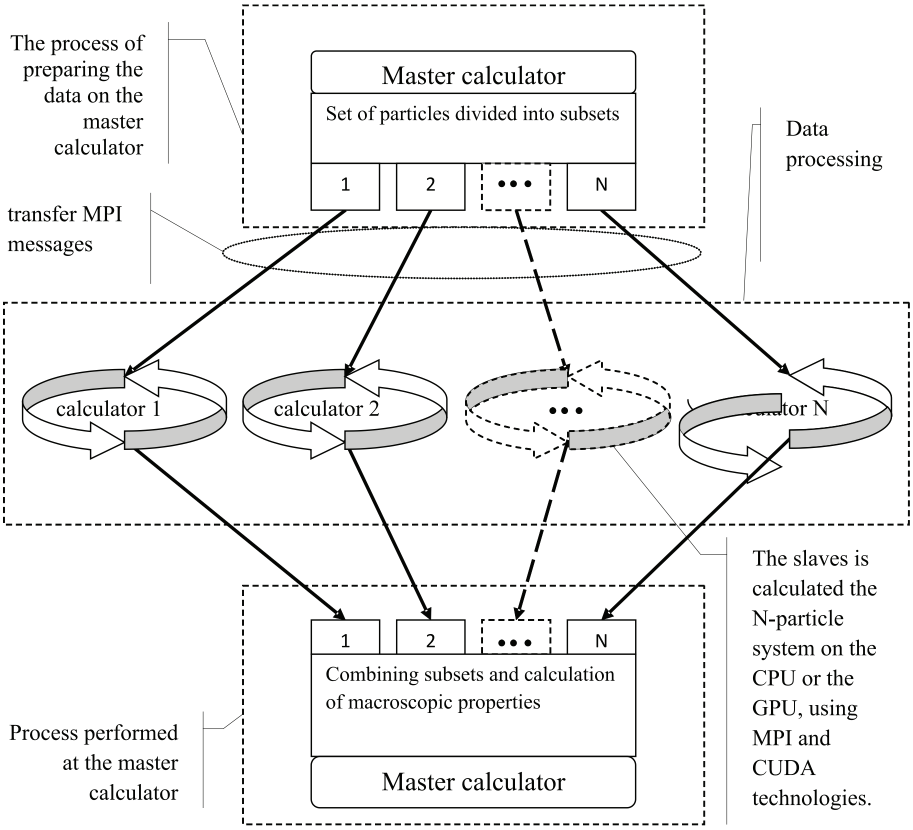
The process of preparing the data on the master calculator
Master calculator
Set of particles divided into subsets
1
2
•••
N
Data processing
transfer MPI messages
calculator 1
calculator 2
•••
tor N
The slaves is calculated the N-particle system on the CPU or the GPU, using MPI and CUDA technologies.
1
2
•••
N
Combining subsets and calculation of macroscopic properties
Process performed at the master calculator
Master calculator

Fig. 1. The model of distributed calculators using the CUDA and MPI technologies

The most pressing is to ensure parallel calculation of objects characterized by the descriptors D∑2(i), since the implementation of this part simulation has the largest computating time (a quadratic dependence on the number of particles N^2). Flowchart of the calculators work based on the model of heterogeneous descriptors is given in Fig. 2.

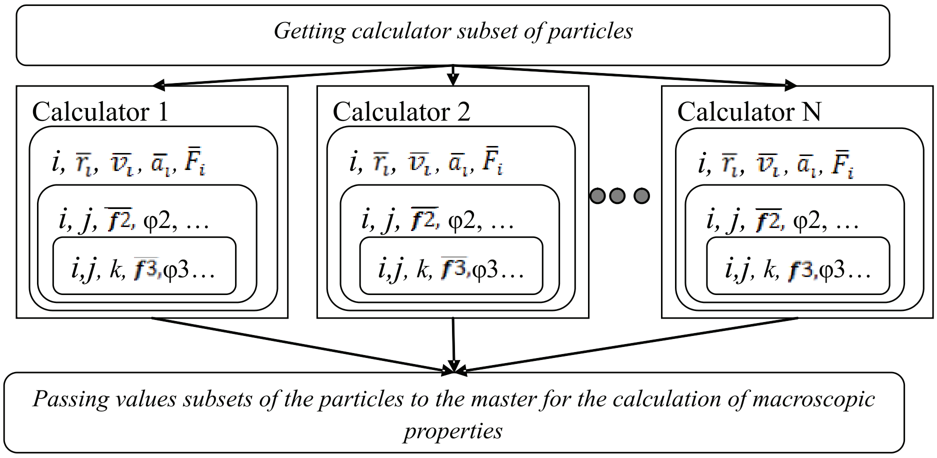
Getting calculator subset of particles
Calculator 1
$i, \bar{r}_i, \bar{v}_i, \bar{a}_i, \bar{F}_i$
$i, j, \overline{f2}$, φ2, …
$i, j, k, \overline{f3}$, φ3…
Calculator 2
$i, \bar{r}_i, \bar{v}_i, \bar{a}_i, \bar{F}_i$
$i, j, \overline{f2}$, φ2, …
$i, j, k, \overline{f3}$, φ3…
●●●
Calculator N
$i, \bar{r}_i, \bar{v}_i, \bar{a}_i, \bar{F}_i$
$i, j, \overline{f2}$, φ2, …
$i, j, k, \overline{f3}$, φ3…
Passing values subsets of the particles to the master for the calculation of macroscopic properties

Fig. 2. Flowchart of the calculators work based on the model of heterogeneous descriptors

The formal description of parallel calculation correlated N-particle system using CUDA and MPI, is given in Fig. 3 and 4 in the form of pseudo-code.

To make parallel calculation of correlated N-particle system using the technology MPI calculators are placed in communication field of MPI_COMM_WORLD. Each calculator is given a unique number, stored in the variable *rank*. Data on the calculators quantity are stored in a variable *size*.

MPI_BCAST function (a variable containing the information of the quantity of descriptors in a subset, quantity of values containing in the variable, variable type, transmitter, communication area, error number) sends to each calculator the number of descriptors for further combining. The calculators independently determine their own subset of the descriptors, using variables StrSml and FshSml. The variables correspond to the start and final value of the subset descriptors and set computing field for calculator.

```
If  rank = master then                            } Definition of power subsets for each
   Call function: subset_range(interval)          } calculator
End if
Вызов функции: MPI_BCAST(interval, 1, MPI_INTEGER, 0, MPI_COMM_WORLD, ierr)
StrSml = (rank*interval)+1                        } Definition of its range for the calculation of
FshSml = (rank+1)*interval                        } each calculator

Do i = StrSml, FshSml                               Calculation  D? 2(i), D? 3(i), {Dlv(i)}
   Do j = 1,n
     Call subroutine: lng-rng_ion                 } long-range ionic and short-range
     Call subroutine: shrt-rng_rplsv              } repulsive interactions

     Call subroutine: 2prt_cvl                    } two and three-particle covalent
     Call subroutine: 3prt_cvl                    } interactions
   end do
 end do

Call function MPI_GATHER (full_force_buf (StrSml: FshSml), interval,
MPI_DOUBLE_PRECISION, full_force, interval, MPI_DOUBLE_PRECISION, 0,
MPI_COMM_WORLD, ierr)
```

Fig. 3. Pseudocode parallel calculation of correlated N-particle system with MPI technology

After completion of computing calculators call the function MPI_GATHER (new values of elements of a subset descriptors *{Dlv(i)}*, number of values, data type, variable for the gather subsets of descriptors on the master, variable contains number of descriptor elements, variable type, communication area, error number) that sends a subset of the one-particle descriptors *{Dlv(i)}*to the master for their combining.

The calculation of the correlated N-particle system at CUDA is divided into two parts. In sequential part there are set of variables(memory GPU). In the GPU memory values of the elements of one-particle descriptors are placed. The subsets of descriptors distributed between processor threads and when the parallel calculation is completed the new values of the elements *{Dlv(i)}* are combined in RAM on the master.

Descriptions are distributed between threads in parallel part. Then each thread conducts its independent calculation and stores the data in memory GPU.

To estimate the efficiency of the distributed calculators model a number of experiments are carried out. The spared time for modeling using MPI and CUDA technologies are compared. Descriptors are filled with test values of the elements. For the model using MPI technology organized cluster containing 16 calculators. Testing of the model using CUDA technology was carried out on the processor GForce GTS 450. The results of computer simulations to model distributed calculators are given in Table 3.

According to the results of the simulation the least time is observed at the calculation of the system GPU GTS 450. At the calculation using MPI technology it is not possible to achieve the acceleration equal to the number of processors because frequent calls to slow speed RAM. In the code on the GPU provides accommodation of variables in a quick shared and constant memory, which reduces the simulation time.

The model of heterogeneous descriptors allows to create a computing model, which is used MPI and CUDA technology for GPU and CPU and allows to obtain experimental results of practical value.

```
Sequential part
Call subroutine: cpu_descriprion ()
 real, device, allocatable :: d1v_d    // Declaring variables in GPU memory
 d1v_d = d1v                          // Copy {D1v(i)} in GPU memory
Call subroutine: gpu_lng-rng_ion <<<N/1024,1024>>>
Call subroutine: gpu _shrt-rng_rplsv <<<N/1024,1024>>>
Call subroutine: gpu _2prt_cvl <<<N/1024,1024>>>
Call subroutine: gpu _3prt_cvl <<<N/1024,1024>>>
d1v = d1v_d                          // Copy new values {D1v(i)} in CPU memory
 end subroutine

Parallel part

subroutine gpu_lng-rng_ion ()
i=(blockidx%x-1)*blockdim%x+threadidx%x
If  i <= N then
Do j = 1,n
……………………
Calculation  D? 2(i),{Dlv(i)}
……………………
 end do
end subroutine
```

Fig. 4. Pseudocode parallel calculation of correlated N-particle system with CUDA technology

Table 3

The results of computer simulations to model distributed calculators

Number particle	Local version	MPI			CUDA
	P = 1	*P* = 2	*P* = 8	*P* = 16	*P* = 1
	Calculation time in seconds				
50176	7,6	5,91	3,72	2,31	0,14
250880	180,4	131,93	88,45	54,68	3,53
401408	548,2	421,64	282,47	156,34	9,11

The model of distributed calculators was being tested win the software package of RIS «MD-SLAG-MELT» [1], [12].

The study was supported by The Ministry of education and science of Russian Federation, project 14.132.21.1792.

Reference

1. Brown W.M., Kohlmeyer A., Plimpton S.J., Tharrington A.N. Implementing molecular dynamics on hybrid high performance computers – Particle-particle particle-mesh // Computer Physics Communications. – Vol. 183(3). – 2012. – P. 449–459.

2. Harvey M.J., De Fabritiis G. An implementation of the smooth particle mesh Ewald method on GPU hardware // Journal of Chemical Theory and Computation. – Vol. 5 (9). – 2009. – P. 2371–2377.

3. Jha P.K., Sknepnek, R., Guerrero-García, G.I., Olvera De La Cruz, M. A graphics processing unit implementation of coulomb interaction in molecular dynamics // Journal of Chemical Theory and Computation. – Vol. 6(10). – 2010. – P. 3058–3065.

4. Le Grand S., Götz A.W., Walker R.C. SPFP: Speed without compromise – A mixed precision model for GPU accelerated molecular dynamics simulations // Computer Physics Communications. – Vol. 184(2). – 2013. – P. 374–380.

5. Peter, E., Pande, V.S. Efficient nonbonded interactions for molecular dynamics on a graphics processing unit // Journal of Computational Chemistry Vol. 31(6). – 2010. – P. 1268–1272.

6. Research-Information System (RIS) «MD-SLAG-MELT»: http://www.nano-md-simulation.com.

7. Voronova L.I., Grigorieva M.A. Information model of physical-chemical melt properties developing for the research complex MD_SLAGMELT"// Interbranch information service. – 2011. – № 2. – P. 30–36.

8. Voronova L.I., Grigorieva M.A., Voronov V.I. Nanostructure computer modeling methods development for multicomponent slag melts // Fundamental researches. – 2011. – № 8 (part 3). – P. 617–622. http://www.rae.ru/fs/?section = content&op = show_article&article_id = 7981162.

9. Voronova L.I.,Grigorjeva M.A., Voronov V.I., Trunov A.S. Computer simulation of the polymerizable oxide melts nanostructure using the descriptor-graph model // Materials Science and Metallurgy Engineering. – 2013. – № 1. – P. 1–12. DOI:10.12691/msme-1-1-1 http://pubs.sciepub.com/msme/1/1/1/#12.

10. Voronova L.I., Grigorjeva M.A., Voronov V.I., Trunov A.S. The program complex «MD-SLAG-MELT» modeling nanostructure and properties of multicomponent melts // Melts. – 2013. – № 2. – P. 1–14.

11. Voronova L.I., Trunov A.S. The parallel algorithm optimization for distributed molecular-dynamic modeling system // Interindustry information service (IIS). – 2011. – № 3. – P. 1–12. http://infogoz.vimi.ru/Izdanie/MIC_03_11.htm#chaper1.

12. Voronova L.I., Trunov A.S. Subsystem of distributed molecular dynamics simulation information research system «MD-SLAG-MELT». – Certificate of state registration of computer program № 2012615017, 5.06.2012.

The work was submitted to the International Scientific Conference «Modern high technologies» Spain-France, July 27 - on August 3, 2013, came to the editorial office on 24.07.2013.

COMPUTER MODELLING OF THERMAL PROCESSES OF ANY DESIGNS OF RADIO-ELECTRONIC MEANS

Shalumov M.A.

The Russian Academy of National Economy and Public Service under the President of the Russian Federation (Vladimir Chapter), Vladimir;
«Scientific research institute» ASONIKA» Ltd., Kovrov, e-mail: ALS140965@mail.ru

At construction of models of thermal processes (MTP) blocks of radio-electronic means (REM) it is necessary to consider all possible kinds of thermal interactions and all possible variants of cooling [1–8]. Calculation for following variants of cooling is provided:

– designs with natural air cooling;

– the designs maintained in vacuum;

– designs with compulsory air cooling (having blown air between printing knots of the block);

– designs with punching;

– designs with a heat-conducting path.

Operating mode of blocks REM can be stationary (not dependent on time) and non-stationary (thermal processes depend on time), therefore at construction MTP it is necessary to consider both variants.

For construction MTP we will consider that the volume of each elementary volume (cube) has identical temperature.

Following kinds of heat exchange are thus considered:

– thermal interaction by means of a conduction between the cubes belonging to one element of a design;

– thermal interaction by means of contact heat exchange between the cubes belonging to different elements of a design;

– return of heat from the cubes which are boundary elements of air in apertures of punching, in environment by means of radiation;

– heat return between the cubes of air which are in the block, by means of radiation;

– return of heat from the cubes adjoining on environment, in environment by means of radiation;

– return of heat from the cubes adjoining on air inside, in this air by means of radiation;

– return of heat from the cubes which are extreme elements outside of the case, in environment by means of обдува (compelled convection);

– return of heat from the cubes which are boundary elements of air in apertures of punching, in environment by means of ventilation (compelled convection);

– return of heat from the cubes which are extreme elements in the case, in air in the case by means of ventilation (compelled convection);

– return of heat of cubes of air which is in the case, among themselves by means of ventilation (compelled convection);

– thermal interaction of the cubes adjoining on the heat-removing basis, by contact heat exchange;

– return of heat from the cubes which are extreme elements outside of the case, in environment by means of natural convection;

– return of heat from the cubes which are boundary elements of air in apertures of punching, in environment by means of natural convection;

– return of heat from the cubes which are extreme elements in the case, in air in the case by means of natural convection;

– return of heat of cubes of air which is in the case, among themselves by means of natural convection.

Scheme of algorithm of the automated synthesis of models of thermal processes of any designs REM is developed for a stationary mode. We will consider algorithm in detail.

Block 1. Constant ambient temperature is set.

Block 2. The cycle on axes X in which the design is looked through on length with the step equal to the party of a cube is set.

Block 3. The cycle on axis Y in which the design is looked through on depth with the step equal to the party of a cube is set.

Block 4. The cycle on axis Z in which the design is looked through on height with the step equal to the party of a cube is set.

Block 5. A conduction between elements of one firm body. The cubes belonging to one element of a design, co-operate among themselves a conduction.

Block 6. Contact heat exchange between adjoining elements of different firm bodies. The Cubes belonging to different elements of a design, co-operate among themselves by means of contact heat exchange.

Block 7. Radiation: a design with air outside of the block. Boundary cubes of a design co-operate with air outside radiation. If cubes are boundary elements of air in punching apertures they also co-operate with air outside radiation.

Block 8. Radiation: a design with air in the block. Extreme cubes in a design co-operate with cubes of air adjoining on them inside radiation.

Block 9. Radiation: air with air in the block. Cubes of air in the block co-operate among themselves radiation.

Block 10. Check on presence of natural cooling. In case natural cooling is available, blocks 11–13 (convection) are carried out.