

Particle System Dynamics Software for the Design of Constructional Composites

Vladimir A Smirnov, Evgeniy V Korolev and Alexandr V Evsigeev

Moscow State University of Civil Engineering

Abstract: Particle system dynamics is one of the most general simulation methods. Different terms are used for denoting this method: “many body problem”, “molecular dynamics” and others. The underlying algorithms of the particle system dynamics are currently implemented in many software packages, including packages specifically targeted to the parallel computational environments. However, the cases can be encountered in the constructional material science when force fields are quite complex and are hard to represent within existing software. In this work we have made the overview of the modeling tools available for the numerical solution of tasks in material science; the overview is based on the current theory and production technology of building materials. Two open source molecular dynamics software packages are examined; common features of such packages are outlined. Then we describe our implementation which is tailored for the specific tasks of the constructional material science.

Keywords: particle system dynamics software, numerical simulation, composite materials, multiscale modeling

1. Introduction

Modern construction is in need of materials with predetermined set of operational properties. The traditional way of R&D in material science of construction is heavily based on experimental design, regression and further analysis of the obtained models; this well-known method is often denoted as “statistical modeling”. Correct application of mathematical procedures allows prediction of properties with predefined confidence. The latter depends on experimental conditions; as a rule, high level of confidence requires extra resource consumption and costly experimental setups.

This is why during decades there is a constant interest to “computational” design of materials – computational material science. A world of research papers, thesis and monographs on the subject are published; software for various models and algorithms are developed (and are often available in source form). Constructional material science takes advantages of plethora of models, algorithms and software tools already developed in other branches of science – mostly in theory of condensed state (“clean” models based on quantum mechanics; “ab initio” models).

Still, in day to day practice of R&D in material science of construction we can frequently observe the gap between traditional and numerical experiments. The gap is due to the fact that no theoretical description of building material currently allows to derive a model for assured prediction of important operational properties (compressive strength, water and chemical resistance etc.). At most, numerical simulation is used for reduction of dimension and measure of factor space, eliminating knowingly unpromising mixture designs. And proper selection of modeling methods and simulation software is not trivial, even for aforesaid aim.

Several steps have to be taken to facilitate the adoption of numerical methods during design of construction materials; the first step is formulation of productive classification which allows to oversee the entire set of available modeling methods and computational software. In the next section we propose the classification that is tightly coupled with current production technology of building materials.

2. Polystructural Theory of Constructional Composites

Representation of building material in terms of appropriate mathematical model is central to successful application of numerical methods. Building materials are heterogeneous disperse systems (except for some

peculiar cases, e.g. constructional glass) formed during hardening (setting) of technological mix; even constructional steel can be considered as a composite. Unfortunately, as it was already stated, thermodynamics and physical chemistry of colloids are only of limited use during prediction of operational properties. Prevalent method to control the operational characteristics consists in alteration of mixture (specific surface of the filler, grain size distribution of aggregates, rheology of the binder) and technology (mode of laying, compaction and hardening). Mixture and technology are tightly connected to each other and usually cannot be selected independently; there exist groups within each there is a predominant interaction (which is caused by mixture and composition) determines not only applicable technology, but also suitable modeling methods. These groups loosely correspond to spatial scale, but still defined by prevalent interaction; depending on the characteristic size of the system, different techniques of numerical analysis can be applied – methods based on quantum and classical mechanics; geometry and probability theory; continuum mechanics (Table 1).

TABLE I: Spatial scales, interactions and modeling software in material science of construction

Spatial scale	Level of structure	Theory, models, modeling software
~ 1 m	Homogeneous structure	Continuum mechanics, finite difference and finite element representations. Numerous verified and certified computational tools are available.
> 0.1 mm	Macrostructure: gravitation and technological impacts	Geometry of the system and laying of elements may be investigated with statistical simulation. Few examples of particle dynamics examination are known. Computation tools for constructional material science either unavailable or unsuitable for productive R&D.
0.1-100 um	Microstructure: surface effects	Particle systems with classical mechanics and statistical simulation combined with analysis of ensemble geometry. Numerous general-purpose computation tools for classical mechanics of particle systems. Specific software implementations for constructional material science have to be developed.
1...100 nm	Nanostructure: size effects	Particle systems with classical mechanics can be used in conjunction with quantum chemistry methods. Models for statistical simulation and analysis of ensemble geometry are mostly unknown. Theory in active development, semi-empirical models are also in use. Software tools for neighbor spatial scales can be utilized (e.g. both quantum chemistry and particle system dynamics).
0.1...1 nm	Atomistic level: quantum effects	Quantum chemistry methods. Many computational tools are available.

The first row of Table 1 may correspond to distinctive spatial levels and reflects the fact that in some circumstances heterogeneous structure of constructional composite may be ignored. This fact is the fundamental assumption for production-level software packages currently used for design in construction. If only macroscopic properties – strength, modulus, Poisson ratio – are considered, than a lot of discrete models developed in applied mathematics (finite difference, finite element, boundary element, finite volume) can be applied for investigation of stress-strain state, acoustics, conservative transfer (heat, mass), hydrodynamics and aeroelasticity.

The next two structural levels – macro- and microstructure – are of great interest for material scientists [1]. Microstructure of the building material is the spatial level where structure formation is strongly affected by surface effects (free surface energy, surface tension, wetting). This level is formed when binder is combined with fine filler (specific surface of such filler is usually above 100 m²/kg and corresponding particle size is below 10 um). Numerous processes take place during hardening of microstructure; cluster formation [2,3] is one of such processes.

Macrostructure of the building materials is defined as a level where structure formation is primarily affected by weight of the particles and technological actions (pressure, vibration). Macrostructure is formed when coarse grained (one millimeter and above) aggregates are combined with microstructure. Exact spatial boundary between macro- and microstructure is not strictly defined; it is near 100 um.

Since at the macroscopic level the main type of interaction is the interaction caused by gravity force, the motion equations for elements of the system are simple. These are equations of classical mechanics. In

constructional material science, transient problem of structure formation process reflects technology of preparation and laying of constructional mix. If this problem is out of scope for the research, then modeling is often preformed only to determine the final configuration of coarse aggregates. The corresponding problems are theoretical examination and statistical simulation of dense packing in polydisperse systems, as well as percolation problems for frame of aggregates (the results can be used for prediction of mechanical properties) and pores (properties of porous space affects permeability and coupled properties – water, chemical and frost resistance). These problems has long history (at least, starting from early XX century [4,5]). Theoretical aspects of packing problem for polydisperse spheres and particles with complex shape were discussed in many research works due to importance of method which allows to design an ideal particle size distribution for constructional mix. Various geometric models were proposed, and different methods from percolation theory applied during simulation of packing. But surprisingly, despite the great interest to the packing and percolation problems at the macroscopic level of constructional composite, there are no well known software tools for simulation in this area. Tools with quite limited functionality were created mostly by authors of algorithms; such tools often requires mutually incompatible formats of input and output data, and implementation of algorithms is not verified extensively. In no conditions such software tools can be considered as production-ready, even if some of them proposed for commercial use.

To adequately represent the microstructure of the disperse system with fine filler, not only gravity force must be taken into account; moreover, gravity force can usually be ignored during numerical experiments. The primary forces are caused by surface properties of the disperse phase and free energies on the inter-phase boundaries. Within the framework of particle system dynamics (more on this below) the surface effects can be represented by pairwise forces with semi-empirical potentials.

The last row of the Table 1 corresponds to atomistic scale. It is the domain of quantum mechanics: there are no particles; there are only waves and probability. Time-independent Schrödinger equation $H\psi=E\psi$ is very simple in form – it states that valid energy E of the system must be eigenvalue of Hamiltonian H , and corresponding probability (“shape” of orbital) should be square $\psi\psi^*$ of the eigenfunction ψ (wave function). However, “solution” of the Schrödinger equation – even in time-independent case – is not a simple task; many complex methods were specifically designed for. These methods often referred as quantum chemistry methods, and corresponding software is called quantum chemistry software.

The fourth row of Table 1 corresponds to nanoscale level. “Nanostructure” of constructional composite is the spatial level where properties of material are considerably affected by size effects. Submicrocrystalline structures of hardening inorganic binder and molecular globules of thermosetting polymers are examples of formations at the nanoscale level. Software tools for neighbor spatial scales can be utilized during investigation of nanostructure. For instance, modeling based on the analysis of the ensemble geometry can be used for solution of percolation problem at nanoscale level [6].

3. Particle System Dynamics

The system under investigation can be discretized by means of various ways. Three main types of discrete representations are mentioned in Table 1 – finite difference, finite element and particle system. The latter type is usually used during evolutionary modeling (solution of time-domain Cauchy problem).

Representation in form of particle system has many advantages: simplicity of the underlying theory (in many cases this is classical dynamics); natural representation of many heterogeneous systems (e.g. building materials); ability to model arbitrary pairwise forces (that cannot be represented by harmonic potential); possibility to investigate large and nonlinear deformations, crack propagation and contact interactions; presence of efficient parallel algorithms. The known disadvantage is tremendous amount of computational operations, especially for “direct” particle-particle models that represent any physical particle by computational one.

Most pioneering works on particle system dynamics are dated back to 1960-s (the first numerical investigation in the field concerning irreversibility paradox [7] was reported by Fermi, Pasta and Ulam in 1953

[8]). These works are primarily inspired by problems arising in astrophysics and thermonuclear fusion. Significant results summarized in [9]; in particular, attention was made to integration schemes suitable for particle dynamics. During the course of elaboration, particle system dynamics is segregated into several areas with designated set of models – molecular dynamics (additional semiclassical provisions from quantum physics), dynamics of complex and non-Newtonian fluids, smoothed particle hydrodynamics and more [7-11].

4. Particle System Software

As it can be concluded from previous brief review and relevant works [7,9,11,12] in the field, solution of set of ordinary differential equations (ODE) is the essence of any particle system dynamics software. The model is based on classical mechanics; this model includes potential (force field) representing pairwise interaction. The derivative of pairwise potential along the direction between two particles is equal to binary forces acting between these particles; this force is used during solution of the motion equations. After the model is constructed and parameters of pairwise potential are determined, many existing software packages can be used for solution of motion equations.

However, this simple representation of the particle system dynamics pays no attention to the many important details of the software implementation. At the moment, many of such details are already revealed by the authors of most complete particle system dynamics software packages; other details and requirements arises during distinctive demands of material science.

4.1. Available Open Source Software Packages for Particle System Dynamics

First of all, it must be noted again that there are numerous software packages for particle system dynamics currently exist, both commercial and open source.

Wide availability of the software tools is due to rather curious fact that so-called *N-body problem* (particle system with pairwise forces and classical equations of motion) was first occurred long ago in spatial scale that was completely different from ones mentioned in Table 1, when motions of a group of celestial objects were studied. Thus, at present there are already a lot of efficient algorithms and highly-optimized computer code [13] for the purpose. The comprehensive list of resources concerning software suitable for N-body simulation can be easily obtained [14]; and, as of 2015, at the top of the search query [14] there are links related to the high-performance (GPU-accelerated) code.

Advanced particle system dynamics software packages are designed mostly for the modeling at the atomic and nanoscale levels. Concerning open source implementations, at least two of such packages have to be mentioned – “Groningen machine for chemical simulation” (GROMACS) and “Large-scale atomic/molecular massively parallel simulator” (LAMMPS).

The development GROMACS [15-18] molecular dynamics package was started at the University of Groningen in the early 1990s [15]. The package is the set of command line driven tools (about one hundred utility and analysis programs) for POSIX environment. It was initially a part of the parallel hardware project with the same name. The hardware part of the project was obsoleted over time (it was based on the Intel i860 processors [16]), while the simulation software, being properly written, is proven to be significant as a standalone entity. The key features of the GROMACS package are the follows.

- Using of the programming language with extreme level of portability – ANSI C. Because of such a choice of programming language and portable design, GROMACS package is now available in binary form for many Linux distributions (Debian etc.) and can also be easily compiled from source for different POSIX environments. In particular, it is included in FreeBSD ports collection and can be built from similar source-based repositories (e.g. Arch Linux AUR).
- Suitability for high-performance computing (HPC) environments. GROMACS was inherently designed with HPC in mind; it includes numerous parallelization techniques [18].

- GROMACS can handle pair interactions with Coulomb and Lennard-Jones or Buckingham potentials [16], polarizable shell models and flexible constraints [17].
- Force routines in various representations can be added by the user [15, 17].
- There is a free web based database of molecular data specifically created for the package [19].

LAMMPS [20] was designed and implemented in Sandia National Laboratories in the mid 1990s. Early versions of LAMMPS were written in FORTRAN. Starting from the first open source release in 2004, the code is rewritten in C++. The current key features of the LAMMPS are similar to ones of the GROMACS; in particular, great attention was paid to the special-purpose parallelization algorithms [21,22]. It is also stated that LAMMPS was originally designed for materials applications and offers a convenient mechanism for code modification [11].

The overall portability of the LAMMPS is somewhat less than one of the GROMACS; e.g., LAMMPS is not maintained within FreeBSD ports collection. This fact brings back to mind several arguments about relative portability of programs written in C and C++; in short, “C is still portable, while C++ is already not”. This statement, though formulated twenty years ago for Microsoft Windows environment [23], perfectly applies to the current state of software design.

4.2. Special Implementation for the R&D in Construction Material Science

Obviously, features of the packages similar to the GROMACS and LAMMPS make them suitable for almost every branch in material science. However, concerning the primary criterion of usability [24]

$$\sum_{i=1}^N t_i \rightarrow \min, \quad (1)$$

where t_i are total amounts of time required to exploration of features, development or extension, training and application, the decision was made to implement necessary functionality from scratch.

At present, there is a wide variety of textbooks and other references concerning software design. At the same time it should be noted that many key requirements were formulated decades ago and can be found even in monographs on numerical modeling; [9] is, again, an excellent example. In general, key requirements for the design are similar to ones that were already mentioned above as features of the GROMACS and LAMMPS packages, though suitability for HPC was not in the priority list.

There are cases when pairwise forces between particles of the filler in constructional composite are too complex and depend not only on distance between surfaces of particles. Such cases cannot be adequately handled by existing ODE software packages. Nevertheless, high availability and open source license for general-purpose numerical libraries allow to implement only “material science” part of the code – a part which is solely based on geometry, physical chemistry and thermodynamics, leaving the underlying mathematics aside. As an ODE solver we have used embedded Runge-Kutta with error estimation [13]. Such choice significantly improves the stability of the solution process in case of complex pairwise and boundary forces, albeit it negatively impacts the performance. The latter is currently at least an order of magnitude worse than the performance of packages similar to GROMACS, even in sequential case.

As an example of complex pairwise force we can consider force caused by solvation shells of the particles. The particle with solvation shell can be modeled as two concentric spheres composed with rigid (impenetrable) core and viscous shell; such model is general enough to represent both micro- [25] and macrostructure level of constructional composite. The viscous force can be estimated as a value proportional to cross section of the shells (outer spheres). The direction of the force is collinear to the relative velocity $\dot{\mathbf{r}}_{ij}$ of the particles but not to the vector \mathbf{r}_{ij} drawn between particles; such force can not be represented as a (spatial) gradient $-\nabla U$ of the unknown potential U . There are also other types of force fields (both pairwise and volumetric) that represent technological impacts, e.g. pressure and vibrating at the compaction stage of constructional mix; many of such types are implemented in the developed software.

Solution of ODE for dynamical systems requires appropriate initial and boundary conditions; this is especially important for many strongly nonequilibrium problems arising in constructional material science. For “perfect” solids (crystals) simple algorithms can be used to generate fragments of common (FCC, BCC, etc.) lattices. To model the initial distribution of the dispersed phase in constructional mix it is necessary to employ several statistical densities which correspond to mixing conditions. Together with statistical distributions we have used constructive solid geometry (CSG) to model the boundary. The representation of the initial state in constrained dynamical problem has to be written in textual form. The example of such representation is shown below:

```
union
{
  cylinder [-101,0,0;1,0,0] [30;202]
  union {
    cylinder [0,-101,0;0,1,0] [30;202]
    union {
      cylinder [0,0,-101;0,0,1] [30;202]
      sphere [0,0,0] 60
    }
  }
}
```

Such fragments are embedded in the task file (the solver is command line driven tool). During the parsing of the task file the CSG trees (currently only binary) are constructed as necessary and particle generation performed for the selected types of spatial distributions. The examples of generated systems are shown of the Fig. 1 and 2.

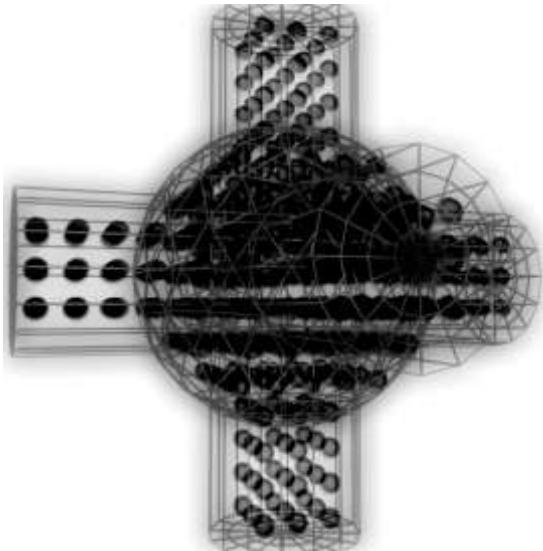


Fig. 1: Regular cubic lattice for the listed CSG code.

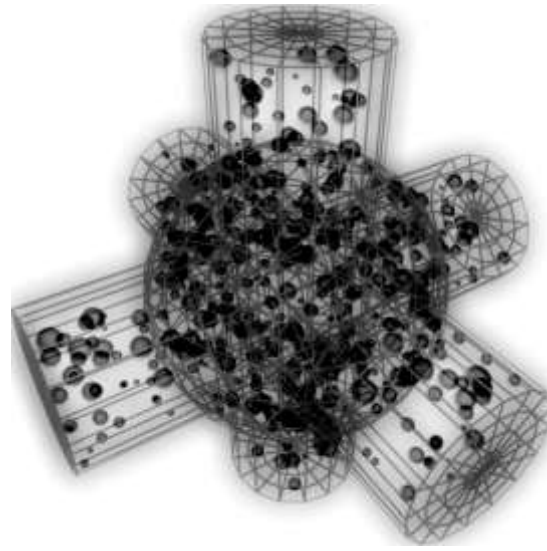


Fig. 2: Uniform distribution for the listed CSG code.

Visualizations shown on the Fig. 1 and 2 were carried out with help of 3DS MAX. Both pre- and postprocessors of the dynamics package are able to export representation of the system in form of the MaxScript program which can be used for scene generation and rendering. The visualization functionality is also implemented in solver (we have used portable OpenGL interface for rendering), though quality and detail of the produced images are far from typographic. Such visualization is mostly useful for monitoring of the solution process.

First implementation of the software was performed for the 32-bit Microsoft Windows environment. The software is entirely written in ANSI C. The portability is one of the primary aims; there are several abstraction layers. Lower layer serves as isolation from the Windows API calls; this layer is very thin, and during the course of the development was extended for the POSIX environments.

Middle layer is introduced as a support measure for object oriented programming in object agnostic ANSI C language. We believe that proper object oriented software design is far more important than using modern fashionable “object oriented” programming dialects; good object oriented design can easily be implemented in C language. Total amount of code is almost the same, while porting and maintaining are simplified significantly. Middle layer (middleware) incorporates basic algorithms, auxiliary functionality, lexical analysis and ODE solving algorithms. Latter are exposed through small number of interfaces (`odeCreateSolver()`, `odeSet()` / `odeGet()` and `odeIntegrate()`) which take opaque handle and pointer to the ODE right side; such implementation allows to exchange the solver without any alteration of the “applied” functionality.

The developed software was successfully used in several basic and applied research works. As an example of so-called *multiscale modeling*, when modeling results from one spatial level are used in another level (both as initial conditions and during the numerical simulation), the structure formation of the framework constructional composites [26] can be considered. The framework composite is the building material produced according to two-stage technology: on the first stage the permeable framework of aggregates is created, and during the second stage free space in the framework is filled with matrix material. The modeling of the structure formation was also carried out in two stages. The formation of the framework is modeled as a dynamics of aggregates enclosed in solvation shells with technological force fields. At the second stage the stabilized framework of coarse grained particles used as a boundary during the numerical simulation of filtration process. The characteristic time of the impregnation process is one of the results of the numerical simulation.

5. Conclusion

Due to the fact that particle system dynamics is widely used for numerical modeling in material science, many software packages are developed for such simulations. However, in constructional material science we constantly encounter cases when adoption of the existing software leads to unacceptable time consumption for exploration of features and extending the functionality; because of this, at some point the decision was made to implement the software tailored for the specific tasks of construction. The developed software is of layered object oriented design; it is implemented entirely in ANSI C and uses portable programming interfaces. Currently it can be compiled for Microsoft Windows (with Microsoft Visual C, Open Watcom and GNU C compilers) as well as many POSIX (in particular, Linux and FreeBSD) environments. This software was successfully used in several basic and applied research works in material science.

The goals of the further software development are: parallelization of algorithms; code cleanup; implementation of the advanced visualization features; resolving the licensing issues (the latter requires rewriting several important parts of the ODE solver).

6. Acknowledgements

This work is supported by the Ministry of Science and Education of Russian Federation, Project #2014/107 “Structure formation of sulfur composites: phenomenological and ab initio models”.

7. References

- [1] Yu.M. Bazhenov, I.A. Garkina, A.M.Danilov and E.V Korolev, *System analysis in Construction Material Science*. Moscow: Moscow State University of Civil Engineering, 2012. (in Russian)
- [2] E.V. Korolev, V.A. Smirnov, A.P. Proshin and A.M. Danilov, “Modeling of Evolution of Liophobic Disperse Systems,” *News of Higher Educational Institutions. Construction*, 8, pp. 32-38, 2004. (in Russian)
- [3] I.A. Garkina, A.M.Danilov, E.V Korolev and V.A. Smirnov, “Flocculation in Disperse Systems,” *Regional Architecture and Construction*, 1, pp. 124-131, 2008. (in Russian)

- [4] W.B. Fuller and S.E. Thompson, "The Laws of Proportioning Concrete," *Trans. of American Society of Civil Engineers*, vol. 59, pp. 67-143, 1907.
- [5] C.C. Furnas, *Flow of Gasses Through Beds of Broken Solids*. Bureau of Mines Bulletin, 1929.
- [6] V.A. Smirnov, E.V. Korolev and A.I. Albakasov, "Size Effects and Topological Characteristics of Nanomodified Composites," *Nanotechnologies in Construction*, 4, pp. 17-27, 2011.
- [7] W.G. Hoover, *Molecular Dynamics*. Berlin: Springer, 1986.
- [8] E. Fermi, J. Pasta and S. Ulam, *Collected Papers of Enrico Fermi*, vol. 2. Chicago: Chicago Press, 1965, pp. 978-987.
- [9] R.W. Hockney and J.W. Eastwood, *Computer simulation using particles*. Bristol: IOP Publishing, 1988.
<http://dx.doi.org/10.1201/9781439822050>
- [10] G. Ciccotti and R. Kapral, *Theory and Applications of Computational Chemistry: The First Forty Years*. Amsterdam: Elsevier, 2005, pp. 425-440.
- [11] B. Leimkuhler and C. Matthews, *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Heidelberg: Springer, 2015.
<http://dx.doi.org/10.1007/978-3-319-16375-8>
- [12] J.M. Haile, *Molecular dynamics simulation*. New York: Wiley, 1997.
- [13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge University Press, 1992.
- [14] Search query "List of N-body simulation software" [Online]. Available: [https://www.google.ru/search?q=list+of+"n-body"+simulation+software](https://www.google.ru/search?q=list+of+)
- [15] H. Bekker, H.J.C. Berendsen, E.J. Dijkstra, S. Achterop, R. van Drunen, D. van der Spoel, A. Sijbers, H. Keegstra, B. Reitsma, M.K.R. Renardus, "Gromacs: A Parallel Computer for Molecular Dynamics Simulations," in *Proc. 4th Intl. Conf. Physics Computing*, 1993. pp. 252-256.
- [16] H. Berendsen, D. van der Spoel and R. Vandrunen, "GROMACS – a Message-passing Parallel Molecular-Dynamics Implementation," *Comput. Phys. Commun.*, vol. 91, pp. 43-56, 1995.
[http://dx.doi.org/10.1016/0010-4655\(95\)00042-E](http://dx.doi.org/10.1016/0010-4655(95)00042-E)
- [17] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A.E. Mark and H.J.C. Berendsen, "GROMACS: Fast, Flexible and Free," *J. Comp. Chem.*, vol. 26, pp. 1701-1718, 2005.
<http://dx.doi.org/10.1002/jcc.20291>
- [18] B. Hess, C.Kutzner, D. van der Spoel and E. Lindahl, "GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation," *J. Chem. Theory Comput.*, vol. 4 (3), pp. 435-447, 2008.
<http://dx.doi.org/10.1021/ct700301q>
- [19] D. van der Spoel, P.J. van Maaren and C. Caleman, "GROMACS Molecule & Liquid Database," *Bioinformatics*, vol. 28 (5), pp. 752-753, 2012.
<http://dx.doi.org/10.1093/bioinformatics/bts020>
- [20] LAMMPS History. [Online]. Available: <http://lammps.sandia.gov/history.html>
- [21] S. Plimpton, "Fast Parallel Algorithms for Short-range Molecular Dynamics," *J. Comput. Phys.*, vol. 117 (1), pp. 1-19, 1995.
<http://dx.doi.org/10.1006/jcph.1995.1039>
- [22] S. Plimpton and B. Hendrickson, "A New Parallel Method for Molecular Dynamics Simulation of Macromolecular Systems," *J. Comput. Chem.*, 17, 3, 1996, pp. 326–337.
- [23] L. Grinzo, *Zen of The Windows 95 Programming*, Coriolis Group, 1995.
- [24] V.A. Smirnov, E.V. Korolev and O.I. Poddaeva, "Language for Scientific Imaging: Design, Implementation and Applications," presented at the International Conference on Data Mining, Electronics and Information Technology (DMEIT 2015), Pattaya, Thailand, August 10-11, 2015.
- [25] V.A. Smirnov, E.V. Korolev and A.S. Inozemtcev, "Dynamic Simulation of Nanoscale Systems," *Nanotechnologies in Construction*, vol. 4 (3), pp. 26-34, 2012. (in Russian)
- [26] E.V. Korolev, A.P. Samoshin, V.A. Smirnov and O.V. Koroleva, *Radiation-Protective Framework Composites with Thermoplastic Matrices*, Penza: Penza State University of Architecture and Construction, 2010. (in Russian)