Hindawi
Complexity
Volume 2018, Article ID 5083247, 16 pages
https://doi.org/10.1155/2018/5083247

Research Article

Integrated Modeling, Simulation, and Visualization for Nanomaterials

Feiwei Qin[1], **Haibin Xia**[2], **Yong Peng**[1] and **Zizhao Wu**[3]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
[2]*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China*
[3]*School of Media and Design, Hangzhou Dianzi University, Hangzhou, China*

Correspondence should be addressed to Feiwei Qin; qinfeiwei@hdu.edu.cn

Received 23 October 2017; Accepted 19 March 2018; Published 24 April 2018

Academic Editor: Vittorio Loreto

Computer aided modeling and simulation of nanomaterials can describe the correlation between the material's microstructure and its macroscopic properties quantitatively. In this paper, we propose an integrated modeling, simulation, and visualization approach for designing nanomaterials. Firstly, a fast parametric modeling method for important nanomaterials such as graphene, nanotubes, and MOFs is proposed; secondly, the material model could be edited adaptively without affecting the validity of the model on the physical level; thirdly a preliminary calculation for nanomaterials' energy is implemented based on the theory of surface fitting; finally, an integrated framework of nanomaterials modeling, simulation, and visualization is designed and implemented. Experimental results show that the proposed approach is feasible and effective.

1. Introduction

Design and optimization of materials are eternal and new problems; almost all matter is made up of materials. The emerging nanotechnology which changes the distribution and arrangement of atoms to get materials of different properties has brought significant revolution for materials' design. Nowadays nanotechnology has been extensively applied in various fields such as medical, aerospace, and energy [1–3].

Computer modeling and simulation of nanomaterials quantitatively describe the correlation between the materials' microstructure and macroproperties to a certain extent [4]. The strong computing power provided by computer overcomes the shortcomings of traditional material design and reduces the material production cycle [5–7]. In the 21st century, computer aided design is playing a huge role in designing nanomaterials [8, 9]. However, the existing nanomaterials structure modeling software and simulation software have some problems. On one hand, the modeling and simulation systems have their respective emphases. The modeling software often lacks material simulation, and the simulation software lacks effective modeling [10–12]. On the other hand, structure modeling and material simulation are integral parts for material research. Researchers usually need to transfer data between the two systems iteratively, which is time-consuming [13]. Hence it is vital to develop an integrated modeling and simulation platform for nanomaterials.

The primary target of our research is to optimize the design of the material structure through modeling and simulation in nanoscale. We propose an integrated modeling and simulation approach for nanomaterials' design, to assist researching on nanomaterials' structure and macroperformance.

2. Related work

Currently nanomaterials' modeling and simulation have become necessary tools for predicting molecular motion. There are many related commercial software programs and tools. However, the existing software programs are not functionally complete enough. Some of them are modeling systems, and the others are visualization software programs. Meanwhile, these software programs are very expensive.

Structure Modeling. Researchers build geometric models of different structures firstly, then validate the performance of

materials through computer simulation, and finally optimize materials' geometric structures. Overvelde et al. [14] design different pore materials by changing holes' shape, size, and arrangement and verify materials' properties by employing finite element analysis and physical experiments. Also, though atom reconstruction is essentially a problem of K nearest neighbor search, some algorithms such as KD-tree and octree are studied for modeling atoms. Lou et al. [18] use KD-tree based space partitioning algorithm to merge finite element triangle meshes for fast prototyping. Muja [19] approximates nearest neighbors with automatic algorithm configuration. The adopted algorithm is based on hierarchical K-means trees. Ooi et al. [20] use KD-tree for indexing spatial databases.

Simulation. Material Studio is a relative mature modeling and simulation software for nanomaterials [15]. Some nanomaterials could be modeled and simple attribute calculation can be carried out on this software. However most simulation functions need to be purchased. The compatibility between Material Studio and other software programs is also a big problem. LAMMPS [16] is a molecular dynamics simulation software developed by Sandia laboratory with C++ language. It supports simulation for millions of molecules in form of gas, liquid, and solid. However, visualization interface is not provided. The user cannot watch the entire modeling process intuitively. In addition, interactive operation modeling is not supported.

Recently some researchers use GPU to accelerate the modeling and simulation for nanomaterials. Stone et al. [21] concluded that the era of accelerating molecular simulation by utilizing GPU has come. The effects of simulation for nanoscale molecule are very well with the aid of GPU's parallel computing ability.

Visualization. RasMol is a commonly used software for visualizing nanomolecular model [17]. It provides powerful and convenient display interface for operation. A variety of types of files such as proteins, DNA, and big molecule can be loaded into this software. Some simple calculation such as counting numbers of amino and carboxy can also be proceeded. However the simulation of nanomaterials cannot be carried out. Hence RasMol is just a display tool. Similar toolkits such as VMD [22] and ICMLite also provide well displayed interfaces. The designer can draw sketches based on these toolkits. However the function of material simulation is absent.

There are some visualization and analysis software programs focusing on some special material's molecule. For example, ANTHEPROT [23] and VHMPT are used for analyzing proteins, and RNAstructure and RnaViz are used for analyzing RNA. These software programs have powerful visualization abilities and provide simple functions for calculating attributes. However they are only used for some special materials; the applicability is not comprehensive.

3. Structure Modeling for Nanomaterials

Structure modeling is the basis for researching correlation between nanomaterials' structure and properties. The main problem is that analyzing the structural rule of complicated nanomaterials is difficult. Hence modeling many nanomaterials automatically by program is hard; furthermore, the number of nanomolecules is huge, which can reach hundreds of millions. The algorithm and the performance of the computer are very demanding. Aiming at the aforementioned problem, we use adjacency list to provide parametric modeling for common nanomaterials, supporting the construction of macromolecular material model.

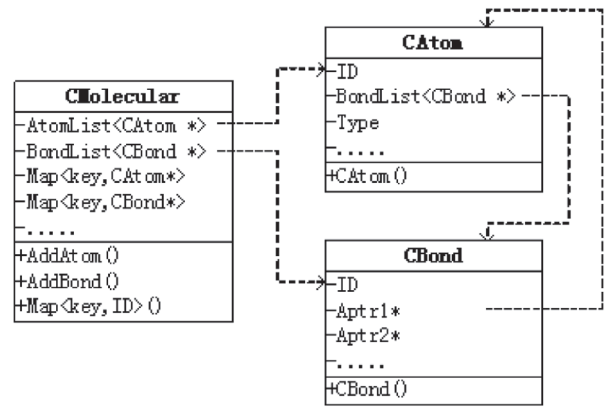


Figure 1: Data structure of nanomolecules.

3.1. Modeling Based on Adjacency List. For supporting macromolecular model and interactive modeling, data structure of molecular nanomaterials must have these functions: finding adjacent atoms quickly; assembling and constructing nanomaterials model rapidly; mapping geometric structure and topological structure rapidly. Hence we design nanomolecules' data structure based on adjacency list. It stores the relevant physical information of the atom, and mappings between geometric model and materials' internal structure, and supports to reconstruct macromolecular model quickly. The main idea is: each vertex represents an atom and each edge represents the chemical bond between atoms, as shown in Figure 1. The concrete structure is as follows:

(i) Nanomolecule is composed of atoms and chemical bonds.

(ii) Each molecule uses two continuous spaces (AtomList and BondList) to save all the atoms and bonds, respectively.

(iii) Each atom has a type, and a BondList to save all the chemical bonds and other related information.

(iv) Each bond has a type, and the two connected atoms.

Figure 2 shows an example of molecular structure which is represented by the adjacency list. It has four atoms, A, B, C, and D, and three bonds, 1, 2, and 3. Among them atoms B, C, and D only have one bond. Furthermore, there is a problem about mapping between geometric shape and topological structure when editing the material model interactively. For example, if we delete a sphere with mouse, how to delete its

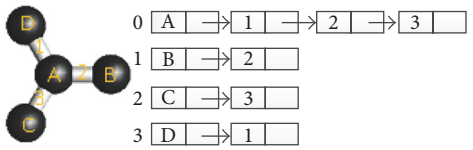


Figure 2: Molecular structure representation based on adjacency list.

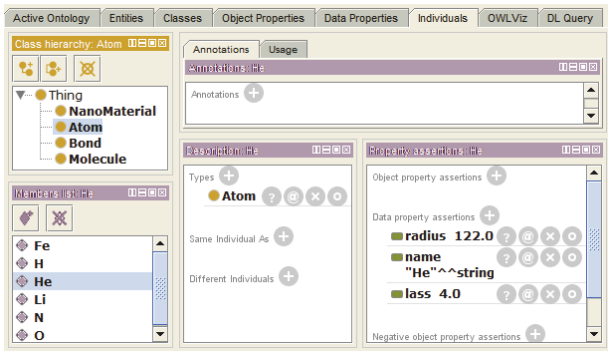


Figure 3: OWL file for saving atom properties.

corresponding atom in topology? For solving this problem, the molecular structure representation also saves mapping between geometric information and molecule's topological structure. Hence changes on geometric shape can update to nanomolecule's structure instantly.

The basic properties of atom such as mass and radius are usually neglected during structure modeling for nanomaterials. These properties are insignificant for modeling but necessary for simulation. If these properties are added during the modeling process, a huge number of repeating atoms' property information will waste memory. Hence we define some common atoms' basic properties according to OWL format and store them in local file [24–26]. The atoms' information could be retrieved through their types during the simulation process. Figure 3 shows the OWL format for saving atoms' property information including their names, types, radii, and lass.

3.2. Macromolecule Reconstruction Based on Space Partition. Different from other format files, for nanomolecule model file, there is the special problem of how to reconstruct nanomolecule model which has huge atoms rapidly. The existing model file only stores atoms' coordinate information without chemical bond information. For example, graphene is composed of single atom. The bonding distance is only 0.142 nanometers. Its representation such as *XYZ* file does not have bond information that needs to be recovered. If we compute the distance for each atom to other atoms, the time complexity is $O(n^2)$, not suitable for macromolecule reconstruction.

Analyzing molecule's structure, we find that actually each atom only needs to be compared to its neighboring atoms. Though which atoms are neighboring is unknown, we must traverse all the atoms. That is the reason why the algorithms for reconstructing macromolecule have high time

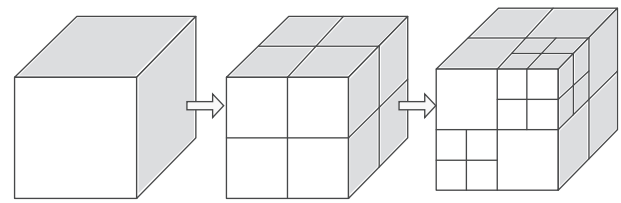


Figure 4: Octree based space partition.

complexities. Hence we can subdivide a molecule model into several parts with space partition method [18–20] and search neighboring atoms through space adjacent relations.

Octree is a typical space partition method. It divides a cube into eight small cubes and then subdivides them repeatedly according to requirements, as shown in Figure 4. When carrying out nearest neighbor search, we search 12 adjacent edges, 6 adjacent faces, 8 adjacent vertices, and the 26 corresponding adjacent cubes for each atom. Each cube has a unique serial number, which is obtained by calculation.

The algorithm for macromolecule reconstruction is now described. The open source library PCL [27] is used for building the octree in this paper.

Algorithm I. Macromolecule reconstruction based on octree space partition.

(1) Initialize the octree. Traverse each atom and insert it into the octree according to its coordinates. The operation is implemented by using function *OctreePointCloudSearch<AtomPosList>*.

(2) Traverse each atom and search their neighbors within the neighborhood of radius *R*. Atoms only bond in a given distance *R*, as long as they find the distance *R* of nearby atoms. This operation is implemented by function *RadiusSearch*.

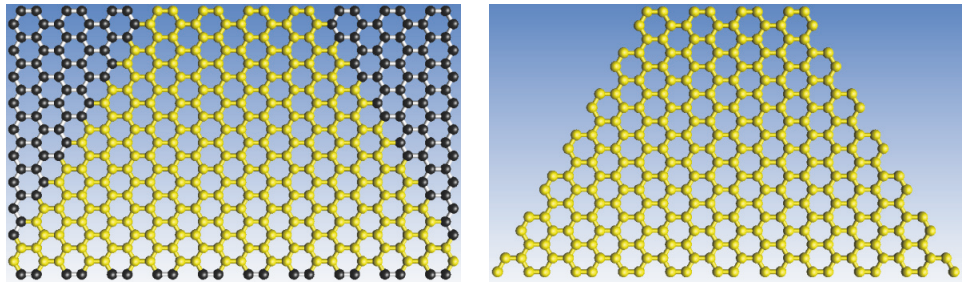
(3) Judge whether they bond or not for each atom and its returned neighbors. If so, an edge is added between them.

Using the proposed algorithm, the space usually is divided by cubes which are 1.5 times longer than the chemical bonds. Each cube approximately contains 2 or 3 atoms, and each atom only compares and match with the atoms contained in its 26 adjacent cubes. Hence each atom compares with almost 100 atoms. A macromolecule containing 10^6 atoms can be reconstructed in 10 seconds.

3.3. Modeling for Common Nanomaterials

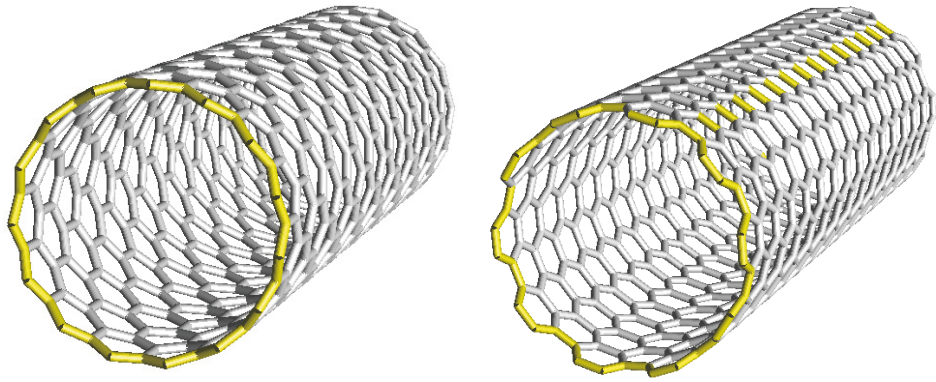
Graphene. Graphene is a regular hexagon. Given an atom's coordinates, all the other atoms' coordinates can be calculated. However the existing software programs cannot control graphene's shape well or lack functions for building personalized shapes. An algorithm is proposed in this paper for modeling graphene parametrically, which provides good interaction for building different shapes of graphene.

Algorithm II. Parametric modeling for graphene.



(a) (b)

Figure 5: Personalized tailoring for graphene.



(a) Armrest shape (b) Chiral shape

Figure 6: Modeling for carbon nanotube.

(1) Generate all the carbon atoms of graphene. Arrange all the atoms according to row and column, and generate coordinates of atoms, respectively, for odd and even rows.

(2) Generate chemical bonds. For each atom, find the left and right adjacent atoms and bond them. Though each row has a fixed number of atoms, it is easy to find the up and down adjacent atoms.

It is easy to generate regular shape of graphene. However modeling different shapes of graphene or defective graphene is difficult. The user can select and tailor local regions interactively to model personalized shapes. How to map user's operation to 3D space? HOOPS [28] provides functions to map windows' coordinates and viewpoints to a geometric structure and return the selected object. It also provides functions for selecting polygon area. The user can tailor shapes according to the demand. Figure 5 shows a tailored parallelogram of graphene created by our system.

Carbon Nanotube. Carbon nanotube can be modeled by bending a planar graphene. The degree of bending can be adjusted by two parameters n and m. Given an atom O, n and m are, respectively, the numbers of the unit vectors through the point O in two directions. Chiral vector $\overrightarrow{C} = n\overrightarrow{a_1} + m\overrightarrow{a_2}$. When $m = n$, then chiral angle $\theta = 30^\circ$, and an armrest shape of carbon nanotube is created, as shown in Figure 6(a); when $m \neq n$, then chiral angle $0^\circ < \theta < 30^\circ$, and a chiral shape of carbon nanotube is created, as shown in Figure 6(b).

MOFs. MOFs (Metal Organic Frameworks) is a new type of porous material composed of organic material and metal material. Though MOFs have a wide variety, varying molecular structure, and uncertain organic combination, modeling MOFs is difficult. Liu and Zhong [29] analyze the gas separation properties of MOFs by computer modeling. But they do not provide the specific modeling method. A template based method is proposed for modeling certain MOFs materials in this paper. A cube template is defined first and then metal ions and organic compounds are matched to the cube according to their coordinates. Metal ion group corresponds to vertices of cube, and organic skeletons correspond to edges of cube, as shown in Figure 7. The algorithm for mapping MOFs to cube template is described below.

Algorithm III. Mapping MOFs to cube template.

(1) Match metal ion group to 8 vertices of cube through 3 symmetries.

(i) Let $x = R$ be the symmetry plane; R is a half of the side length. The new point's coordinates can be calculated by $(x', y', z') = (2R-x, y, z)$. Then two vertices can be gotten after this operation.

(ii) Let $y = R$ be the symmetry plane. Four points can be gotten after this operation.

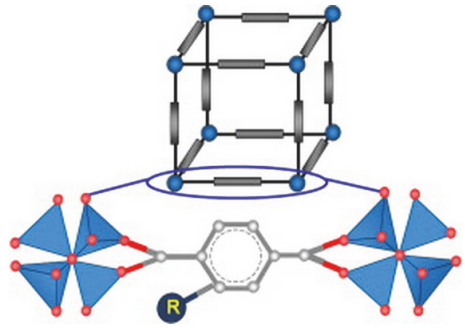
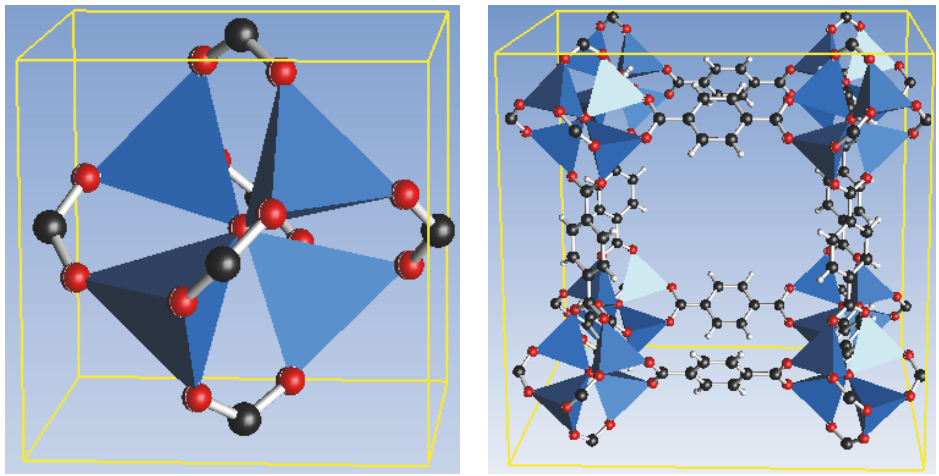


Figure 7: Organic metal frame structure.



(a) (b)

Figure 8: Template based modeling for MOFs.

(iii) Let $z = R$ be the symmetry plane. Eight points can be gotten after this operation.

(2) Match organic compound to 12 edges of cube through symmetry and flip operations.

The MOFs model created by our method is shown in Figure 8. Among them the blue tetrahedron represents metal ions surrounded by nonmetallic materials.

3.4. Advantages of Adjacency List Based Modeling. The adjacency list with OWL representation and octree data structures is applied to model topology and spatial partition of atoms in molecules. The proposed data structure satisfies the needs of macromolecular material modeling. In our experiments, a big model which has atoms more than one hundred thousand can be modeled and visualized in our prototyping system. For example, Figure 9 shows part of a graphene model which has 160,400 atoms and 240,000 chemical bonds. It can be modeled and visualized within 6.4 seconds in our prototyping system. The data structure for nanomaterials defined in this paper has the following advantages:

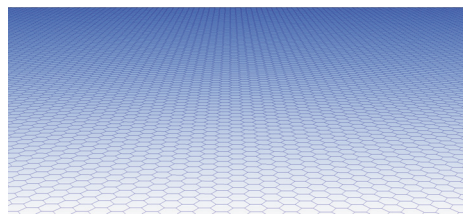


Figure 9: Macromolecular material modeling.

(i) They rapidly search for neighboring atoms and chemical bonds. The traditional data structure only stores the adjacency nodes without edges. Hence the neighboring edges can not be acquired. In contrast, we also store and represent each atom's chemical bonds, that is, the corresponding edges. Hence the adjacent atoms can also be found out quickly by traversing these edges. The time complexity is $O(1)$.

(ii) The adjacency list could be assembled and generated quickly. For a given nanomolecule, generating the related adjacent list is fast and easy. And the time cost is low; the memory overhead increases linearly.

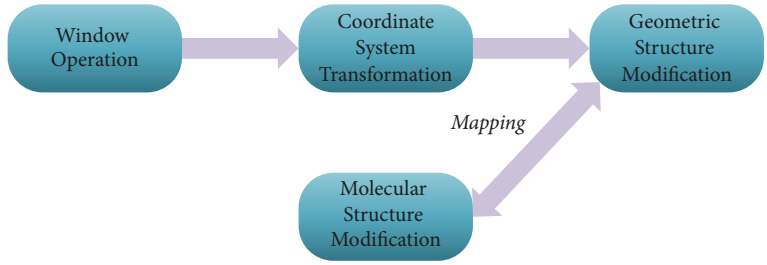


Figure 10: Process of interactive model editing.

Hence the adjacent list is appropriate for building macromolecular model.

(iii) The data structure could be applied for interactive editing, and the response speed is fast. The *map* is used for recording relationships between geometric information and topological information. The corresponding atoms' information could be retrieved when editing geometric information. The time complexity is $\log(n)$.

(iv) They have good extendability. Because the nanomaterials could be modeled uniformly with OWL representation, for new added materials, many undefined properties of atoms such as electric charge and ion track can be plugged incrementally and searched rapidly. The expendability is well.

4. Interactive Model Editing

Many existing molecule modeling systems support editing and modifying the models. At the same time, the physical meaning of the molecules is usually neglected. However, the modified molecule models are usually unstable. For example, if you add or delete an atom, it will affect the other atoms' position. In this paper, we provide interactive operations to modify model's basic structure, update relations of chemical bonds adaptively, and then optimize the model's overall energy. Hence the modified model retains its physical meaning. The process of interactive model editing is shown in Figure 10; window operations are transformed to the world coordinate system; these have been demonstrated by HOOPS. Meanwhile a mapping table is maintained to preserve the correspondence between geometric structure and molecular structure. For example, if an atom is added, then a sphere would be added to the geometric visualization, and an atom is added to the molecular structure. Their correspondence must be added to the mapping table.

4.1. Adaptive Model Editing. An interactive way is provided for editing molecular model adaptively. The effectiveness of topological structure is preserved. When a new atom is added, its positions are calculated. And the new added atom is also matched with its neighboring atoms to form the chemical bonds.

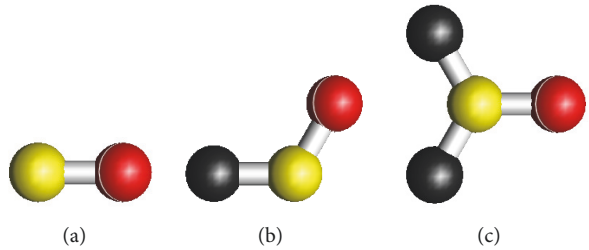
Atom Addition. There are three cases for adding atom: atom has no edge, atom has one edge, and atom has two edges. The positions for the new added atom need to be calculated, respectively. For the first case, an atom with appointed length is added; for the second case, an atom is added to one of the two corresponding edges; for the third case, the coordinates for the new added atom can be calculated directly, as shown in Figure 11. The new added atoms are marked with red.

Of course, the new added atom needs to form the chemical bonds with their neighboring atoms which are less than a certain distance. This process is completed by the octree space partition algorithm. Firstly the topological information is acquired from the geometric sphere of the new added atom; then the atoms are added to its corresponding cube; finally the neighboring atoms are rapidly returned by searching the 26 adjacent cubes. For example, the two chemical bonds (marked with yellow) in Figure 12 are formed automatically.

Atom Deletion. When deleting an atom, the corresponding atom information of the sphere clicked by mouse needs to be removed. And the related chemical bonds need to be removed too. Though mapping between geometric information and topological information is stored in adjacency list, the corresponding atom could be found in $\log(n)$ time. Meanwhile the related chemical bonds could be found by searching the *BondList* and deleted. Hence the correspondence between geometry and topology is retained.

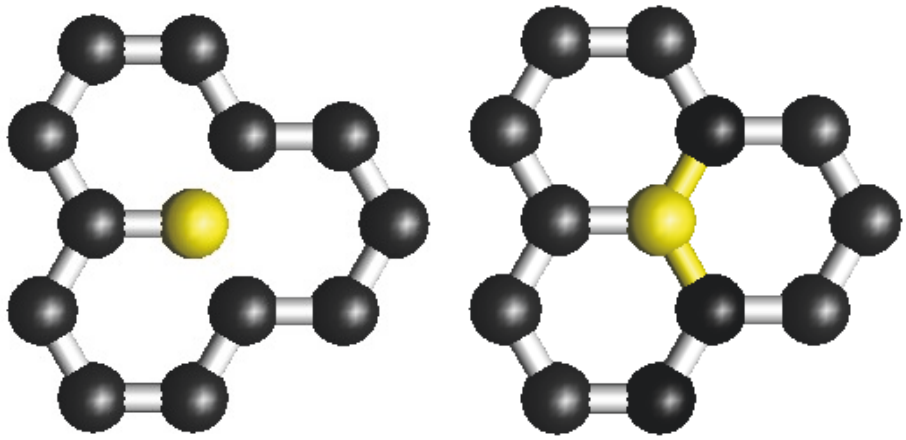
4.2. Model Editing Based on the Physical Meaning. The structure model of nanomaterial often has physical meaning; modifying an atom will affect the other atoms. For example, a molecule of hydrogen H_2O_2 peroxide becomes a water molecule H_2O when deleting an atom O. Its spacing structure will change correspondingly. The system implemented in this paper integrates LAMMPS, optimizes the overall material model through energy minimization, and makes structure tend to be stable. Hence the physical meanings of the nanomaterial model are maintained during the editing process.

The energy minimization is implemented by command *minimize* in LAMMPS. Some parameters such as optimization method min_*style*, largest number of iterations *maxiter*, and convergence precision *etol* can be configured. For example, graphene has relatively stable hexagon. If we delete some atoms during the modeling process, as shown in Figure 13(a), the hexagon is damaged. The remaining atoms will shift because of unbalanced force. The final result generated by our system is shown in Figure 13(b).



(a) (b) (c)

Figure 11: Three cases of atom addition.



(a) (b)

Figure 12: Automatic chemical bond generation.

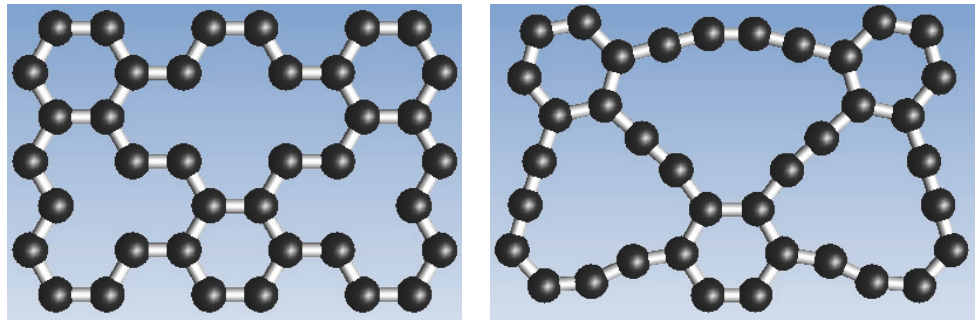
5. Simulation and Calculation for Nanomaterials

How to analyze correlation between structure and macroperformance after building the nanomaterial model? Simulation is an important means of doing this. The key is to establish an efficient calculation system. There exist many simulation strategies for nanomaterials, such as Monte Carlo [30], molecular dynamics [31], and Density Functional Theory (DFT) [32]. The main idea of Monte Carlo is based on the principle of random distribution, which is suitable for macromolecules with millions of atoms, but the accuracy of Monte Carlo is exact. Molecular dynamics predicts the movement of molecules through the force analysis of Newtonian mechanics, which can be used for molecules with hundreds of thousands of atoms, while at the same time holding a relatively high accuracy. DFT mainly analyzes molecular motion according to the relationship among charge, electric field, electron density, and energy, which can be used for molecules with only about one hundred atoms. Based on the tradeoff between accuracy and using range, the existing software programs for nanomaterial simulation often adopt the molecular dynamics method. However, the existing modeling and simulation software programs often cause a problem: modeling and simulation are not integrated. File format conversion is carried out frequently for data exchange between modeling and simulation. The cumbersome conversion process wastes a lot of time and misses some data information. Hence establishing an integrated system for modeling and simulation is vital.

In this section, a RBF (Radial Basis Functions) based method [33, 34] is used for fitting graphene's surface, then Gaussian curvature is calculated according to numerical derivation. After computing the surface energy item of graphene, our system integrates molecular dynamics simulation software LAMMPS for calculation. The simulation results are visualized; and also the results could be edited and simulated iteratively.

5.1. Surface Energy Item Calculation for Graphene. In molecular dynamics, the surface curvature of graphene is related to atom energy [35]. The higher the bending, the greater the energy. At the beginning we notice that graphene can be bent into a wide variety of curved surface but ignore its connection to energy. With the progress of nanotechnology, the graphene plane bending being related to energy has become a consensus. Many scholars suggested that the curvature of the graphene is complementary overlapped with strain capacity, and the energy is related to the Gaussian curvature. Hence the Gaussian curvature calculation is very important for the study of graphene. Because of the interdiscipline, there is rarely research work on calculating surface Gaussian curvature for graphene. Martinez and Jalil [35] calculated graphene's energy for specified parametric surface.

RBF is a simple and powerful tool among common fitting methods. It is easy to realize but meanwhile acquires



(a) (b)

Figure 13: Graphene editing based on physical meaning.

relatively well performance for isotropic data. RBF is applied into multiple domains. de Boer et al. [33] interpolate mesh deformation through RBF. Yoo [34] use RBF and distance field to design heterogeneous porous materials. In this paper, we first fit graphene's surface based on RBF and then compute the Gaussian curvature numerically.

RBF solves a surface function through the known points and makes the known points on the surface function as far as possible. The surface function can be represented as

$$F(X) = \sum_{n=1}^{n} \lambda_n \varphi\left(\left\|X - P_n\right\|\right) + Q(X), \tag{1}$$

where P_n are the n sampling points, λ_n are the coefficients that need to be solved, φ are basis functions, and $Q(X)$ is the function $Q(X) = ax + by + c$. The algorithm for fitting graphene's surface is described below.

Algorithm IV. Fitting graphene's surface by RBF.

(1) Calculate matrix $\varphi_{n,n}$. Input each sampling point to function $\varphi(\|X - P_n\|)$. For example, $\varphi_{12} = \varphi(\|x_1 - x_2\|)$. A $n * n$ matrix will be returned after this step.

(2) Calculate matrices Q_b and Q_b^T. Q_b is a $n * 3$ matrix. It is the value of polynomial $Q(X)$.

(3) Obtain matrix equation

$$\begin{bmatrix} \varphi_{n,n} & Q_b \\ Q_b^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \beta \end{bmatrix} = \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix}. \tag{2}$$

(4) Transform the matrix equation to the following:

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & & \varphi_{1n} & 1 & x_1 & y_1 \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & & \varphi_{2n} & 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ \varphi_{n1} & \varphi_{n2} & \varphi_{n3} & \cdots & \varphi_{nn} & 1 & x_n & y_n \\ 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & x_3 & & x_n & 0 & 0 & 0 \\ y_1 & y_2 & y_3 & & y_n & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3}$$

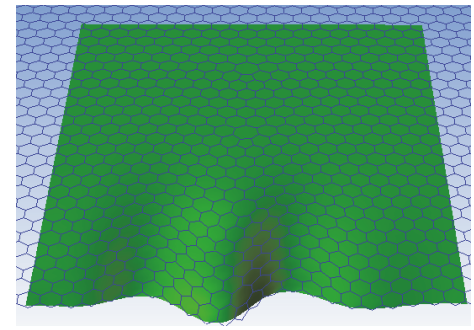


Figure 14: Fitting graphene's surface by RBF.

Gaussian elimination method could be used to solve (3). Figure 14 shows one fitting result. The blue surface is the original graphene, and the green surface is the fitting result. Equation (4) is used for calculating the Gaussian curvature of the fitting surface $F(X)$ [36].

$$K_G = \frac{\nabla F * H^*(F) * \nabla F^T - |\nabla F|^2 \operatorname{Trace}(H(F))}{|\nabla F|^4}. \tag{4}$$

Gaussian curvature is a measurement in surface theory which can reflect surface's geometric properties the most. It describes the degree of bending of surface directly and intuitively. The bigger the curvature, the greater the bending.

5.2. Integration of Molecular Dynamics Simulation. Molecular dynamics simulation is the most commonly used method for nanoscale molecule simulation. It builds equation through Newtonian mechanics. The existing systems usually have some deficiencies. For example, the modeling process and simulation process are separated and scattered in different software kits. The user usually builds models in one software platform and then exports the built models to another simulation platform for calculation; finally the simulation results are imported back to modeling software platform or loaded to other visualization toolkits. Such iteration is time consuming. And some important molecular properties may be lost during the data conversion process. Hence currently research on modeling and simulation for nanomaterials integrally has become a trend in this domain.

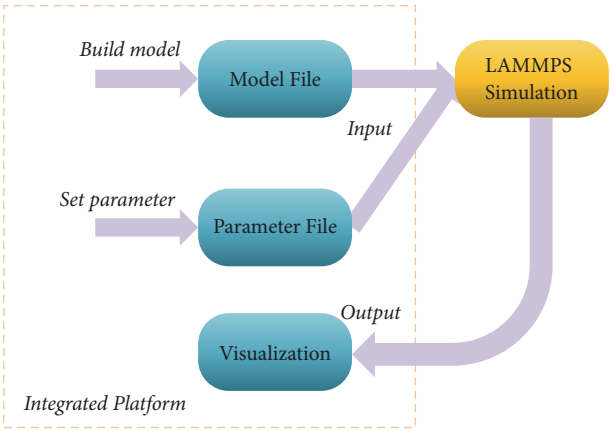


Figure 15: Pipeline for integrated simulation.

Molecular dynamics is composed of two factors: (1) the interaction force between particles decided by the potential energy function and (2) the motion rules between molecules which conform to Newtonian mechanics. Hence we can build the molecular motion model, solve the equation according to Newtonian mechanics theorem, and finally obtain molecular motion trajectory.

The main idea of molecular dynamics simulation is as follows: the stress of atom is the sum of all other atoms' load on it, usually decided by the atom's potential energy. It is related to relative distances between atoms. There are many formulas for computing the atom's potential energy, such as Lennard-Jones potential [37].

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]. \quad (5)$$

Among them σ, ε are constants. Each atom's stress can be calculated by $F = dV(r)/d(r)$. And the sum of stress can be calculated by $F_i = \sum_{j=0}^{n} F_{ij}$. Then the accelerated velocity could be computed and the next position of atom could be predicted. Such a process iterates until atom's energy is minimized.

5.3. Integration of LAMMPS. Our system integrates the molecular dynamics simulation software LAMMPS. It is a mature molecular dynamics simulation software which provides calculation for many properties of nanomolecules and choices for many potential energy functions. However LAMMPS lacks visualization interface and interactive modeling method. The user needs to write script programming language to implement simulation. Meanwhile many times are spent on model format conversion, file import and export, and so on. Hence our prototyping system integrates LAMMPS, sets parameters through graphic user interface, and exchanges information through reading and writing configuration files. This way facilitates the user's operation greatly.

The integrated simulation pipeline is shown in Figure 15. At first nanomaterials are modeled and parameters are set up; then both of them are taken as input to LAMMPS for simulation; finally the results can be output and visualized.

Set LAMMPS Arguments. The main task for implementing molecular dynamics simulation by LAMMPS software is writing LAMMPS's *in* file. The *in* file is vital, though all control parameters, molecular models, and simulation environments are defined here. It includes four main parts:

(i) Initialization: setting up basic information such as energy, temperature using what kind of potential energy, and whether to use multicore parallel computing.

(ii) Atom definition such as file name of the nanomolecule structure and total size of the crystal package.

(iii) Simulation parameters such as simulation parameters, nearest neighbor method, and coefficient of position.

(iv) Operating parameters such as running time and output format.

After writing *in* file, the simulation could be proceeded by running command *LAMMPS − in*. And the corresponding output is returned after completing the computing process, as shown in Figure 16.

Invoke LAMMPS. How to execute files such as molecule model files and configuration files? In this paper, we use multiprocess *CreateProcess* to invoke DOS commands and then wait for commands *WaitForSingleObject* after completing LAMMPS's process. The system is accordance with LAMMPS through process waiting. The correctness of the output files is guaranteed by this way.

6. Visualization and Prototype System

6.1. Prototype System. The prototype system for integrated modeling and simulation are developed by C++ language under VS2010. It is mainly based on HOOPS. Some open source libraries such as OpenCV and PCL are also utilized. The system has four main functions such as structure modeling, model editing, simulation, and visualization, as shown in Figure 17. Structure modeling builds common nanomaterials' molecular model interactively; model editing provides operations such as adaptive bonding and editing based on physical meaning; simulation provides property calculation, surface fitting, curvature calculation, and molecular dynamics simulation; and there are four kinds of visual ways, namely, sphere, bar, sphere-bar, and wireframe. The user can observe the created model from different viewpoints conveniently. Figure 18 is the user interface of the developed prototype system.

MVC (Model-View-Controller) is a common software design pattern. It divides the software into three interconnected parts, so as to separate internal representation of information from the way that information is accepted from or presented to the user. Inspired by this thought, we use MVO (Model-View-Operator) framework to design the prototype system, improving its flexibility and extendibility. Under the MVO framework, *Model* representing molecular structure model is the central component of the framework. It provides definition for molecule model, functions for atom

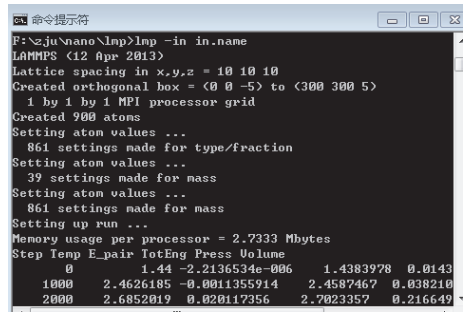


Figure 16: The running process of LAMMPS.

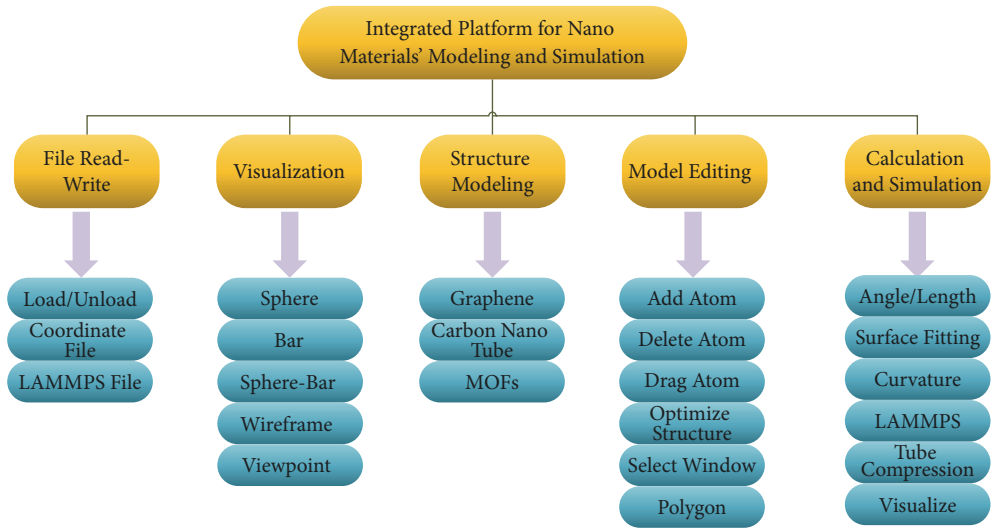


Figure 17: Main functions of the integrated platform.

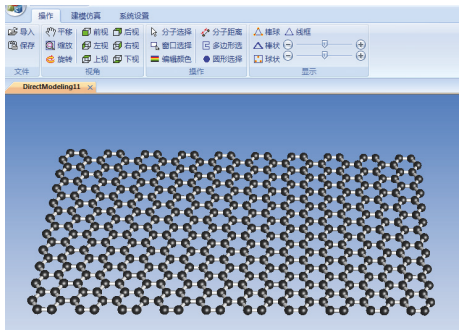


Figure 18: User interface of the prototype system.

addition or deletion, adjacent atom search, adjacent chemical bond search, and so on; *View* generates new output to the user based on changes in the model. Multiple views of the same information are possible, such as sphere, bar, sphere-bar, and wireframe; *Operator* sends commands to change the view's presentation of molecule model (e.g., switching viewpoints), or send commands to the molecule model to update the model's state (e.g., deleting an atom). Our system's framework is composed of four parts including molecule model, geometry visualization, and topology mapping, as shown in Figure 19. Because the correspondence between geometry and topology needs to be maintained for nanomaterials' molecule model, the part of topology mapping is added.

6.2. Experimental Results and Visualization

Geometry Visualization. Our prototype system provides four modes for displaying molecule model from multiple viewpoints, as shown in Figure 20. *Sphere-Bar* is a common molecular display model. It uses sphere of specified radius to represent the atom and bar of specified length to represent the chemical bond. The connections between atoms could be observed intuitively, and sizes and colors of spheres and bars could be changed arbitrarily. *Wireframe* is an important mode for molecular dynamics simulation. The overall shape of molecule can be observed conveniently. And also the rendering time and memory overhead are significantly reduced compared with *Sphere-Bar* mode. The molecule model with a huge number of atoms usually is rendered by *Wireframe* mode. For example, Figure 21 is a graphene which is composed of more than 7000 atoms and 9000 chemical bonds. It can be displayed in real time in our developed prototype system.

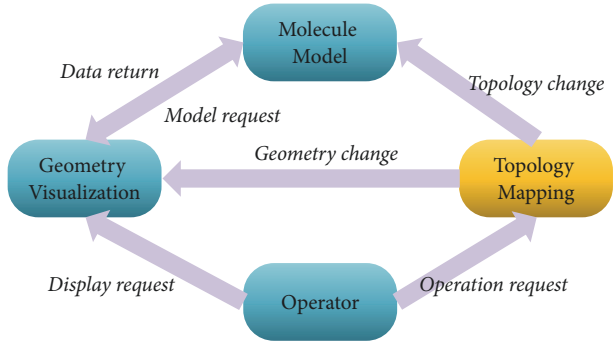
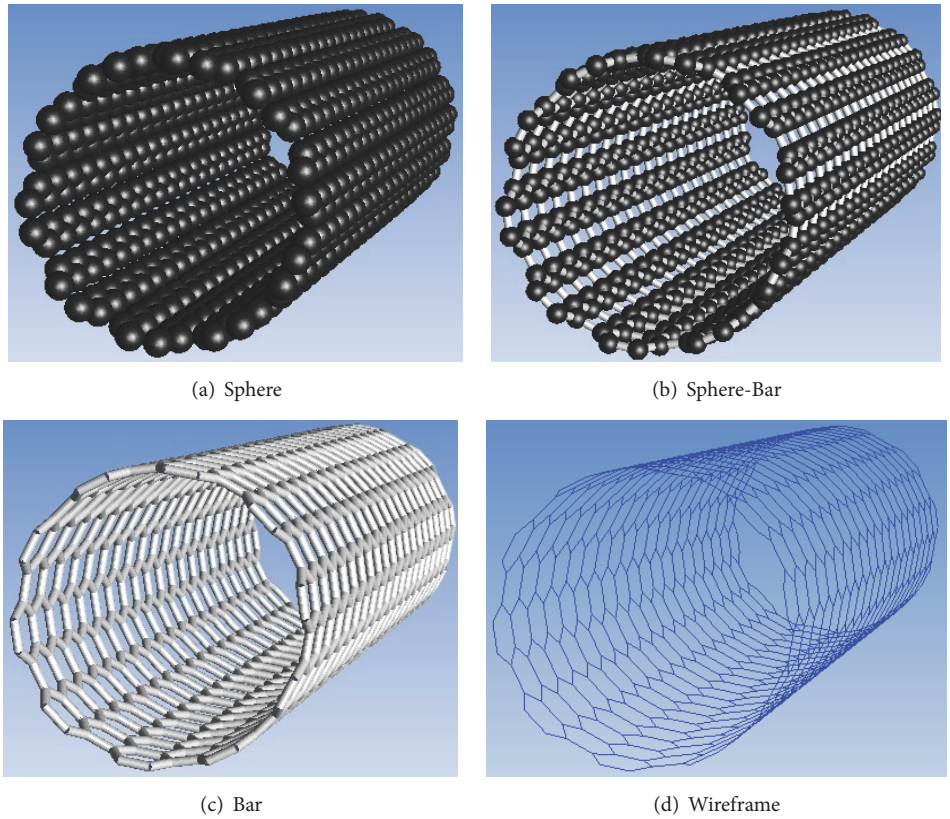


Figure 19: MVO based system framework.



(a) Sphere (b) Sphere-Bar

(c) Bar (d) Wireframe

Figure 20: Multimode visualization.

Modeling for Nanomaterials. Our system provides interactive way to create a variety of common materials. Figure 22(a) is multilayer graphene model. It can be adjusted by parameters (n, m, c), where n is length, m is width, and c is number of layers. Figure 22(b) is an irregular graphene model created by interactive polygon selection. And carbon nanotube with different physical meaning can be modeled by setting parameters (n, m, l), where n and m are chiral parameters and l is length of tube. Figure 23(a) is a tube with $(n, m, l) = (30, 10, 10)$. Figure 23(b) is a multilayer tube. In addition, based on the cube template, the user can compose organic material and metal material to form different MOFs. Figure 24 shows some results of MOFs modeling.

Model Editing. At first, if atoms are added or deleted, our system can generate atoms' positions adaptively. Meanwhile the chemical bonds are added or deleted correspondingly. Figure 25(a) shows the result after deleting two atoms; Figure 25(b) shows the result after adding two atoms. The added atoms are marked with yellow, and the three pink bars are chemical bonds added automatically by our system. In addition, editing for nanomolecule model is closely related to material's physical meaning. The arrangement for other atoms may change after editing one atom. Hence structure optimization is utilized for maintaining the model's physical meaning. For example, Figure 26(a) shows an initial state of a graphene after deleting some atoms; Figure 26(b) is the

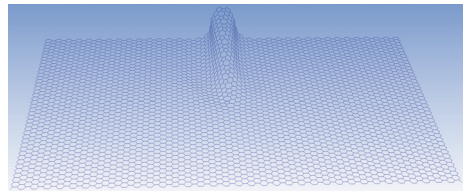
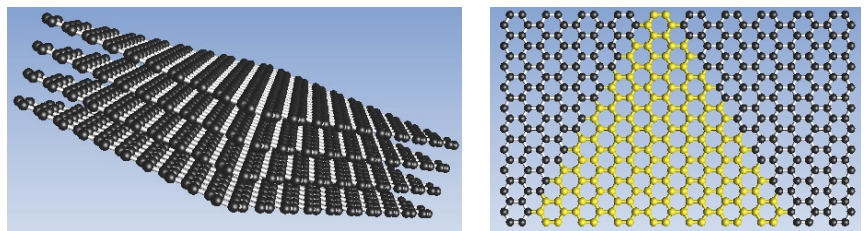
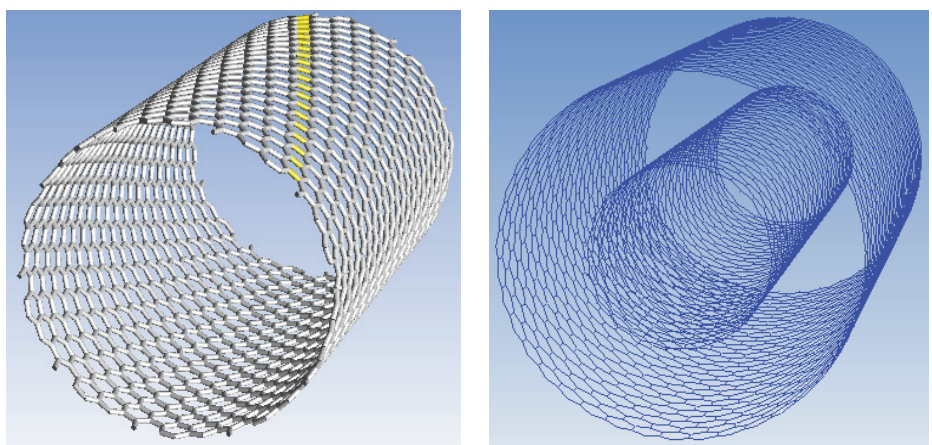


Figure 21: Wireframe display for macromolecular nanomaterials.



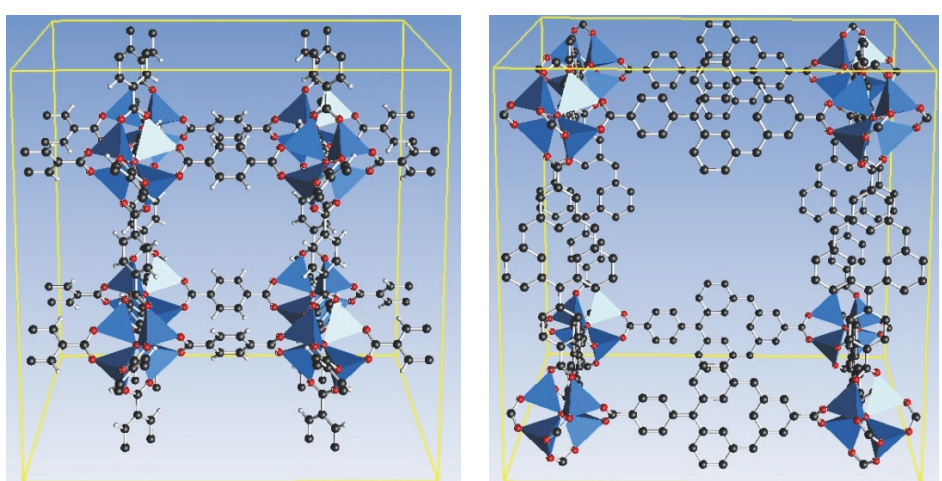
(a) Multilayer (b) Irregular

Figure 22: Modeling for graphene.



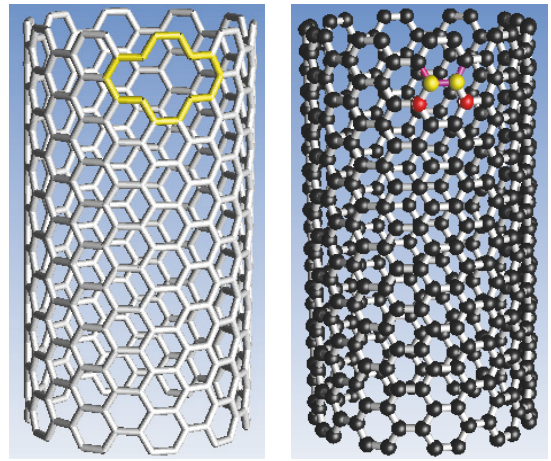
(a) (b)

Figure 23: Modeling for carbon nanotube.



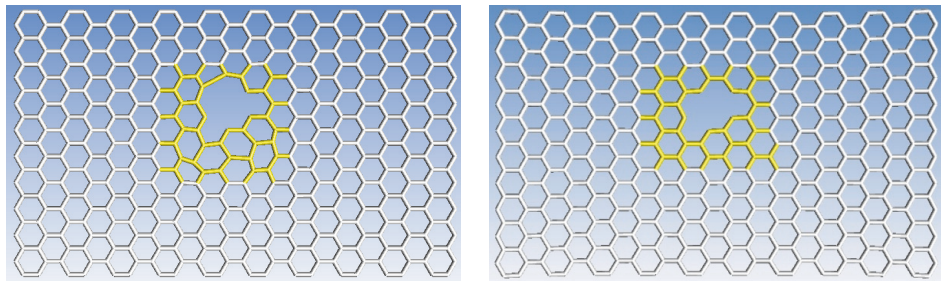
(a) (b)

Figure 24: Modeling for MOFs.



(a) (b)

Figure 25: Adaptive editing for nanotube.



(a) (b)

Figure 26: Physical meaning based editing for graphene.

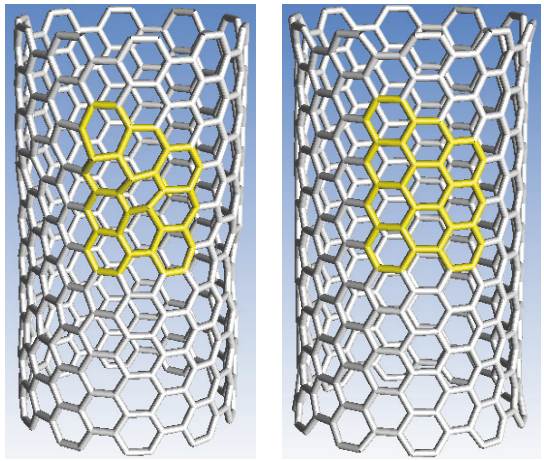
stable model after optimization. Figure 27(a) is an initial state of a nanotube model. Because some atoms move, the distortion occurs; Figure 27(b) is the stable state after optimization.

Calculation and Simulation. The measurement of bond angle and bond length is very important for studying the structure of nanomaterials. At first, our system provides interactive operations for measuring the bond angle and bond length in 2D/3D spaces. As shown in Figure 28, the left is measuring bond length and bond angle of graphene in 2D space; and the right is measuring bond length and bond angle of fullerene molecule in 3D space.

Secondly, functions are provided for calculating nanomaterials' energy item. The prototype system reads the coordinate information of graphene and then fits the surface based on RBF. For example, Figure 29 shows a result of fitting graphene. It is visualized using quadrilateral meshes. Figure 30 is the result of calculating graphene's curvature. The results are output by invoking MATLAB.

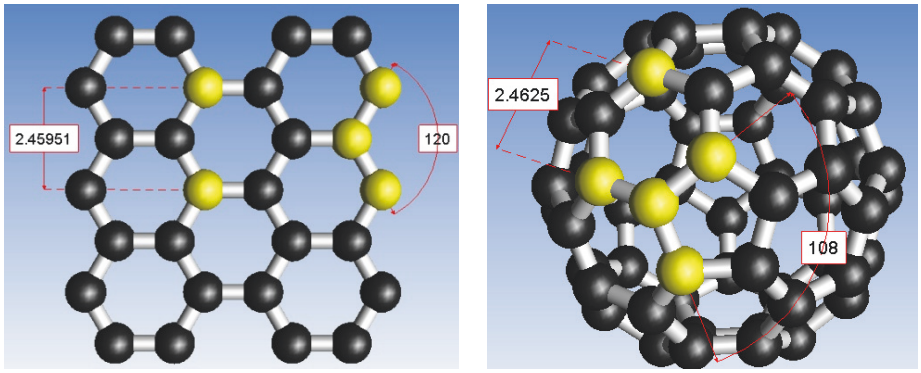
Thirdly, our system integrates the modeling and simulation processes together. Carbon nanotube compression is a common nanomaterial simulation example. Through analyzing the stress distribution of nanotubes, a solid structure could be designed. Figure 31 shows the simulation results of compressing nanotube. Different forces are accurately loaded through lever operations. Though LAMMPS is integrated into our prototype system, the calculating results can be returned in time without tedious file conversion. Moreover, our system supports reediting the simulated model.

6.3. Comparison with the Existing Approaches. We compare our approach with some existing nanomaterial modeling and simulation approaches in Table 1. The second column shows the application domains of these typical existing systems. The third column shows the modeling abilities, the fourth column shows the simulation and calculation capabilities, and the fifth column shows the visualization functions provided by these systems, respectively. The first row is our approach, and the following rows are four typical commercial solutions. All the approaches are capable of completing one or more key aspects in the domain of modeling and simulation for nanomaterials. However, all of these approaches are not powerful enough to implement all the three key aspects including modeling, simulation, and visualization seamlessly in an integrated platform. In contrast, our approach provides an integrated modeling, simulation, and visualization platform for nanomaterials. The designer can use our system to design new nanomaterials conveniently without cumbersome and repetitive file conversion. And the system also



(a) (b)

Figure 27: Physical meaning based editing for nanotube.



(a) (b)

Figure 28: Measurement of bond angle and bond length.

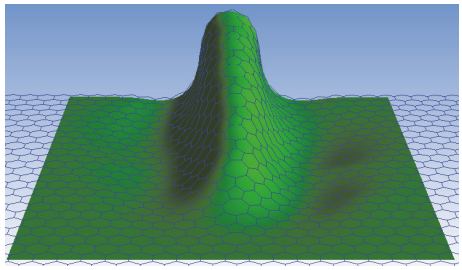


Figure 29: Surface fitting for graphene.

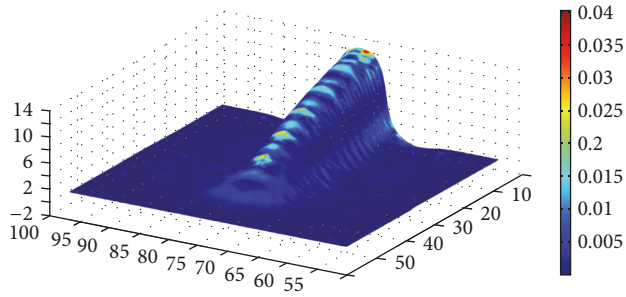


Figure 30: Curvature calculation for graphene.

provides interactive model editing function which maintains the physical meaning.

7. Conclusion and Future Work

In conclusion, this paper builds an integrated modeling and simulation platform for nanomaterials. Based on this platform, the user could establish correlation between materials' structure and macrophysical properties, shortening the development cycle of new nanomaterials. Its novelty and contributions lie the following aspects.

Modeling. Firstly, an adjacency list based data structure which is suitable for macromolecular material modeling is built; secondly, an octree space partition algorithm is adopted to reconstruct macromolecule rapidly; thirdly, interactive ways are provided for modeling a variety of nanomaterials such as

Table 1: Comparison with the existing systems.

Approaches	Domains	Modeling	Simulation	Visualization	Interactive editing
Ours	Common nanomaterials	Modeling based on adjacency list, support rapid macromolecule reconstruction	Calculation for surface energy items, integration of LAMMPS	Multimode visualization based on HOOPS	Adaptive model editing, maintaining physical meaning
Overvelde et al. [14]	Porous materials	Buckling in 2D periodic, soft and porous structures	Stress strain analysis	Simple visualization	—
Materials Studio [15]	Common nanomaterials	Structure modeling	Simple property calculation	Incompatibility problem	—
LAMMPS [16]	Molecular dynamics	Modeling macromolecule	Powerful computing capability	—	—
RasMol [17]	Nanomolecular model	Modeling macromolecule	—	Powerful and convenient display interface	—

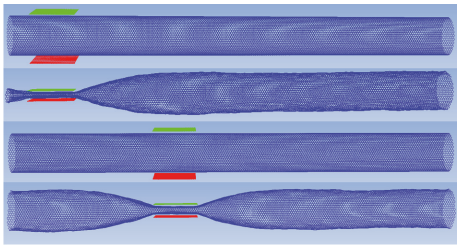


Figure 31: Simulation for carbon nanotube compression.

graphene, carbon nanotube, and MOFs, which is supposed to be convenient and flexible.

Model Editing. An adaptive way is provided for automatically adding or deleting the corresponding chemical bonds after changing the atoms. Moreover, the optimization method is carried out to maintain the physical meaning of nanomaterials during the editing process.

Calculation and Simulation. The energy item of nanomaterials (e.g., graphene) can be computed by RBF based surface fitting and numerical calculation. And by integrating molecular dynamics software LAMMPS, nanomaterials could be modified and simulated iteratively, without tedious and time consuming file format conversion.

User Interaction and Visualization. Based on MVO pattern, our developing prototype system has a user-friendly interface, convenient for operation. Meanwhile, several modes are provided for visualizing the modeling and simulation results from multiple viewpoints.

Due to involving multiple disciplines, how to model and simulate nanomaterials under an integrated manner is a challenging research topic. Though some beneficial exploration has been done in this paper, there is still a lot of work that needs to be completed. In the future, more nanomaterials such as DNA and protein will be modeled in our system. Experiments on these materials will proceed to further verify the feasibility and effectiveness of the proposed integrated modeling and simulation approach. In addition, when editing models based on physical meaning, the current prototype relies on LAMMPS software. More molecular dynamics algorithms and artificial intelligence algorithms [38, 39] could be implemented to optimize molecule's structure. Furthermore, more material properties such as electron density and electron orbit could be considered during the calculation and simulation processes.

Conflicts of Interest

There are no conflicts of interest related to this paper. The authors do not have financial and personal relationships with other people or organizations that could inappropriately influence (bias) their work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 61502129, 61602140, 61602139, and 61772164), the Zhejiang Provincial Natural Science Foundation of China (no. LQ16F020004), the Open Project Program of State Key Lab of CAD&CG, Zhejiang University (nos. A1803 and A1817), and the Science and Technology Program of Zhejiang Province (no. 2017C33049).

References

[1] S. Grudimn and S. Redon, "Practical modeling of molecular systems with symmetries," *Journal of Computational Chemistry*, vol. 31, no. 9, pp. 1799–1814, 2010.

[2] T. Schlick, *Molecular Modeling and Simulation*, Springer, Berlin, Germany, 2002.

[3] A. V. Tran and Y. Wang, "Reliable molecular dynamics: uncertainty quantification using interval analysis in molecular dynamics simulation," *Computational Materials Science*, vol. 127, pp. 141–160, 2017.

[4] S. Torquato and H. Haslach, "Random heterogeneous materials: microstructure and macroscopic properties," *Applied Mechanics Reviews*, vol. 55, no. 4, p. B62, 2002.

[5] X. Zhao, C. Zhang, B. Yang, and P. Li, "Adaptive knot placement using a GMM-based continuous optimization algorithm in

B-spline curve approximation," *Computer-Aided Design*, vol. 43, no. 6, pp. 598–604, 2011.

[6] X. Zhao, Y. Yin, B. Yang, B. Zhu, and X. Tian, "Level set and geodesic active contours based measurement of material removal between serial sections," *Computational Materials Science*, vol. 39, no. 4, pp. 857–861, 2007.

[7] Y. Zhou, F. He, and Y. Qiu, "Optimization of parallel iterated local search algorithms on graphics processing unit," *The Journal of Supercomputing*, vol. 72, no. 6, pp. 2394–2416, 2016.

[8] Y. Wang and I. Horváth, "Computer-aided multi-scale materials and product design," *Computer-Aided Design*, vol. 45, no. 1, pp. 1–3, 2013.

[9] Y. Wang, "CAD at the nano scale," *Mechanical Engineering*, vol. 136, no. 8, pp. 32–37, 2014.

[10] Y. Wang, "Stochastic dynamics simulation with generalized interval probability," *International Journal of Computer Mathematics*, vol. 92, no. 3, pp. 623–642, 2015.

[11] Y. Wu, F. He, D. Zhang, and X. Li, "Service-oriented feature-based data exchange for cloud-based design and manufacturing," *IEEE Transactions on Services Computing*, vol. 1, no. 1, 2015.

[12] D. J. Zhang, F. Z. He, S. H. Han, and X. X. Li, "Quantitative optimization of interoperability during feature-based data exchange," *Integrated Computer-Aided Engineering*, vol. 23, no. 1, pp. 31–50, 2015.

[13] L. Li and J. Liu, "An efficient and flexible web services-based multidisciplinary design optimisation framework for complex engineering systems," *Enterprise Information Systems*, vol. 6, no. 3, pp. 1–27, 2012.

[14] J. T. B. Overvelde, S. Shan, and K. Bertoldi, "Compaction through buckling in 2D periodic, soft and porous structures: effect of pore shape," *Advanced Materials*, vol. 24, no. 17, pp. 2337–2342, 2012.

[15] S. J. Clark, M. D. Segall, C. J. Pickard et al., "Materials Studio CASTEP," version 5.0, Accelrys, San Diego, California, Calif, USA, 2002.

[16] SandiaLab, LAMMPS Molecular Dynamics Simulator, 2017, http://lammps.sandia.gov/.

[17] S. Artemova, S. Grudinin, and S. Redon, "Fast construction of assembly trees for molecular graphs," *Journal of Computational Chemistry*, vol. 32, no. 8, pp. 1589–1598, 2011.

[18] R. Lou, J.-P. Pernot, A. Mikchevitch, and P. Véron, "Merging enriched Finite Element triangle meshes for fast prototyping of alternate solutions in the context of industrial maintenance," *Computer-Aided Design*, vol. 42, no. 8, pp. 670–681, 2010.

[19] M. Muja, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the 4th International Conference on Computer Vision Theory and Applications (VISAPP '09)*, vol. 1, pp. 331–340, 2009.

[20] B. C. Ooi, K. J. Mcdonell, and R. Sacks-Davis, "Spatial kd-tree: an indexing mechanism for spatial databases," in *Proceedings of the International Computer Software and Applications Conference*, 1987.

[21] J. E. Stone, D. J. Hardy, I. S. Ufimtsev, and K. Schulten, "GPU-accelerated molecular modeling coming of age," *Journal of Molecular Graphics and Modelling*, vol. 29, no. 2, pp. 116–125, 2010.

[22] W. Humphrey, A. Dalke, and K. Schulten, "VMD: visual molecular dynamics," *Journal of Molecular Graphics*, vol. 14, no. 1, pp. 33–38, 1996.

[23] G. Deléage, C. Combet, C. Blanchet, and C. Geourjon, "ANTHEPROT: an integrated protein sequence analysis software with client/server capabilities," *Computers in Biology and Medicine*, vol. 31, no. 4, pp. 259–267, 2001.

[24] M. Horridge and S. Bechhofer, "The OWL API: a Java API for OWL ontologies," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no. 1, pp. 11–21, 2011.

[25] B. C. Kim, Y. Jeon, S. Park, H. Teijgeler, D. Leal, and D. Mun, "Toward standardized exchange of plant 3D CAD models using ISO 15926," *Computer-Aided Design*, vol. 83, pp. 80–95, 2017.

[26] F. Qin, S. Gao, X. Yang, M. Li, and J. Bai, "An ontology-based semantic retrieval approach for heterogeneous 3D CAD models," *Advanced Engineering Informatics*, vol. 30, no. 4, pp. 751–768, 2016.

[27] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings of the Proceeding of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 1–4, 2011.

[28] TechSoft3D, HOOPS 3D Framework, 2017, http://docs.techsoft3d.com/visualize/3df/latest/.

[29] D. Liu and C. Zhong, "Understanding gas separation in metal-organic frameworks using computer modeling," *Journal of Materials Chemistry*, vol. 20, no. 46, pp. 10308–10318, 2010.

[30] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, CRC press, Boca Raton, Fla, USA, 1995.

[31] S. Nosé, "A molecular dynamics method for simulations in the canonical ensemble," *Molecular Physics*, vol. 52, no. 2, pp. 255–268, 1984.

[32] G. Zhang and C. B. Musgrave, "Comparison of DFT methods for molecular orbital eigenvalue calculations," *The Journal of Physical Chemistry A*, vol. 111, no. 8, pp. 1554–1561, 2007.

[33] A. de Boer, M. S. van der Schoot, and H. Bijl, "Mesh deformation based on radial basis function interpolation," *Computers & Structures*, vol. 85, no. 11-14, pp. 784–795, 2007.

[34] D. Yoo, "Heterogeneous minimal surface porous scaffold design using the distance field and radial basis functions," *Medical Engineering & Physics*, vol. 34, no. 5, pp. 625–639, 2012.

[35] J. C. Martinez and M. B. A. Jalil, "Gaussian curvature energy of graphene sheets," *Physics Letters A*, vol. 375, no. 24, pp. 2437–2440, 2011.

[36] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin, "A comparison of Gaussian and mean curvatures estimation methods on triangular meshes," in *Proceedings of the IEEE International Conference on Robotics and Automation ICRA '03*, vol. 1, pp. 1021–1026, 2003.

[37] Wikipedia, Lennard-Jones potential, 2017, http://en.wikipedia.org/wiki/Lennard-Jones-potential/.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[39] Y. Peng and B.-L. Lu, "Discriminative extreme learning machine with supervised sparsity preserving for image classification," *Neurocomputing*, vol. 261, pp. 242–252, 2017.