



Article

Design of the Algorithm for Packaging of Water Molecules in a Fixed Volume

Ekaterina M. Lorits ^{1,†}, Elena A. Gubar ^{1,†}  and Alexander S. Novikov ^{2,3,*,†} 

¹ Faculty of Applied Mathematics and Control Processes, Saint Petersburg State University, Universitetskii Pr., 35, Petergof, 198504 St. Petersburg, Russia

² Institute of Chemistry, Saint Petersburg State University, Universitetskii Pr., 26, Petergof, 198504 St. Petersburg, Russia

³ Infochemistry Scientific Center, ITMO University, Lomonosova St., 9, 191002 St. Petersburg, Russia

* Correspondence: a.s.novikov@spbu.ru or novikov@itmo.ru

† These authors contributed equally to this work.

Abstract: The question about the closest symmetrical packing of chemical substance species (molecules, ions, polymer chains, nanoparticles, etc.) is a subproblem of predicting the structure of matter, in particular, the structure of a crystal, information about which makes it possible to predict almost all of its properties. The design of mathematical models for the closest symmetrical packing is an important and a challenging task for the practical application of optimization theory in theoretical chemistry. Here, we report about the development of the algorithm for water molecules' symmetrical packing in a closed space of rectangular parallelepiped.

Keywords: closest packing; optimization theory; water



Citation: Lorits, E.M.; Gubar, E.A.; Novikov, A.S. Design of the Algorithm for Packaging of Water Molecules in a Fixed Volume. *Symmetry* **2022**, *14*, 2453. <https://doi.org/10.3390/sym14112453>

Academic Editors: Enrico Bodo and Haruo Hosoya

Received: 9 October 2022

Accepted: 16 November 2022

Published: 19 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The question about the closest symmetrical packing of chemical substance species (molecules, ions, polymer chains, nanoparticles, etc.) is a subproblem of predicting the structure of matter, in particular, the structure of a crystal, information about which makes it possible to predict almost all of its properties. The design of mathematical models for the closest symmetrical packing is an important and a challenging task for the practical application of optimization theory in theoretical chemistry. In this paper, we report the development of the algorithm for water molecules' symmetrical packing in a closed space of the rectangular parallelepiped. In our model, water molecules were represented as regular tetrahedrons, in which two vertices correspond to hydrogen atoms in the molecule (vertices of type A), and the other two correspond to electron pairs (vertices of type B). The connections between them are shown in Figure 1. It is also important to note that due to the chemical properties of water, bonds between molecules in our model can only be formed between hydrogen and an electron pair, i.e., only bonds of the form A~B or B~A will be considered. Thus, it is necessary that the algorithm considers the types of vertices when attaching molecules and does not allow connections of the form A~A and B~B. Note also that this type of calculation has been performed non-quantum mechanically, and no specific quantum effects (e.g., dealing with electron correlation) were taken into account in our model.

The relevance of the topic is supported by numerous papers on structure prediction [1]. In [2] K. K. Fisher studied a way to search for crystal structures using machine learning and modern quantum mechanical methods. N. Braun and G. Huttner developed an algorithm that shows correlations between molecular shape and the type of packing [3]. In [4] N. W. Thomas suggested an analytical approach to characterize organic molecular crystal structures where close packing is an important factor.

The research team from the State University of New York at Buffalo presented the evolutionary algorithm XtalOpt and its refinements [5]. A. R. Oganov and C. W. Glass

developed a reliable methodology for predicting crystal structures based on evolution algorithms and have demonstrated the efficiency of this method for systems with up to 30–40 atoms per cell [6]. Then, A. O. Lyakhov, A. R. Oganov, H. T. Stokes, and C. Zhu Q. further refined the previous algorithm and achieved a reliable representation for systems of 200 atoms per cell [7]. It is worth noting that the algorithm presented in this work is capable of packing a larger number of atoms: up to 3000 in a rectangular parallelepiped.

E. V. Peresypkina and V. A. Blatov determined the crystal structures of 33,575 monosystem organic compounds within the first three coordination spheres, and demonstrated that generally molecular packings are constructed according to one of the close packings [8]. In [9], A. Gavezzotti and L. Lo Presti introduced a new view of crystal packing that results from joining static structural information from X-ray analysis with dynamic upgrades. The main advantage of this algorithm is that it allows us to quickly and efficiently estimate the number of molecules in a limited volume represented as a parallelogram. In the presented algorithm, a mathematical approximation of the water molecule is made, which makes it easy to obtain the structure of the arrangement of molecules in the parallelepiped. This approximation simplifies the calculations and reduces the running time of the algorithm, which further leads to the possibility of scaling the problem. At the same time, the possibility of adequate representation of the chemical structure of the substance is kept [10].

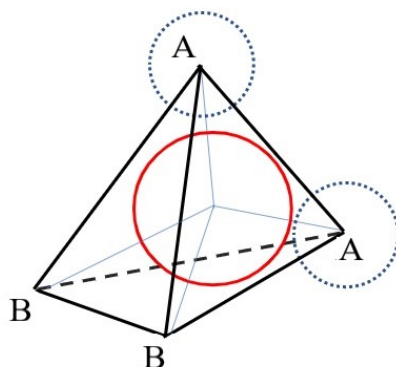


Figure 1. Representation of water molecule. Dotted circles refer to hydrogen atoms, while continuous one depicts oxygen atom.

2. Algorithm

The designed algorithm has a practical value because it allows one to determine the number of water molecules, which can be placed in a rectangular parallelepiped with specified values of length, width and height. This algorithm can be easily used for the creation of initial solvent massive for molecular dynamics' simulations and theoretical studies of the closest packing of molecules. Note that, in the future, we plan to adapt and extend this approach to both the sphere and the ellipsoid, which could be very relevant for fundamental colloid chemistry (problem of correct description of water drops). To the best of our knowledge, to date there are no MATLAB implementations for such algorithms in the literature. Hence, the constructed algorithm can be used as a new tool to estimate the densest packing in specified area. The algorithm consists of the following steps:

Definition 1. *The densest packing in crystallography is the shape of the arrangement of the atoms in the crystal lattice, which is characterized by the largest number of atoms per unit volume of the crystal.*

1. Firstly, it is necessary to specify a volume that is available for completion.
2. Then, we form a matrix that will contain information about the coordinates of the vertices of all added molecules.
3. In accordance with the representation of the water molecule, we need to determine the coordinates of the vertices of the tetrahedron for the first molecule.

4. The coordinates of the nodes of the molecule are recorded in the matrix formed in step 2.
5. Afterwards, it is reasonable to apply the image of the tetrahedron corresponding to the added molecule to the graph.
6. Two variables are defined that store the position of the last added tetrahedron. One of them represents the coordinates of the tetrahedron vertex with the largest abscissa and another one is the number of vertices with the largest abscissa (1 or 2). It is essential to update these variables after each new molecule is added.
7. Similarly, we define two variables that store the position of the first tetrahedron in a row. It is important to update them after moving to the new line.
8. The cycle of molecular attachment is carried out. Firstly, it is necessary to check the ability to include a new molecule in one row with the previous one. If it is possible, we add the molecule using the function that defines vertex coordinates. Steps 4–6 are repeated. Otherwise, the molecule enters the next row. Steps 4–7 are repeated. If in this instance it exceeds the allowable volume as well, the algorithm has to attach the molecule in the first row of a new layer. In this case, steps 4–7 are repeated.

The cycle on step 8 of the algorithm uses a special procedure. It rotates the adding molecule to be connected so as to create a hydrogen–electron pair connection. This procedure determines the coordinates of the vertices of the tetrahedron corresponding to the added molecule, depending on the location of the previous molecule (subject to the types A~B, B~A communication requirements). Furthermore, this procedure takes into account the distance between molecules on the x, y, and z axes, which is set by the user, thus avoiding overlapping and adjusting the distance between molecules depending on the problem.

3. Procedure in MATLAB Language

We implement the algorithm described above in MATLAB. The program calculates the number of molecules that could be stored in a given volume. Furthermore, it forms a table with the coordinates of the vertices of the tetrahedra in the Cartesian coordinate system centred at one of the vertices of the rectangular parallelepiped. This table can be used to interface the results with a visualization program, which required the coordinates of the hydrogen atoms to build atom-based system. The information about the vertices type (having hydrogen or not) can be easily added to this table, because the table stores information about tetrahedron vertices in an ordered way: of every four vertices of tetrahedron, the first two contain hydrogen and the other two do not. Moreover, the program has the ability to determine the interval between the molecules at any of the coordinates (by parameters h_x, h_y, h_z).

The MATLAB procedure is presented below. Firstly, we define the supporting functions.

1. Addition a tetrahedron to a graph

```
function [] = DrawTriangle(new_a,new_b,new_c,new_d)

plot3(new_a(1),new_a(2),new_a(3),'*b','MarkerSize',8);
plot3(new_b(1),new_b(2),new_b(3),'*b','MarkerSize',8);

line([new_a(1),new_b(1)], [new_a(2),new_b(2)], [new_a(3),new_b(3)]);
line([new_c(1),new_b(1)], [new_c(2),new_b(2)], [new_c(3),new_b(3)]);
line([new_a(1),new_c(1)], [new_a(2),new_c(2)], [new_a(3),new_c(3)]);
line([new_a(1),new_d(1)], [new_a(2),new_d(2)], [new_a(3),new_d(3)]);
line([new_b(1),new_d(1)], [new_b(2),new_d(2)], [new_b(3),new_d(3)]);
line([new_c(1),new_d(1)], [new_c(2),new_d(2)], [new_c(3),new_d(3)]);

end
```

2. Determination of coordinates of the new tetrahedron along with the current vertex type and vertex coordinates with the largest abscissa

```

function [a,b,c,d,rightest,type] = NewPutTriangle(type, rightest,
step_x, step_y, step_z,h_x)
if type == 1
b = rightest;
b(1) = b(1) + h_x;
c = [b(1) + 3 * step_x, b(2) + step_y, b(3)];
d = [c(1), b(2)- step_y, b(3)];
a = [b(1) + 2 * step_x, b(2), b(3) + step_z];
rightest = d;
type = 2;
else
a = rightest;
a(1) = a(1) + h_x;
b = [a(1), a(2) + 2 * step_y, a(3)];
c = [a(1) + 3 * step_x, a(2) + step_y, a(3)];
d = [a(1) + step_x, a(2) + step_y, a(3) + step_z];
rightest = c;
type = 1;
end
DrawTriangle(a,b,c,d);
end

```

The body of the main program:

```

tic;
grid on; hold on; axis equal;

alph = 104.25 * pi /180; % angle in radians between two hydrogen atoms
HO = 95.84; % distance between oxygen and hydrogen
AB = sqrt(2 * HO^2 - 2 * HO^2 * cos(alph)); % cosine theorem
step_y = AB/2; % 1/2 the length of the side of the tetrahedron
step_x = AB * sqrt(3)/6; % 1/3 of the height of the triangle
step_z = AB * sqrt(6)/3; % height of the tetrahedron

stop_x = 600; % space where molecules can be
stop_y = 500;
stop_z = 300;

v = AB^3 * sqrt(2)/12; % volume of the tetrahedron
space = stop_x * stop_y * stop_z;
N = ceil(space/v);

% remember coordinates of tetrahedron points A; B; C; D respectively
in matrix where A and B has hydrogen and C and D not
% put that matrix in one big matrix
a = [0, 0, 0];
b = [0, 2 * step_y, 0];
c = [3 * step_x, step_y, 0];
d = [step_x, step_y, step_z];
M = zeros(4 * N, 3);
M(1,:) = a;
M(2,:) = b;
M(3,:) = c;
M(4,:) = d;
DrawTriangle(a,b,c,d);

```

```
type = 1;           % refer to number of dots with max x
rightest = c;      % dot with max x and min y if there are two of them
type1str = 1;     % to remember type of first molecule in the last string
r1s = c;          % to remember dot from first molecule

stop = 0;
h_x = 10;
h_y = 45;         % distance which we left between lines !=0
h_z = 30;         % distance which we left between layers
k = 1;           % to count molecules

while (stop == 0)
if (rightest(1)+ 3 * step_x + h_x) <= stop_x
k = k + 1;
[M(4 * k - 3,:), M(4 * k - 2,:), M(4 * k - 1,:), M(4 * k,:), rightest,
type] = NewPutTriangle(type,rightest, step_x, step_y,
step_z,h_x);

elseif ((type == 1)&&((h_y + 3 * step_y + rightest(2)) <= stop_y))||
((type==2)&&((h_y + 4 * step_y +rightest(2))<= stop_y))
k = k + 1;
if type1str == 1
b2 = [r1s(1) - 3 * step_x, r1s(2) + h_y + step_y, r1s(3)];
c2 = [b2(1) + 3 * step_x, b2(2) + step_y, b2(3)];
d2 = [c2(1), b2(2) - step_y, b2(3)];
a2 = [b2(1) + 2 * step_x, b2(2), b2(3) + step_z];
rightest = d2;
type = 2;
else
a2 = [r1s(1)- 3 * step_x, r1s(2) + h_y + step_y, r1s(3)];
b2 = [a2(1), a2(2) + 2 * step_y, a2(3)];
c2 = [a2(1) + 3 * step_x, a2(2) + step_y, a2(3)];
d2 = [a2(1) + step_x, a2(2) + step_y, a2(3) + step_z];
rightest = c2;
type = 1;
end
type1str = type;
r1s = rightest;
M(4 * k - 3,:) = a2;
M(4 * k - 2,:) = b2;
M(4 * k - 1,:) = c2;
M(4 * k,:) = d2;
DrawTriangle(a2,b2,c2,d2);

elseif (rightest(3) + h_z + 2 * step_z) <= stop_z
k = k + 1;
a3 = [0, 0, rightest(3) + step_z + h_z];
b3 = [a3(1), a3(2) + 2 * step_y, a3(3)];
c3 = [a3(1) + 3 * step_x, a3(2) + step_y, a3(3)];
d3 = [a3(1) + step_x, a3(2) + step_y, a3(3) + step_z];

rightest = c3;
type = 1;
```

```

type1str = type;
r1s = rightest;

M(4 * k - 3,:) = a3;
M(4 * k - 2,:) = b3;
M(4 * k - 1,:) = c3;
M(4 * k,:) = d3;
DrawTriangle(a3,b3,c3,d3);
else
stop = 1;
end
end

y = 4 * k;
while y < size(M,1)
M(4 * k + 1,:) = [];
end

t_End = toc;
disp(t_End);

s1=sprintf('%d molecules were placed', k);
disp(s1);

```

4. Conclusions

We developed the symmetrical packing algorithm for water molecules and implemented it in the MATLAB environment. In addition, a series of numerical experiments were carried out. The advantage of the implemented algorithm is that, unlike the common problem of the dense packing of balls, it works with tetrahedra. In addition, algorithm takes into account chemical bonds and is capable of laying more than a thousand molecules in less than 6 s, while packing almost 4 thousand molecules in just 45 s (see Table 1). However, packing 8.5 thousand molecules will take more than 3 min. Thus, the algorithm's running time increased almost 4.5 times after doubling the number of molecules. Therefore, for practical use, it is advisable to break down the entire volume so that the 4 thousand molecules are the largest number of molecules in each cell. Note also that this type of calculation has been performed non-quantum mechanically, and no specific quantum effects (e.g., deal with electron correlation) were taken into account in our model.

In order to illustrate the location of the water molecules obtained as a result of the algorithm, the program displays a graph with the image of tetrahedra (see Figures 2 and 3). The asterisks mark the vertices of type A.

Table 1. Results of numerical experiments.

X, pm	Y, pm	Z, pm	Volume, pm ³	Number of Molecules	Time, s
130	150	130	2.535×10^6	1	0.075858
290	330	290	2.7753×10^7	8	0.11618
290	330	550	5.2635×10^7	16	0.17162
290	690	550	1.1006×10^8	48	0.30239
570	690	550	2.1632×10^8	96	0.47833
980	980	980	9.4119×10^8	441	2.1261
1270	980	980	1.2197×10^9	567	2.8386
1270	980	1250	1.5558×10^9	729	3.6441
1550	1150	1250	2.2281×10^9	1089	5.6193
2550	1150	1250	3.6656×10^9	1782	10.339
2550	1750	1250	5.5781×10^9	2754	21.02
2550	1750	1850	8.2556×10^9	3978	45.271
2550	2550	2550	1.6581×10^{10}	8424	201.57

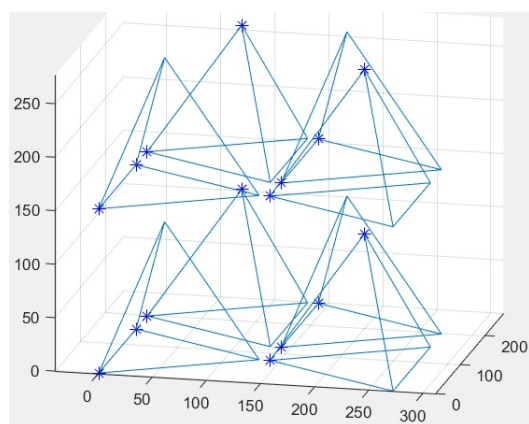


Figure 2. Illustration of the result of the algorithm for 8 molecules. The asterisks mark the vertices of type A.

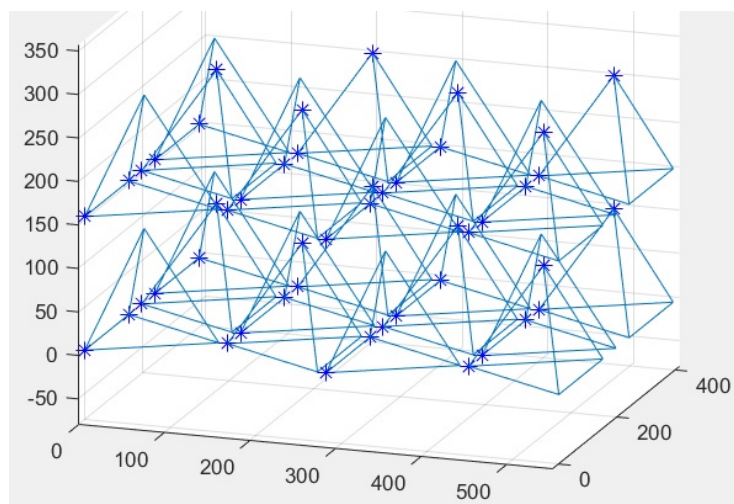


Figure 3. Illustration of the result of the algorithm for 24 molecules. The asterisks mark the vertices of type A.

Author Contributions: Conceptualization, E.A.G. and A.S.N.; methodology, E.M.L., E.A.G. and A.S.N.; software, E.M.L. and E.A.G.; validation, A.S.N.; formal analysis, E.M.L., E.A.G. and A.S.N.; investigation, E.M.L., E.A.G. and A.S.N.; resources, E.M.L., E.A.G. and A.S.N.; data curation, E.A.G. and A.S.N.; writing—original draft preparation, E.M.L., E.A.G. and A.S.N.; writing—review and editing, E.M.L., E.A.G. and A.S.N.; visualization, E.M.L. and E.A.G.; supervision, E.A.G. and A.S.N.; project administration, E.A.G. and A.S.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article and you can also email any questions and comments to authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Novikov, A.C. Symmetry in quantum and computational chemistry. *Symmetry* **2020**, *12*, 2028. [[CrossRef](#)]
2. Fischer, C.C. A Machine Learning Approach to Crystal Structure Prediction. Ph.D. Thesis, MIT, Cambridge, MA, USA, 16 August 2007.
3. Braun, N.; Huttner, G. Packing and molecular shape: Layers of centred hexagons as a guiding principle. *Acta Cryst.* **2005**, *B61*, 174–184. [[CrossRef](#)] [[PubMed](#)]
4. Thomasr, N.W. A novel characterization of organic molecular crystal structures for the purpose of crystal engineering. *Acta Cryst.* **2015**, *B71*, 463–477.

5. Falls, Z.; Avery, P.; Wang, X.; Hilleke, K.P.; Zurek, E. The XtalOpt evolutionary algorithm for crystal structure prediction. *J. Phys. Chem.* **2021**, *125*, 1601–1620. [[CrossRef](#)]
6. Oganov, A.R.; Glass, C.W. Crystal structure prediction using evolutionary algorithms: Principles and applications. *J. Chem. Phys.* **2006**, *124*, 24470.
7. Lyakhov, A.O.; Oganov, A.R.; Stokes, H.T.; Zhu, Q. New developments in evolutionary structure prediction algorithm USPEX. *Comput. Phys. Commun.* **2013**, *184*, 1172–1182. [[CrossRef](#)]
8. Peresypkina, E.V.; Blatov, V.A. Topology of molecular packings in organic crystals. *Acta Cryst.* **2000**, *B56*, 1035–1045. [[CrossRef](#)] [[PubMed](#)]
9. Gavezzotti, A.; Lo Presti, L. Molecular dynamics simulation of organic crystals: Introducing the CLP-dyncry environment. *J. Appl. Cryst.* **2019**, *52*, 1253–1263. [[CrossRef](#)]
10. Nekrasov B.V. *The Basics of General Chemistry. T.1*, 3rd ed.; Chemistry: Moscow, Russia, 1973; pp. 131–134.