

The MolRun molecular dynamics simulation program

Reference Manual

Nazarkin A.V.

andreyvn68@mail.ru

Content

1	Brief view	2
2	The Molecule	2
2.1	Mechanical model of the molecule	2
2.2	The force field.....	5
3	Calculation model	10
3.1	The equation of motion.....	11
3.2	Charge interaction.....	12
3.3	External electricity field.....	12
3.4	Resultant force	14
3.5	The elementary rotation	15
3.6	Parameterization of the orientation.....	18
3.7	Dynamics of rotation	21
3.8	Runge-Kutta methods	25
3.9	The energy calculation.....	28
3.10	The temperature calculation.....	29
3.11	The density calculation	29
3.12	The dipole moment calculation.....	30
3.13	The volume calculation	30
3.14	The pressure calculation.....	34
4	Program Management	37
4.1	Screen projection	37
4.2	Ray tracing.....	38
4.3	Program management	40
4.4	The 'WorkSpace' window	41
4.5	The 'Edit' window	44
4.6	The window 'Power potential'.....	46
4.7	The window 'Analysis'.....	46
4.8	The 'Automation' window.....	47
4.9	File structures.....	49
4.10	The xyzPlayer.exe utility	52
5	Test of the program	53
5.1	Conservation of energy test	53
5.2	Helium heat capacity.....	54
6	Conclusion.....	56

1 Brief view

The 'MolRun' is simulator 3D rigged-body ball-rod classical dynamics.

The program reproduces the interactive world of collective mechanics determined by intermolecular and Coulomb forces. These models have a wide area of applications from thermodynamics to biology. It is well-known that coulomb forces play a central role in the mutual recognition of biological molecules [1].

The program includes algorithms that can be grouped as follows.

The physical modeling algorithms: numerical integration of Newton's and Euler's equations, an algorithm for calculating the principal values and principal axes of the inertia tensor, an algorithm for calculating physical average values, an algorithm for calculating the volume given by an irregular set of points, a cluster search algorithm.

The visualizing of 3D world algorithms: an algorithm for constructing a central projection on an arbitrary plane, an algorithm for triangulating a surface, a ray tracing algorithm for a sphere and a plane element, an algorithm for a video file creating.

The superstructure algorithms: a series of calculation supervisor, a three-dimensional molecular editor.

Molecular mechanics concern different part of physics and mathematics: classical mechanics, and kinetics, thermodynamics, statistical physics, electrostatics, vector and tensor algebra, linear algebra and geometry, the theory of differential equations, and finally, the theory of numerical methods.

This description is an invitation to follow the same path that the author did, to perceive the theoretical material necessary for molecular dynamics modeling.

Perhaps the description will seem too detailed in terms of mathematical transformations, but remember how much time sometimes takes to find a lost sign in the program code, and also, thick books on mathematics are read faster than thin ones!

The MolRun program runs under the Windows operating system, it is made the portable style, does not require installation.

The program available at <http://www.tricemem.com>

2 The Molecule

A molecule is defined as a rigid formation of point masses, point charges, and central forces defined by interaction potentials. Each atom has a mass, charge and coefficients that determine the potential for interaction with other atoms.

2.1 Mechanical model of the molecule

The complete picture of the mass distribution of the molecule is determined by 6 values written in the form of a diagonally symmetric matrix

$$I = \begin{pmatrix} \sum m(y^2 + z^2) & -\sum mxy & -\sum mxz \\ -\sum myx & \sum m(x^2 + z^2) & -\sum myz \\ -\sum mzx & -\sum mzy & \sum m(x^2 + y^2) \end{pmatrix} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix}. \quad (2.1)$$

The inertia tensor I is a rank two tensor, a 3x3 matrix that is reduced to a diagonal form by rotating the coordinates. The inertia tensor defines an ellipsoid of inertia whose equation can be written explicitly

$$I_{xx}x^2 + I_{yy}y^2 + I_{zz}z^2 - 2I_{xy}xy - 2I_{xz}xz - 2I_{yz}yz = 1. \quad (2.2)$$

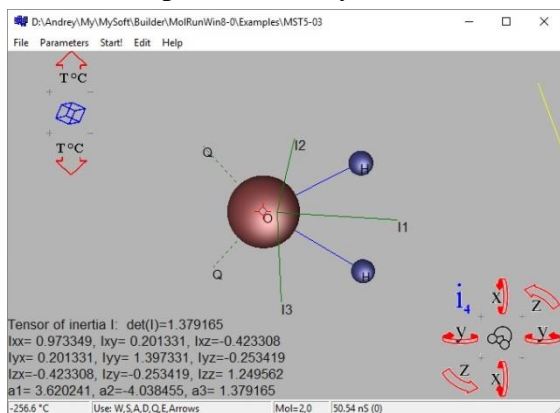
It would seem that if there is an explicit expression for the ellipsoid of inertia (2.2) is written, is it so difficult to find its axis? It turns out, yes! The problem of finding the principal axes of inertia is solved in [2] the authors find the gradient vector to the surface of the ellipsoid, which will be normal to the surface of the ellipsoid and determine the direction of the principal axes. The result is an eigenvalue problem and eigenvectors of the matrix (2.1). In General, you need to solve the problem of eigenvalues and eigenvectors of the matrix I or equation (2.3), as it is written in any course of mechanics. An example of solving the problem of finding eigenvalues and eigenvectors of the inertia tensor can be found in [3].

$$\det(I - \lambda E) = 0 \quad (2.3)$$

If determinant (2.3) equal to zero, then nontrivial solutions of a homogeneous system are presence. And for a heterogeneous system – it is the opposite, for existence of the solution, the determinant must not be zero. The first thing must be done is write the determinant (2.3) and solve the cubic equation (2.4)

$$\begin{aligned} \lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 &= 0 \\ a_1 &= I_{xx} + I_{yy} + I_{zz} \\ a_2 &= I_{xz}^2 + I_{yz}^2 + I_{xy}^2 - I_{xx}I_{yy} - I_{xx}I_{zz} - I_{yy}I_{zz} \\ a_3 &= \det(I) \end{aligned} \quad (2.4)$$

The λ coefficients are invariants of the tensor I , and they independent of rotation. You can rotate the selected molecule, look at the main axes of inertia, the inertia tensor and the values a_1 a_2 a_3 by selecting the information block **i3** (Fig. 2.1). The constancy of the quantities a_1 a_2 a_3 to prove that they are calculated correctly.



```

Izx=0.113315, Izy=0.000965, Izz=1.115745
det(I - PE) = 0
I^3 - (3.620222)I^2 - (-4.038414)I - (1.379146) = 0
Eigenvalues: I1=0.665863, I2=1.810111, I3=1.144248
Root error: 0.000000, -0.000000, 0.000000
Hesse matrix det=0.000000
0.030111, -0.042108, 0.113315
-0.042108, 1.142641, 0.000965
0.113315, 0.000965, 0.448882
diagonal matrix
1.000000, -1.396422, 3.763215
0.000000, 1.000000, 0.147106
0.000000, 0.000000, -0.000000
Eigenvector: -3.968932, -0.147106, 1.000000
  
```

Fig. 2.1. The Information block “i3” – principal axes of inertia tensor. Fig. 2.2 The principal axes found solution report.

The cubic equation (2.4) is solved through the trigonometric Fourier formulas, however, for large coefficients that appear in large molecules, the accuracy of trigonometric formulas is not sufficient, therefore, if necessary, to found roots in the vicinity of first approximation. The cubic equation is solved by Newton's method to achieve the specified accuracy 10^{-5} .

Founded eigenvalues $\lambda_1, \lambda_2, \lambda_3$ are the inertia moments about the principal axes. Denote them

$$I_X = \lambda_1 \quad I_Y = \lambda_2 \quad I_Z = \lambda_3, \quad (2.5)$$

It is not very important what indexes we assign to the moment values, because then we will find the vectors corresponding to them. The eigenvalues $\lambda_1, \lambda_2, \lambda_3$ are one by one substituted in the equation $I - \lambda E = 0$ and the eigenvectors $(a_{11}, a_{12}, a_{13}), (a_{21}, a_{22}, a_{23}), (a_{31}, a_{32}, a_{33})$ of tensor I are found, which are the main axes of inertia.

The eigenvectors are found by the Gauss method, the original matrix and the diagonal matrix are shown in the figure. 2.2. since linear dependence of the equations (2.3), one of the lines of the diagonal matrix contains zeros, which is a good sign confirming the correctness of the solution found. The found eigenvectors of the inertia tensor are stored in the program as direction cosines, which are calculated as (for all three vectors)

$$a_{11} = \frac{x_{1ev}}{r_{1ev}} \quad a_{12} = \frac{y_{1ev}}{r_{1ev}} \quad a_{13} = \frac{z_{1ev}}{r_{1ev}}, \quad r_{1ev} = \sqrt{x_{1ev}^2 + y_{1ev}^2 + z_{1ev}^2} \quad (2.6)$$

and are displayed on the Atoms tab (Fig. 2.3) in the fields Inertia Axis(1), Inertia Axis(2), Inertia Axis(3), and in the editing window (Fig. 2.1) are shown as vectors signed by I_1, I_2, I_3 .

If you click on the “Rotate Eigenvectors to WCS” button, the program will rotate all atoms of the edited molecule according to the formulas

$$\begin{aligned} x &= x'a_{11} + y'a_{12} + z'a_{13} \\ y &= x'a_{21} + y'a_{22} + z'a_{23} \\ z &= x'a_{31} + y'a_{32} + z'a_{33} \end{aligned} \quad (2.7)$$

as a result of rotation of the direction cosines given by the matrix (2.6), the main axes of inertia will be aligned with the axes of the world coordinate system (WCS), and the recalculated inertia tensor will take the diagonal form (Fig. 2.4).

Molecule, Atoms, Mass:		5	14.000000	
Velocity:		-0.019147	0.001276	0.007978
Angle Velocity:		0.020341	-0.052596	0.039641
Mass Center:		1.906024	0.285084	-0.507631
Inercia:		0.728427	156.125143	155.410746
Inercia Axe(1):		0.709805	0.091740	-0.698400
Inercia Axe(2):		-0.617938	0.557025	-0.554863
Inercia Axe(3):		-0.338122	-0.825414	-0.452069
Atom 0 Name:		8	0	
Atom Position:		1.877184	0.244467	-0.402101
Mass,Radius,Charge		12.000000	0.300000	0.000000
Color		200	100	100
Atom 1 Name:		1	H	
Atom Position:		1.529514	0.677705	-1.233869
Mass,Radius,Charge		1.000000	0.100000	0.230000
Color		100	100	200
Atom 2 Name:		1	H	
Atom Position:		2.628833	0.379374	-1.047913
Mass,Radius,Charge		1.000000	0.100000	0.230000
Color		100	100	200

Fig. 2.3 Edit molecule information window.

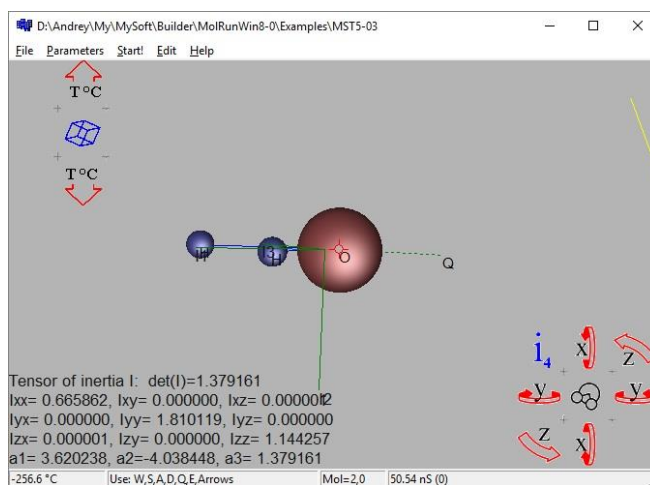


Fig. 2.4 Rotation that combines the main axes of inertia with the axes of WCS.

For monatomic molecules, the moment of inertia is considered as the inertia of the ball $\frac{5}{2}mR$, where R is the radius of the atom. For molecules whose atoms are arranged on the same

axis, the characteristic equation (2.3) will have only two roots. In this case, the program contains two eigenvectors, and the third vector is constructed as a vector product of the eigenvectors found.

The final touch is the coordinate system of triple vectors (I_1, I_2, I_3) must be a right-handed coordinate system. Or, more precisely, the orientation of the local coordinate system should be the same as the orientation of the world coordinate system.

2.2 The force field

The force field is constructed as the sum of atom-atomic potentials and Coulomb forces of point charges.

When choosing the type of potential describing the atom-atomic interactions, it is natural tendency a compromise between the physical meaningfulness and computational simplicity of the model function, today we know a large number of model potentials, the main of which are given below [4].

The Potential of *Sutherland*

$$U = \begin{cases} \infty, r < \sigma \\ -u(\sigma/r^6), r > \sigma \end{cases}. \quad (2.8)$$

The *Lennard-Jones* potential

$$U = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]. \quad (2.9)$$

The minimum of the potential is at the point $U = \sigma^{\sqrt[6]{2}}$, ε – point of intersection of the axis r , and the minimum of the potential well, respectively.

The potential of the “6-exp”

$$U = \frac{A}{r^6} - Be^{-Cr}, \quad (2.10)$$

The solutions for the inflection point are given by the transcendental equation.

The Potential of the *Buckingham*

$$U = Ae^{-Br} - \frac{\lambda}{r^6} - \frac{\lambda'}{r^8}, \quad (2.11)$$

Modified *Buckingham* potential

$$U = \frac{\varepsilon}{1-6/\alpha} \left\{ \frac{6}{\alpha} \exp \left[\alpha \left(1 - \frac{r}{r_m} \right) \right] - \left(\frac{r_m}{r} \right)^6 \right\}, \quad (2.12)$$

where ε – is the depth of the potential pit.

The *Morse* potential is used to describe the energy levels of the vibrational degrees of freedom. The Morse potential is finite at zero distance and is less consistent with experience at large distances. Similar properties have the potential of *Rydberg*.

$$U = D \left\{ \exp(-2\alpha(r-r_m)) - 2 \exp(-\alpha(r-r_m)) \right\}, \quad (2.13)$$

The potential *Varshni*

$$U = D \left\{ 1 - \frac{r_m}{r} \exp(-\beta(r^2 - r_m^2)) \right\}, \quad (2.14)$$

The potential *SPF*

$$U = b_0 \left(\frac{r - r_m}{r} \right)^2 \left\{ 1 + \sum b_n \left(\frac{r - r_m}{r} \right)^n \right\}, \quad (2.15)$$

Experience has shown that one term $n=1$ is clearly not enough for modeling the potential of SPF, the expression for the derivative is too cumbersome.

By discarding spectroscopic approximations, one can narrow the choice to three representations: The Lennard-Jones potential, the modified Buckingham potential, and the “6-exp” potential.

Below are graphs of these potentials for the parameters $r_{\min} = 1.5$, $\varepsilon(r_{\min}) = -4$. To ensure such conditions for the potential of the “6-exp”, we have to solve the transcendental equation, for the potential of the Lennard-Jones two parameters completely define the curve, the potential of the Buckingham remains a free parameter α , which can vary the slope of the curve (Fig. 2.7).

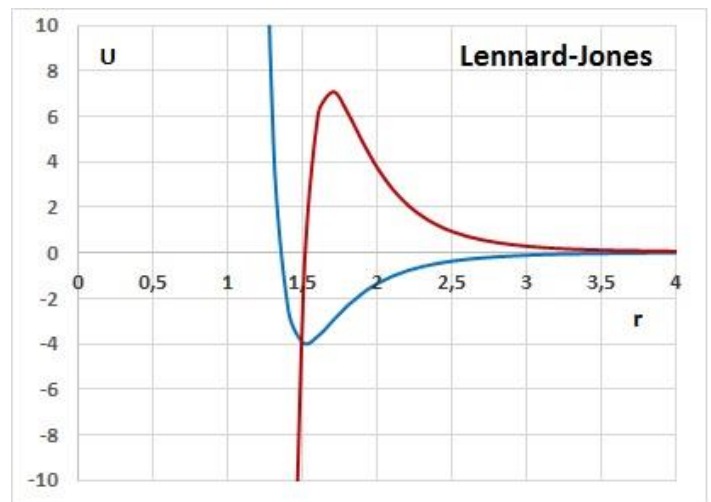


Fig. 2.5 The *Lennard-Jones* potential (blue) and it's derivative (red) $\varepsilon=4$; $\sigma=1.3486$.

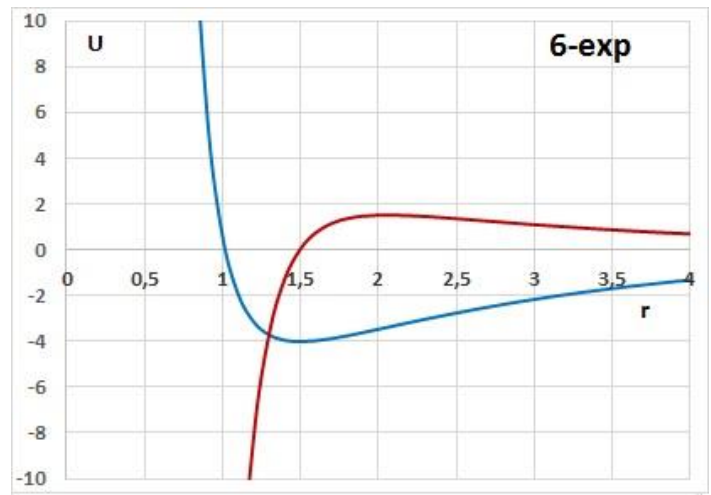


Fig. 2.6 The “6-exp” potential (blue) and it's derivative (red). $A=6.52$; $B=9.6887$; $C=0.5$.

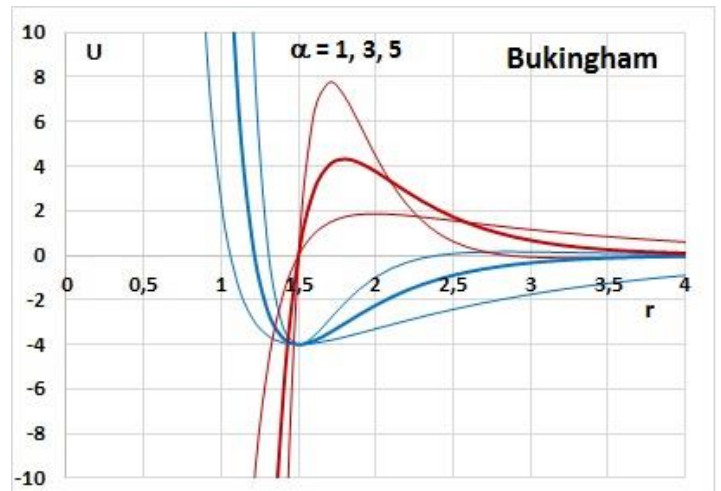


Fig. 2.7 The modified *Buckingham* potential (blue) and it's derivative (red).
 $\varepsilon=4.0$; $\alpha=1.0$; 3.0; 5.0; $r_m=1.5$

The presence of a free parameter and an explicit representation of physically meaningful values directly in the potential formula determined the choice of the type of potential curve in the form of a modified *Buckingham* potential

$$U = \frac{\varepsilon}{1-6/\alpha} \left\{ \frac{6}{\alpha} \exp \left[\alpha \left(1 - \frac{r}{r_m} \right) \right] - \left(\frac{r_m}{r} \right)^6 \right\}. \quad (2.16)$$

The derivative of the potential

$$U' = \frac{6\varepsilon}{1-6/\alpha} \frac{1}{r_m} \left\{ -\exp \left[\alpha \left(1 - \frac{r}{r_m} \right) \right] + \left(\frac{r_m}{r} \right)^7 \right\}. \quad (2.17)$$

After the potential approximation is chosen, it is necessary to fill the system with paired interaction atoms. Different papers use different approximations, therefore, below are the results of the approximation of different potential curves in the form of Buckingham potential.

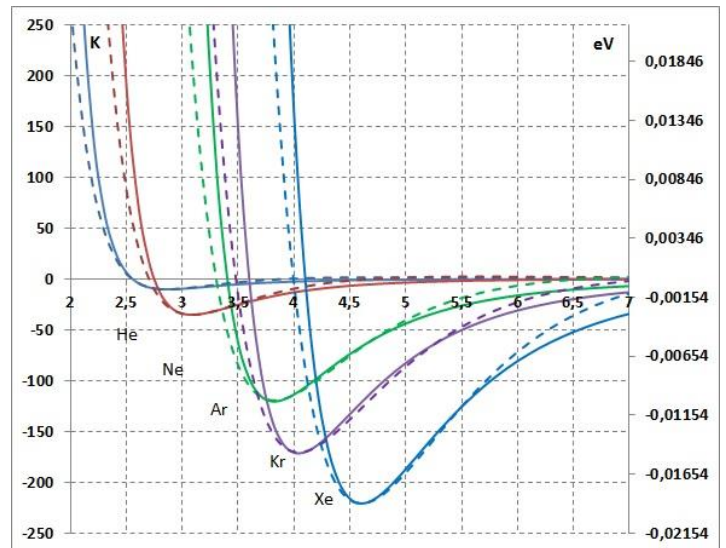


Fig 2.8 Pair potentials for inert gases by "6-12" (continuous line) and their approximation by *Buckingham* (dotted) [4].

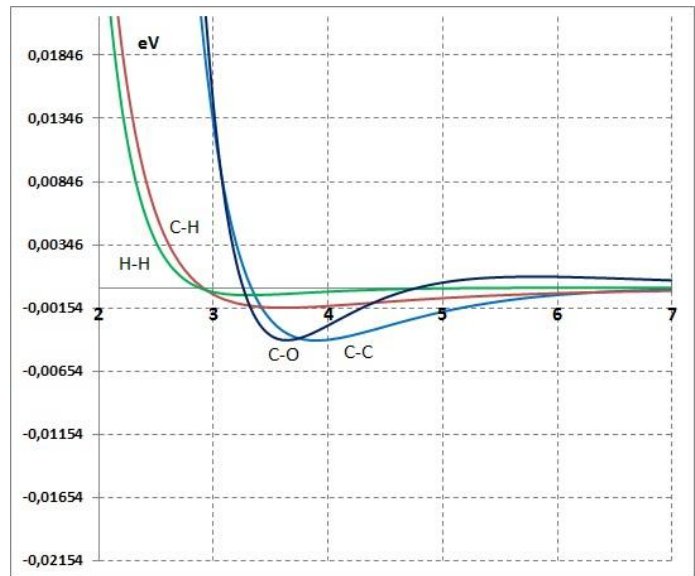


Fig. 2.9 Pair potentials in *Buckingham* form [5].

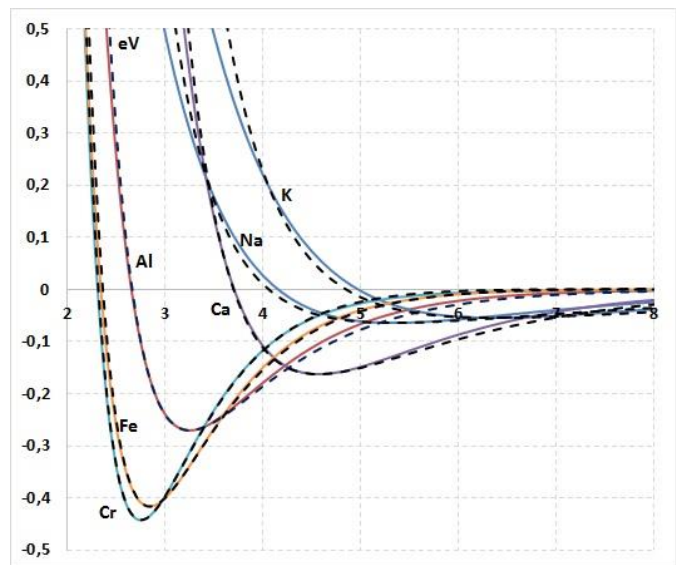


Fig. 2.10 Pair potentials in form *Morze* (continuous line) and *Buckingham* (doted) [6].

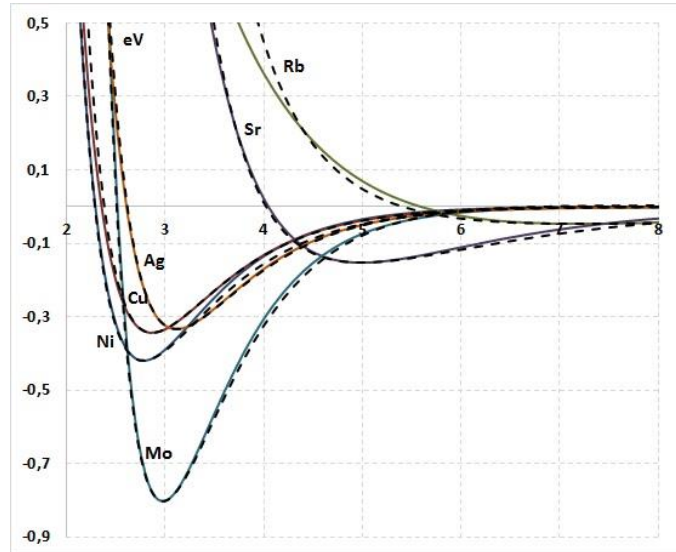


Fig. 2.11 Pair potentials in form *Morze* (continuous line) and *Buckingham* (doted) [6].

Рис. 2.12 Pair potentials in form *Morze* (continuous line) and *Buckingham* (doted) [6].

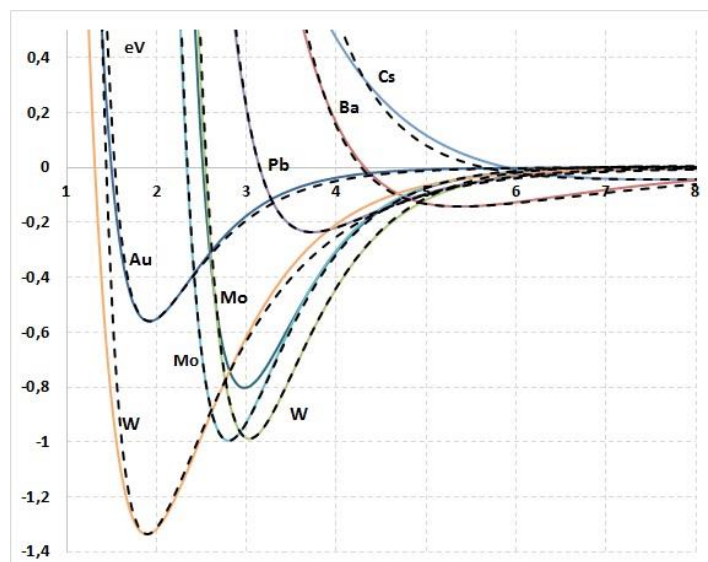


Table 2.1

The modified *Buckingham* potential parameters.

	ϵ , eV	α	r_m , +		ϵ , eV	α	r_m , +
He	0.000880	5.4	2.869	C-C	0.004139	4.5	3.9
Ne	0.003067	5.1	3.086	C-H	0.001487	3	3.61
Ar	0.010323	4.8	3.822	H-H	0.000562	5	3.31
Kr	0.014735	4.7	4.041	C-O	0.00415	5.5	3.64
Xe	0.019044	4.9	4.602				
Na	0,06334	1,8	5,336				
Al	0,2703	3,3	3,253				
K	0,05424	1,8	6,369				
Ca	0,1623	3	4,569				
Cr	0,4414	4,1	2,754				
Fe	0,4174	3,7	2,845				
Ni	0,4205	3,4	2,78				
Cu	0,3429	3,7	2,866				
Rb	0,04644	1,6	7,207				
Sr	0,1513	2,9	4,988				
Mo	0,8032	4,1	2,976				
Ag	0,3323	4	3,115				
Cs	0,04485	1,5	7,557				
Ba	0,1416	2,6	5,373				
W	0,9906	4,1	3,032				
Pb	0,2348	4,1	3,733				
Mo	0,997	3,9	2,8				
W	1,335	1,8	1,894				
Au	0,56	2,7	1,922				

It remains to put together selected pair of atoms and a set of parameters describing the interaction potential. The program, this is done as follows. The choice of potential is determined by the field 'atom type'. The 'atom type' field stores a single-byte value, which is the default number of

the atom in the periodic table, this parameter is available on the 'Atoms' tab, Fig. 2.13.

On the 'Force' tab, Fig. 2.14 the pairs of atoms with the same numbers and three parameters ϵ , α , r_m , determining the interaction potential (2.17) are given. The 'Force' tab of the 'Edit' window is a reflection of the file with the extension *.ptn, which stores a list of all potentials used in the scene. File format *.ptn is described in § 4.8.

On the 'Molecule' tab of the same 'Edit' window, there is a 'Check Force List' button, which goes through all possible pairs of atoms of the loaded scene and forms lines with the description of interaction potentials. If there are parameters of the pair interaction potential in the periodic table, for atom 'A', and for atom 'B', then the parameters

$$\epsilon_{A-B} = \frac{\epsilon_{A-A} + \epsilon_{B-B}}{2}, \quad \alpha_{A-B} = \frac{\alpha_{A-A} + \alpha_{B-B}}{2}, \quad r_{m,A-B} = \frac{r_{m,A-A} + r_{m,B-B}}{2}.$$

An attempt was made in the article [7] to find a universal rule for combining the interatomic interaction potentials. At the current stage of the program implementation, the potentials are combined as arithmetic means for the coefficients.

In General, the contents of the field 'atom type' can be disposed of quite freely, since it is used only for communication with the list of interaction potentials. For example, the value "0" is used for points where the charge density is localized. Values greater than 90 can be used if you want to play with arbitrary particles that are not atoms from the periodic table.

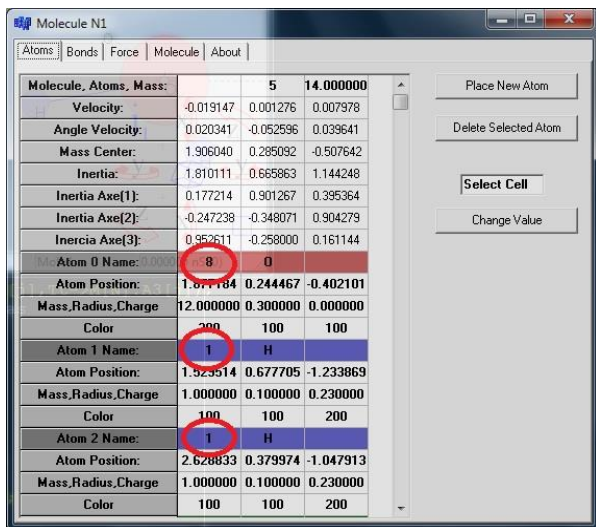


Fig. 2.13 There are fields "atom type", pointed

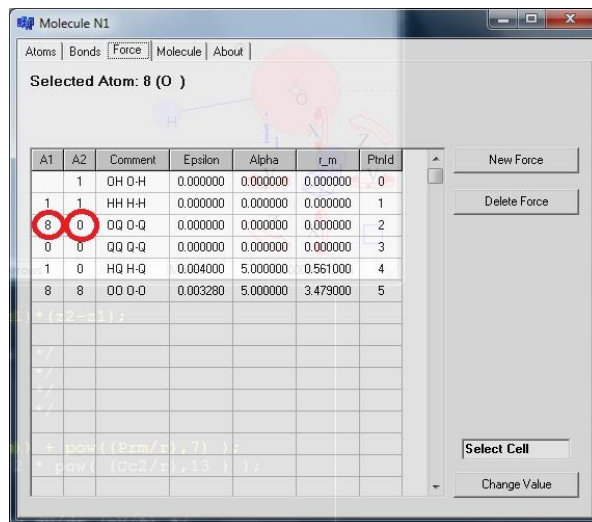


Fig. 2.14 Interaction potential parameters for atoms 8-0.

3 Calculation model

Bringing the equations to dimensionless form allows to 'drive' the numerical values of the variables in the range close to one. Usually this problem is solved by dividing the variable by some characteristic constant. But this problem for the microcosm has already been solved, there are a large number of units of measurement introduced to describe the phenomena of molecular scale.

Below, we will bring all the necessary equations to the scale of numerical values within no more than hundreds by simply selecting the most appropriate units of measurement. We write bona fide laws of classical mechanics in units of molecular scale.

3.1 The equation of motion

Let there be an interaction potential U , which is a change in the energy of the system when an element is shifted by a unit distance. The force due to the change in potential is the ratio of the energy increment to the coordinate increment. Units of variables will be signed in the lower index, and enclosed in brackets, then,

$$F_{(N)} = \frac{\Delta U_{(J)}}{\Delta x_{(m)}} = \frac{(\Delta U_{(eV)} \times 1,6021 \times 10^{-19})_{(J)}}{(\Delta x_{(A)} \times 10^{-10})_{(m)}} = \frac{\Delta U_{(eV)}}{\Delta x_{(A)}} \times 1,6021 \times 10^{-9} = F_{(eV/A)} \times 1,6021 \times 10^{-9} \quad (3.1)$$

For the equation of motion, obtain

$$\frac{d^2 x_{(m)}}{dt_{(S)}^2} = \frac{F_{(N)}}{m_{(Kg)}} = \frac{\left(\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} \times 1,6021 \times 10^{-9} \right)_{(N)}}{\left(m_{(amu)} \times 1,6597 \times 10^{-27} \right)_{(Kg)}} = \frac{F_{(eV/A)}}{m_{(amu)}} \times 0,9653 \times 10^{18}, \quad (3.2)$$

$$\frac{d^2 (x_{(A)} \times 10^{-10})_{(m)}}{d(t_{(?)}) \times 10^{-n})_{(S)}^2} = \frac{d^2 x_{(A)}}{dt_{(?)^2}} \times 10^{2n-10}.$$

Choose the time scale n so that the factor 10^{18} is reduced, take 10 femtoseconds as a unit of time, then

$$t_{(S)} = t_{(10fs)} \times 10^{-14}, \quad (3.3)$$

and the equation of motion becomes

$$\frac{d^2 x_{(A)}}{dt_{(10fs)}^2} = \frac{F_{(eV/A)}}{m_{(amu)}} \times 0,9653. \quad (3.4)$$

That is, the motion of atom a mass of 1 u at the scale of 1 Angstrom, 10 femtoseconds, practically does not differ from the movement of a body of mass 1 kg at the scale of a meter a second. The equation of rotation written for one degree of freedom is as follows

$$\frac{d^2 \alpha_{(rad)}}{dt_{(S)}^2} = \frac{M_{(N \times m)}}{I_{(Kg \times m^2)}}. \quad (3.5)$$

We write down the transformation of the moment of force and moment of inertia to the same units as the equation of motion, while the force is still considered in units of “electronvolt on Angstrom”,

$$\frac{M_{(N \times m)}}{I_{(Kg \times m^2)}} = \frac{\left(\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} \times 1,6021 \times 10^{-9} \right)_{(N)} \times (x_{(A)} \times 10^{-10})_{(m)}}{\left(m_{(amu)} \times 1,6597 \times 10^{-27} \right)_{(Kg)} \times (x_{(A)} \times 10^{-10})_{(m)}^2} = \frac{\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} x_{(A)}}{m_{(amu)} x_{(A)}^2} \times 0,9653 \times 10^{28} \quad (3.6)$$

$$\frac{d^2 \alpha_{(rad)}}{dt_{(S)}^2} = \frac{d^2 \alpha_{(rad)}}{d(t_{(10fs)} \times 10^{-14})_{(S)}^2} = \frac{d^2 \alpha_{(rad)}}{dt_{(10fs)}^2} \times 10^{28}$$

A “miracle” happened, all the exponents were destroyed, and the equation of rotation takes the following form

$$\frac{d^2 \alpha_{(rad)}}{dt_{(10fs)}^2} = \frac{\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} x_{(A)}}{m_{(amu)} x_{(A)}^2} \times 0,9653 \quad (3.7)$$

Total, the units of measurement for the atomic scale in which the program operates are as follows:

Time	$t_{(s)} = t_{(10fs)} \times 10^{-14}$	(3.8)
Distance	$x_{(m)} = x_{(A)} \times 10^{-10}$	
Velocity	$v_{(m/s)} = v_{(A/10fs)} \times 10^4$	
Mass	$m_{(kg)} = m_{(amu)} \times 1,6597 \times 10^{-27}$	
Force	$\frac{\Delta U_{(eV)}}{\Delta x_{(A)}}$	
Moment of Inertia	$m_{(amu)} x_{(A)}^2$	
Charge	$q_{(C)} = q_{(e)} \times 1,6021 \times 10^{-19}$	

3.2 Charge interaction

To write Coulomb's law

$$F_{(N)} = \frac{1}{4\pi\epsilon_0 \left(\frac{C^2}{N \times m^2} \right)} \frac{q_{(C)} q_{(C)}}{x_{(m)}^2} \quad (3.9)$$

in units of atomic scale, use the coefficient (3.1) and write the force in units (eV/Å)

$$\begin{aligned} F_{(eV/A)} &= \frac{1}{4\pi\epsilon_0} \frac{q_{(C)} q_{(C)}}{x_{(m)}^2} \Big/ 1,6021 \times 10^{-9} = \frac{1}{4\pi\epsilon_0} \frac{\left(q_{(e)} q_{(e)} \left(1,6021 \times 10^{-19} \right)^2 \right)_{(C^2)}}{\left(x_{(A)} \times 10^{-10} \right)_{(m)}^2} \Big/ 1,6021 \times 10^{-9} = \\ &= \frac{q_{(e)} q_{(e)}}{x_{(A)}^2} \times \frac{1,6021}{4 \times 3,14159 \times 8,854} \times \frac{10^{-38}}{10^{-9} \times 10^{-20} \times 10^{-12}} = \frac{q_{(e)} q_{(e)}}{x_{(A)}^2} \times 14,3993. \end{aligned} \quad (3.10)$$

3.3 External electricity field

Let there be an external electric field of tension E (V/m), the field will act on the charge with force

$$F_{(N)} = q_{(C)} \times E_{(V/m)} = \left(q_{(e)} \times 1,6021 \times 10^{-19} \right)_{(C)} \times E_{(V/m)}. \quad (3.11)$$

To compare with (3.1), obtain

$$F_{(eV/A)} = q_{(e)} \times E_{(V/m)} \times 10^{-10}. \quad (3.12)$$

The resulting factor 10^{-10} will not disappear, which means that the intramolecular electric fields are 10^{10} times greater than the external one! How, then, does the dielectric between the capacitor plates work, which gives appreciable response at millivolts?

To overcome a multiplier of the order of 10^{10} , we need a factor of comparable scale, such a factor can only be the number of particles. With the number of particles of the order of 10^{23} , the fluctuations caused by molecular chaos are so well averaged that the asymmetry introduced by an external force, even by 10 orders of magnitude less, gives an average appreciable response.

However, in our case, understanding the mechanism of action of the external electric field does not bode well. The program manipulates hundreds of atoms, with such a number of particles, direct averaging of the orientation contribution to the polarizability of a unit volume is not possible.

Let the molecule have a dipole moment vector μ , at an angle Θ with the x axis, the external electric field vector E , coincides with the direction of the axis x . The moment of forces is acting to molecule is $\mu E \sin(\theta)$, to convert the moment of forces to atomic scale units (3.8),

$$\left(\left(\mu_{(D)} \times 3.3356 \times 10^{-30} \right)_{(C \times m)} \times E_{(V/m)} \sin(\theta) \right)_{(N \times m)} = \left(M_{(eV/A \times A)} \times 1.6021 \times 10^{-19} \right)_{(N \times m)}, \quad (3.13)$$

where $M_{(eV/A \times A)}$ - the moment of forces acting to a molecule. The force expressed in units of electron volts per angstrom, and the distance is expressed in angstroms. From (3.13)

$$M_{(eV/A \times A)} = \mu_{(D)} E_{(V/m)} \sin(\theta) \times 2.082 \times 10^{-11}, \quad (3.14)$$

approximately the same ratio as (3.12).

The dipole energy in the external field $-\mu E \cos(\theta)$, so the distribution function [8]

$$f(\theta) = \frac{e^{-a \cos(\theta)}}{\int_0^\pi e^{-a \cos(\theta)} d\theta} = \frac{e^{-a \cos(\theta)}}{\pi I_0(a)}, \quad a = \frac{\mu E}{kT}, \quad (3.15)$$

where $I_0(a)$ - modified Bessel function of the I-kind.

A water molecule dipole moment is 1.8 D, the dimensionless value a at zero degrees Celsius and a field of 1000 V/m will be 1.6×10^{-6} , the probability density (Fig. 3.1) does not differ from the uniform distribution.

So for example the relation $f(0)/f(\pi/2)$ for the water molecule in uniform field 1000 V/m at temperature 0°C , gives the value 1.0000016. So, the probability of the dipole orientation in the direction of the external field differs from the probability of orientation perpendicular to the direction of the field in the sixth sign.

Let's average the magnetic moment projection of the x axis according to the distribution (3.15), for a sample containing N molecules

$$M_x = \frac{N}{\pi I_0(a)} \int_0^\pi \mu \cos(\theta) e^{-a \cos(\theta)}, \quad (3.16)$$

on the integration interval from 0 to $\pi/2$, the projection of the magnetic moment will give a contribution with a positive sign, on the interval from $\pi/2$ to π , the projection of the magnetic moment will give a contribution with a negative sign. Due to the fact that the distribution is close to

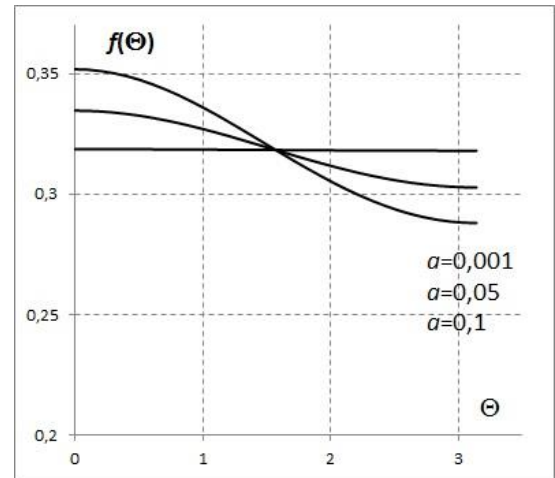


Fig. 3.1 The distribution density from the dipole moment vector orientation angle.

uniform, the integral is a very small quantity, and N is a very large quantity.

Considering that a small value, expand the exponent and limited to the first terms of expansion, also replace the zero-order Bessel function from a small argument with unit

$$M_x = \frac{N}{\pi I_0(a)} \int_0^\pi \mu \cos(\theta)(1 - a \cos(\theta)) = -\frac{Na\mu}{2}. \quad (3.17)$$

The formula for the average magnetization of the sample includes a large value N and a small value a as a multiplication, so there is a response with a very small asymmetry in the distribution of dipoles.

Thus, when modeling molecular dynamics, it will not be possible to visually distinguish the influence of an external field on the behavior of hundreds of particles, unless the external fields are reduced to an atomic scale, such as, for example, the needle tip of a tunneling microscope.

3.4 Resultant force

After the pair of interacting particles is determined and the distance r between them is calculated, the formula (2.17) is the scalar value of the force F , which acts on the atom with coordinates (x_1, y_1, z_1) from the atom with coordinates (x_2, y_2, z_2) . Let's construct a vector directed from point 1 to point 2, make it of unit length and multiply by the found value F

$$F_x = F \frac{x_2 - x_1}{r}, \quad F_y = F \frac{y_2 - y_1}{r}, \quad F_z = F \frac{z_2 - z_1}{r}. \quad (3.16)$$

Find the moment of force acting on the molecule at center of mass point with coordinates $r_0 = (x_0, y_0, z_0)$.

$$\vec{M} = (\vec{r}_1 - \vec{r}_0) \times \vec{F}, \quad (3.17)$$

in the matrix form

$$\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{vmatrix} i & j & k \\ x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ F_x & F_y & F_z \end{vmatrix},$$

in the component form

$$\begin{aligned} M_x &= (y_1 - y_0)F_z - (z_1 - z_0)F_y \\ M_y &= (z_1 - z_0)F_x - (x_1 - x_0)F_z \\ M_z &= (x_1 - x_0)F_y - (y_1 - y_0)F_x \end{aligned} \quad (3.18)$$

Let's deal with the signs and direction of rotation. Consider two cases, in the first, the positive non-zero component of the force vector is F_y , in the second, F_x , and the force vector lies at the XY plane. The moment vector of forces will have only one nonzero component, M_z , which is shown in Fig. 3.2. From figure 3.2, we see that if you look in the direction of the moment force vector, the rotation should always be clockwise.

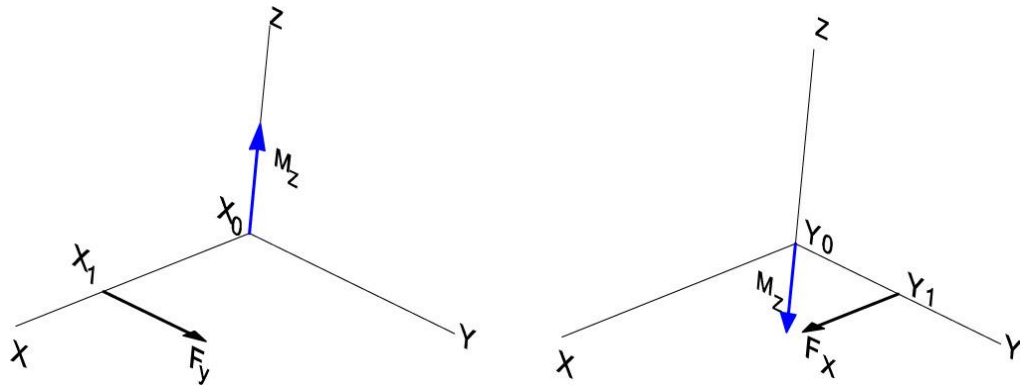


Fig. 3.2. There are force vector and the moment of forces vector.

The moment of force vector cause to change of the angular velocity vector. If at the initial moment the body isn't moving, under the moment of forces, it will begin to rotate at the angular velocity $\vec{\omega}_2 - \vec{\omega}_1 = \frac{1}{I} \vec{M} \cdot \Delta t$, in such a situation, the vector of the angular velocity is directed the same as the moment of forces vector.

The next question is the transformation of atom coordinates to reflect the action of the angular velocity. Since we have found that the rotation under an act of the moment of forces occurs clockwise, if we look in the direction of the vector of the moment of forces, find the coordinate transformation corresponding to this rule.

3.5 The elementary rotation

Flat rotation can be attributed to elementary transformations, however, flat rotation is the basic transformation that determines all combinations of rotations recorded in the matrix representation, or in the quaternion algebra. Because of the special importance of rotation transformations, they are written in detail.

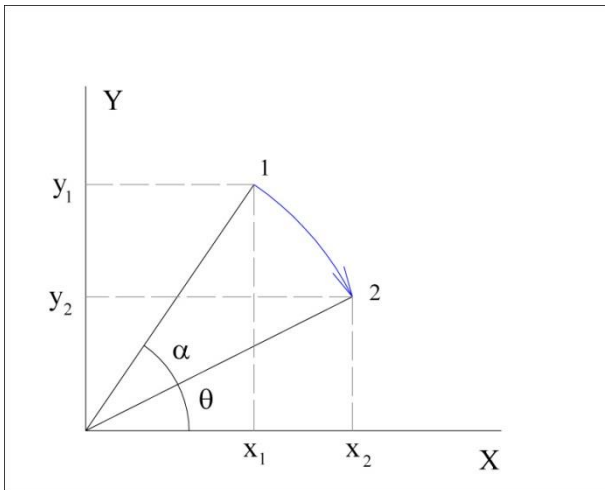


Fig. 3.3. Radius vector rotation.

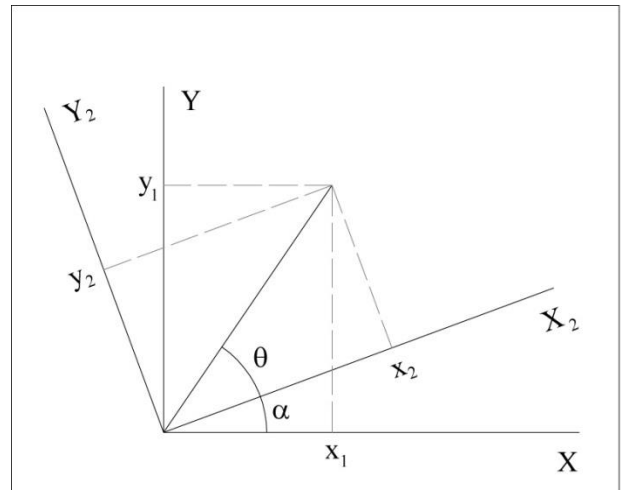


Fig. 3.4. Coordinate system rotation.

Consider the rotation of the radius vector to the angle α along the z-axis, rotation will lead to the transformation of the coordinates x, y (Fig. 3.3). We write the trigonometric identity

$$\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha) \quad (3.19)$$

$$\sin(\theta + \alpha) = \sin(\theta)\cos(\alpha) + \cos(\theta)\sin(\alpha).$$

Replacing $\cos(\theta)$ and $\sin(\theta)$ ratios of the triangles and multiplying all the length to the radius vector, we obtain the transform coordinates of point 2 coordinates of point 1

$$\begin{aligned} x_1 &= x_2 \cos(\alpha) - y_2 \sin(\alpha) \\ y_1 &= x_2 \sin(\alpha) + y_2 \cos(\alpha) \end{aligned} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}. \quad (3.20)$$

Find the transformation, the coordinates of point 1 to the coordinates of point 2. Multiply the upper equality by $\cos(\alpha)$, and the lower one by $\sin(\alpha)$ and add them, then multiply the upper equality by $\sin(\alpha)$, and the lower one by $\cos(\alpha)$ and subtract the upper one from the lower one.

$$\begin{aligned} x_2 &= x_1 \cos(\alpha) + y_1 \sin(\alpha) \\ y_2 &= -x_1 \sin(\alpha) + y_1 \cos(\alpha) \end{aligned} \quad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \quad (3.21)$$

Now consider the rotation of the coordinate axes (Fig. 3.4). Same as before for triangles

$$\begin{aligned} \cos(\theta + \alpha) &= \frac{x_1}{r} = \frac{x_2}{r} \cos(\alpha) - \frac{y_2}{r} \sin(\alpha) \\ \sin(\theta + \alpha) &= \frac{y_1}{r} = \frac{y_2}{r} \cos(\alpha) + \frac{x_2}{r} \sin(\alpha). \end{aligned} \quad (3.22)$$

from which,

$$\begin{aligned} x_1 &= x_2 \cos(\alpha) - y_2 \sin(\alpha) \\ y_1 &= x_2 \sin(\alpha) + y_2 \cos(\alpha) \end{aligned} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}. \quad (3.23)$$

Exactly the same as (3.20), but in the first case the rotation was clockwise, so the coordinate system is rotated counterclockwise. Previously, we have already obtained the inverse transformation, which for the rotation of the coordinate system has the form

$$\begin{aligned} x_2 &= x_1 \cos(\alpha) + y_1 \sin(\alpha) \\ y_2 &= -x_1 \sin(\alpha) + y_1 \cos(\alpha) \end{aligned} \quad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \quad (3.24)$$

Total, we conclude that the rotation of the radius vector is equivalent to the rotation of the coordinate system in the direction of the reverse rotation of the radius vector.

In the monograph [9] is called *the active point of view* the transformation that rotates the radius vector in which the coordinate system remains unchanged. Rotation as a transformation of the coordinate system is called *a passive point of view*.

Now let's expand the space to three-dimensional and discuss the choice of axis orientation. Obviously, due to the isotropy of space, the first two directions OX and OY can be chosen arbitrarily. Now the space has acquired two selected directions, consider the rotation from the OX axis to the OY axis and by the rule of the right screw determine the direction of the OZ axis. But we have no priority what the side to look at the XY plane. When viewed from the "opposite" side, the clockwise direction will look like the counterclockwise. We obtain that the determination of the coordinate system orientation depends on the orientation of the observer.

To distinguish the direction of the right rotation from the left rotation, without having some template, it seems, impossible.

Returning to §3.1, and using the transformations of the elementary rotation, to write the transformation of rotations, which will make the body rotating at an angular velocity $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$.

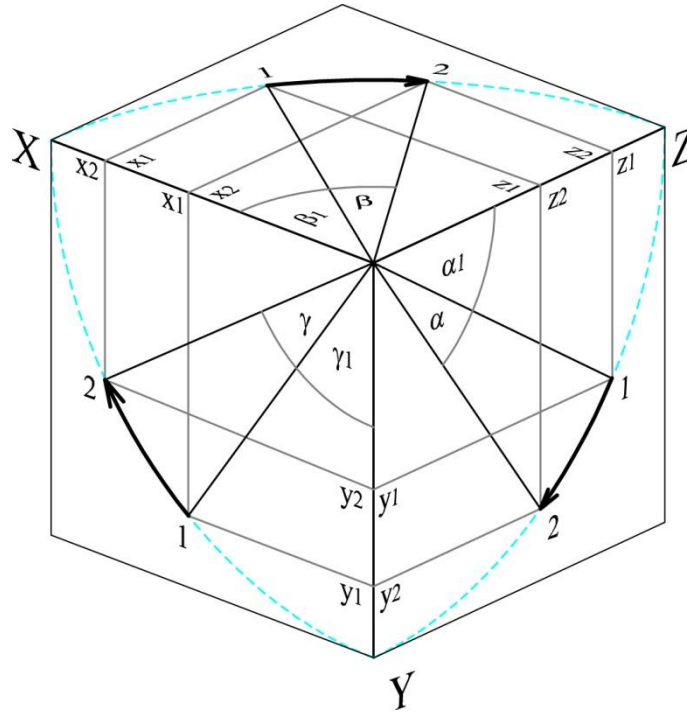


Fig. 3.5. Rotation transformations for each plane of a rectangular coordinate system.

As before, we write down the sine of the sum of the angles and the cosine of the sum of the angles shown at Fig. 3.5, we obtain the rotation matrix for each of the quadrants of the coordinate system.

If we look in the direction of the angular velocity vector, the rotation is clockwise, as we demand at §3.4:

-around X axis:

$$\begin{pmatrix} y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} y_1 \\ z_1 \end{pmatrix}, \quad (3.25)$$

-around Y axis:

$$\begin{pmatrix} x_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix} \begin{pmatrix} x_1 \\ z_1 \end{pmatrix}, \quad (3.26)$$

-around Z axis:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\gamma) & \sin(\gamma) \\ -\sin(\gamma) & \cos(\gamma) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \quad (3.27)$$

Now add the matrix to the size of 3x3 and multiply with Maple

$$\begin{pmatrix} \cos \beta \cos \gamma & \sin \gamma \cos \alpha + \sin \beta \sin \alpha \cos \gamma & \sin \gamma \sin \alpha - \sin \beta \cos \alpha \cos \gamma \\ -\cos \beta \sin \gamma & \cos \gamma \cos \alpha - \sin \beta \sin \alpha \sin \gamma & \cos \gamma \sin \alpha + \sin \beta \cos \alpha \sin \gamma \\ \sin \beta & -\sin \alpha \cos \beta & \cos \alpha \cos \beta \end{pmatrix}. \quad (3.28)$$

We have obtained a matrix that implements a sequence of rotations around The X, Y, Z axes.

Now ease to write infinitesimal turn, which takes place in the interval dt . Limited by the values of the first order of smallness, discard the product of two or more sinuses, replace the cosines

with units, and sine values replaced by their arguments

$$\begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}. \quad (3.29)$$

3.6 Parameterization of the orientation

The choice of parameters that determine the orientation of the body, is not a trivial task.

Transformations (3.20) are sufficient to describe the Euler angles. Euler angles are a sequence of rotations relative to three axes, each of the rotations is carried out relative to the current position of the axes, the standard sequence of rotations Z, X, Z. Nothing special about the fact that the Z axis occurs twice, it also changes its position as a result of the previous rotation. The sequence of turns on angles (α, β, γ) is shown in Fig. 3.6, at the initial time both coordinate systems are the same.

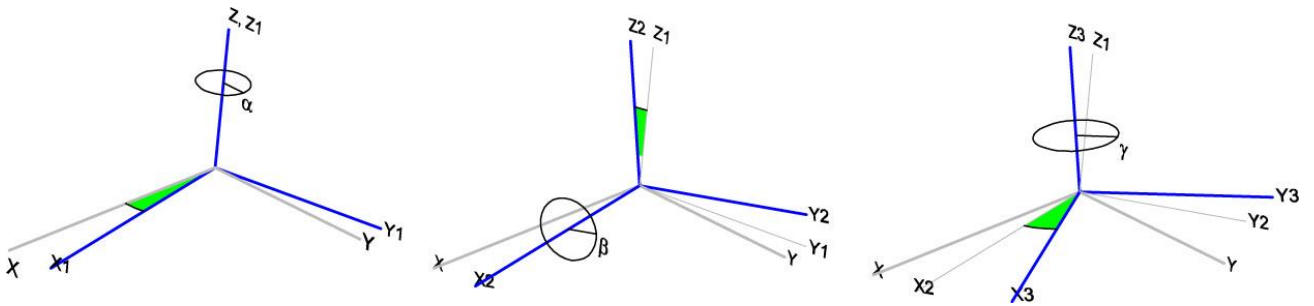


Fig. 3.6. Euler angles are a sequence of rotations relative to the current position of the coordinate axes.

Now we can write down the matrix of plane rotations around the axes Z, X, Z, and multiplying them to obtain the resulting rotation transformation. We use a matrix of the form (3.20), the rotation operation will be interpreted as a transformation of the coordinate system (passive point of view)

$$\begin{aligned} A_Z &= \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ A_{X2} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) \\ 0 & -\sin(\beta) & \cos(\beta) \end{pmatrix} \\ A_{Z3} &= \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (3.30)$$

To multiply $A_{Z3}A_{X2}A_Z$ by the Maple, we obtain the result turn:

$$A = \begin{pmatrix} \cos \gamma \cos \alpha - \sin \gamma \cos \beta \sin \alpha & \cos \gamma \sin \alpha + \sin \gamma \cos \beta \cos \alpha & \sin \gamma \sin \beta \\ -\sin \gamma \cos \alpha - \cos \gamma \cos \beta \sin \alpha & -\sin \gamma \sin \alpha + \cos \gamma \cos \beta \cos \alpha & \cos \gamma \sin \beta \\ \sin \beta \sin \alpha & -\sin \beta \cos \alpha & \cos \beta \end{pmatrix}. \quad (3.31)$$

The rotation matrix coinciding with (3.31) is found in many courses of mechanics [10, 9].

In the numerical implementation of the rotation operation, it is most convenient to store the orientation in the form of a matrix of guide cosines, since it is also a matrix of rotation.

We write a matrix that stores the orths of the axes of the movable coordinate system (Fig. 3.5) $X_3=(a_{11}, a_{12}, a_{13})$, $Y_3=(a_{21}, a_{22}, a_{23})$, $Z_3=(a_{31}, a_{32}, a_{33})$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}. \quad (3.32)$$

Equating (3.31) and (3.32) we can express the Euler angles through the guide cosines

$$\begin{aligned} \alpha &= \arcsin\left(\frac{a_{31}}{\sqrt{1-a_{33}^2}}\right), \\ \beta &= \arccos(a_{33}), \\ \gamma &= \arccos\left(\frac{a_{33}}{\sqrt{1-a_{33}^2}}\right) \text{ или } \gamma = \operatorname{arctg}\left(\frac{a_{13}}{a_{23}}\right). \end{aligned} \quad (3.33)$$

If $\beta=0$, we get $a_{33}=1$ and divide by zero! The problems with small angles of nutation, chase the description of the orientation using Euler angles continuously.

From the matrix guides of the cosines of (3.32) we can obtain the parameters of *Rodrigues-Hamilton* [11], they are the same, quaternion $\Lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$,

$$\begin{aligned} \lambda_0 &= \frac{1}{2} \sqrt{1 + a_{11} + a_{22} + a_{33}} \\ \lambda_1 &= \frac{a_{32} - a_{23}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}} \\ \lambda_2 &= \frac{a_{13} - a_{31}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}} \\ \lambda_3 &= \frac{a_{21} - a_{12}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}} \end{aligned}. \quad (3.34)$$

And the inverse transform [12]

$$A = \begin{pmatrix} 2(\lambda_0^2 + \lambda_1^2) - 1 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 2(\lambda_0^2 + \lambda_2^2) - 1 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 2(\lambda_0^2 + \lambda_3^2) - 1 \end{pmatrix}. \quad (3.35)$$

The quaternion can be considered as the rotation operator of the coordinate system, since there is a one-to-one correspondence between the quaternion and the rotation matrix.

However, not every matrix of guiding cosines can be correctly transformed into *Rodrigo-Hamilton* parameters and vice versa. For example, the left three vectors (V_1, V_2, V_3) are not aligned

with the right coordinate system. The transformation into the parameters of *Rodrigo-Hamilton* and back leads to the fact that the matrix before the transformation and after the transformation do not coincide. When the algorithm the main axis of inertia calculation (§ 2.1), chose the left three vectors (I_1, I_2, I_3), the program swaps the two vectors and the corresponding values of the components of the moments of inertia.

It should be noted another difference between the rotation matrix (3.35) obtained through the *Rodrigo-Hamilton* parameters, and the rotation matrix (3.31) constructed from trigonometric functions. If you rotate the model for each time step, the program generates a recursive algorithm of the form $X^{(i+1)}=A \cdot X^{(i)}$, where A is the rotation matrix, X is the radius vector of coordinates of the model points.

The rotation matrix, composed of trigonometric functions, contains the alternating sign of the error, which does not lead to the accumulation of geometric distortions of the rotating model. In the case of the rotation matrix composed through the *Rodrigo-Hamilton* parameters, the recursive algorithm cannot be used, because it accumulates an error, as a result of which the model is deformed.

To avoid deformation of the model, the program stores the original image of the molecule with $X^{(init)}$ coordinates, deployed so that the main axes of inertia coincide with the axes of the world coordinate system. The rotation transformation is performed by the matrix guides of the cosines of A

$$X^{(init)} = A(X - X_0), \quad (3.36)$$

X_0 - the coordinate vector of the center of mass of the molecule.

The reverse turn pulls the molecule out of the “storage” indicated by the index (*init*) and places the molecule in the same position where it was. This rotation is performed by the transposed matrix A , which is calculated by the quaternion parameters (3.35), and the quaternion initialization is performed by the formulas (3.34),

$$X = A^T X^{(init)} + X_0. \quad (3.37)$$

No need to store of the inertia tensor main axes, the quaternion matrix A is the matrix composed of the guiding cosines of the inertia tensor principal axes.

The rotation matrix must meet the following rules [11] the determinant of the rotation matrix is unit, the axis of rotation is the eigenvector of the rotation matrix, the eigenvector corresponds to the eigenvalue is 1.

At very large calculation times, the rotation matrix becomes a weak point of the program code, because it is repeatedly recalculated through the values obtained in the previous step, which leads to the accumulation of an error. For example, heating 58 helium atoms (Fig. 5.2), the total calculation time was 1500 pS, with a time step of $0.001 \times 10 \text{ fS} = 10^{-14} \text{ S}$, 150×10^6 multiplications of the rotation matrix were performed. The quaternion norm conservation problem described in [11].

The ‘**Debug**’ tab in the molecule editor provides the ‘**Rotation Matrix Check**’ button, which checks the matrix of guiding cosines according to the following formulas [11].

By the elements of the matrix A (3.53) the guiding cosines of the eigenvector (e_1, e_2, e_3) are calculated, which is the solution of the eigenvector problem and eigenvalues of the matrix A

$$e_1 = \frac{a_{23} - a_{32}}{\sqrt{4 - (1 - a_{11} - a_{22} - a_{33})}},$$

$$e_2 = \frac{a_{31} - a_{13}}{\sqrt{4 - (1 - a_{11} - a_{22} - a_{33})}},$$

$$e_3 = \frac{a_{12} - a_{21}}{\sqrt{4 - (1 - a_{11} - a_{22} - a_{33})}}.$$

The eigenvector $e = (e_1, e_2, e_3)$ found as a result of multiplication by the rotation matrix should remain unchanged $A e = e$, which can be seen in the last two lines of the ‘[Rotation Matrix Check](#)’ dialog:

Eigenvector of matrix A: $e = (0.052518, -0.058722, 0.258164)$

Check, $A * e = (0.052517, -0.058722, 0.258162)$

As soon as the quaternion (3.34) $\Lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ begins to depend on time, the molecule “comes to life”, turn (3.37) pulls the molecule out of position (*init*) and puts it in a new position, which is determined by the quaternion value and the corresponding value of the rotation matrix.

3.7 Dynamics of rotation

The basic equation describing the rotational dynamics of a solid body is the equation of momentum conservation

$$\frac{dL}{dt} = M, \quad (3.38)$$

where, L is the moment of impulse (kinetic moment), M is the moment of forces. The equation (3.38) is a direct consequence of Newton's second law [13, §56] and it is true in any inertial frame of reference.

The momentum L is the product of the inertia tensor I on the angular velocity vector ω .

$$L = I\omega. \quad (3.39)$$

All you need to do is differentiate the product of the inertia tensor by the instantaneous value of the angular velocity vector. In the projection on the coordinate axes of WCS, we have

$$\begin{aligned} L_x &= I_{xx}\omega_x + I_{xy}\omega_y + I_{xz}\omega_z \\ L_y &= I_{xy}\omega_x + I_{yy}\omega_y + I_{yz}\omega_z \\ L_z &= I_{xz}\omega_x + I_{yz}\omega_y + I_{zz}\omega_z \end{aligned} \quad (3.40)$$

However, in such a record, not only the angular velocity components depend on time, but also the components of the inertia tensor. The standard approach is to describe the rotation in a coordinate system rigidly connected to the rotating body - the coordinate system of the object. In this case, the inertia tensor is a constant value and taken out as a derivative.

Let the coordinate system rigidly connected to the rotating body at each moment of time is transformed from the world coordinate system (X_1, Y_1, Z_1) to the coordinate system (X_3, Y_3, Z_3) by a sequence of rotations to the Euler angles (Fig. 3.5). Find the angular velocities relative to the axes (X_3, Y_3, Z_3) , considering the known angular velocities $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$.

If $\dot{\alpha}$ - the angular velocity at which the coordinate system rotates (X_1, Y_1, Z_1) (the first rotation in Fig. 3.5), the angular velocity must be transformed because the coordinate system is subjected to two more rotations. The problem of conversion of angular velocities is solved sequentially for each of the rotations at an infinitely small angle in [14, §382]. As a result, the kinematic Euler equations are obtained

$$\begin{aligned}
\omega_{X3} &= \dot{\alpha} \sin \gamma \sin \beta + \dot{\beta} \cos \gamma \\
\omega_{Y3} &= \dot{\alpha} \cos \gamma \sin \beta - \dot{\beta} \sin \gamma \\
\omega_{Z3} &= \dot{\alpha} \cos \beta + \dot{\gamma}
\end{aligned} \tag{3.41}$$

Dynamic Euler equations are derived from the law of conservation of momentum (3.38) [13, §89] and recorded for angular velocities and moments of forces in the coordinate system (X_3, Y_3, Z_3) .

$$\begin{aligned}
I_{X3} \frac{d\omega_{X3}}{dt} + \omega_{Y3} \omega_{Z3} (I_{Z3} - I_{Y3}) &= M_{X3} \\
I_{Y3} \frac{d\omega_{Y3}}{dt} + \omega_{Z3} \omega_{X3} (I_{X3} - I_{Z3}) &= M_{Y3} \\
I_{Z3} \frac{d\omega_{Z3}}{dt} + \omega_{Y3} \omega_{X3} (I_{Y3} - I_{X3}) &= M_{Z3}
\end{aligned} \tag{3.42}$$

The kinematic Euler equations (3.41) are not obligate. The dynamic Euler equations relate the projections of angular velocities on the axis of the OCS and explicitly do not require knowledge of the current value of the angles. The current angle values may or may not be included in the moment expressions. In any case, if we want to consistently draw the position of the body, we need to recalculate the projections of angular velocities in the increment of angles, for this we can use the kinematic Euler equations resolved with respect to derivatives [15]

$$\begin{aligned}
\dot{\alpha} &= \frac{1}{\sin(\beta)} (\omega_{X3} \cdot \sin(\gamma) + \omega_{Y3} \cdot \cos(\gamma)) \\
\dot{\beta} &= \omega_{X3} \cdot \cos(\gamma) - \omega_{Y3} \cdot \sin(\gamma) \\
\dot{\gamma} &= \omega_{Z3} - (\omega_{X3} \cdot \sin(\gamma) + \omega_{Y3} \cdot \cos(\gamma)) \operatorname{ctg}(\beta)
\end{aligned} \tag{3.43}$$

In the same monograph [15] talk about especially with small angles of nutation β . The integration of these equations leads to a completely non-physical behavior, at low angular velocities, the molecule, being with a small angle β , suddenly tumbles 180°

The problem of solving the kinematic Euler equations are devoted specific articles [16, 17]. A productive approach to the solution of the kinematic Euler equations is the transition to the *Rodrigo-Hamilton* variables, which can be considered as a replacement of variables, leading to the linearization of the equations (3.41), as a result, the kinematic equations acquire the following form [11]

$$\begin{aligned}
\dot{\lambda}_0 &= -\frac{1}{2} (\omega_{X3} \lambda_1 + \omega_{Y3} \lambda_2 + \omega_{Z3} \lambda_3) \\
\dot{\lambda}_1 &= \frac{1}{2} (\omega_{X3} \lambda_0 + \omega_{Z3} \lambda_2 - \omega_{Y3} \lambda_3) \\
\dot{\lambda}_2 &= \frac{1}{2} (\omega_{Y3} \lambda_0 + \omega_{X3} \lambda_3 - \omega_{Z3} \lambda_1) \\
\dot{\lambda}_3 &= \frac{1}{2} (\omega_{Z3} \lambda_0 + \omega_{Y3} \lambda_1 - \omega_{X3} \lambda_2)
\end{aligned} \tag{3.44}$$

Quaternion is an absolute synonym of the rotation matrix, since there is a one-to-one transformation (3.34), (3.35) between them.

In §3.6 we came to the conclusion that it is advisable to store the molecule in some “storage” where we put it, performing a rotation defined by the matrix of the guide cosines A . Next, it is

necessary to pull the molecule out of storage and place it in its former place in the WCS, this can be done using the A^T matrix. For such a matrix it is possible to construct a quaternion through transformation (3.34). The resulting quaternion we set as the initial value for the kinetic equation. Then, at time $t = 0$, the quaternion and its corresponding rotation matrix will provide the necessary rotation from the “storage” to the initial position in the WCS.

Now, it is time to solve the *Euler* equations (3.42), which will give us the values of angular velocities relative to the axes of the OCS (X_3, Y_3, Z_3). Substitute angular velocities into kinetic equations (3.44) and obtain the quaternion increment due to the rotation for the time dt . The new value of the quaternion gives us the rotation matrix (3.35), which will take a molecule from the repository to the new location.

Unfortunately, that's not all. To solve the kinetic equation (3.42) it is necessary to calculate the forces, and to organize the correct transformation of the moments of forces from WCS to OCS.

Since the atomic coordinates and force fields are described in WCS, forces and moments are considered easy as described in §3.4. For the solution of dynamic *Euler* equations need only the moments of the forces calculated in WCS moment of the force vector is converted to OCS using the matrix guides of the cosines of A (3.35).

We check the dimensions for the dynamic *Euler* equations (3.42). Returning to §3.1, we will, as before, measure the angular velocity in radians for 10 femtoseconds and rewriting the transformation (3.6) for the ratio of the moment of force to the moment of inertia, we obtain

$$\omega_{\left(\frac{rad}{s}\right)} = 10^{14} \cdot \omega_{\left(\frac{rad}{10fs}\right)}, \quad \frac{d\omega_{\left(\frac{rad}{s}\right)}}{dt_{(s)}} = \frac{10^{14} \cdot d\omega_{\left(\frac{rad}{10fs}\right)}}{d(t_{(10fs)} \times 10^{-14})_{(s)}} = \frac{d\omega_{\left(\frac{rad}{10fs}\right)}}{dt_{(10fs)}} \times 10^{28},$$

$$\frac{M_{(N \times m)}}{I_{(Kg \times m^2)}} = \frac{\left(\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} \times 1,6021 \times 10^{-9} \right)_{(N)} \times (x_{(A)} \times 10^{-10})_{(m)}}{\left(m_{(amu)} \times 1,6597 \times 10^{-27} \right)_{(Kg)} \times (x_{(A)} \times 10^{-10})_{(m)}^2} = \frac{\Delta U_{(eV)} x_{(A)}}{\Delta x_{(A)} m_{(amu)} x_{(A)}^2} \times 0,9653 \times 10^{28}. \quad (3.45)$$

Multiplier 10^{28} is reduced, and the coefficient 0.9653 remained before the ratio of the moment of force to the moment of inertia.

$$\frac{d\omega_{X3}}{dt} = 0,9653 \times \frac{M_{X3}}{I_{X3}} - \frac{(I_{Z3} - I_{Y3})}{I_{X3}} \omega_{Y3} \omega_{Z3}. \quad (3.46)$$

The last equation shows the total moment of the M_{X3} forces acting on the molecule in the OCS. The coordinates of all atoms are stored in the WCS, the moment of force relative to the center of mass of the molecule is calculated by the formula (3.17), performing the summation of all atoms, we obtain

$$\vec{M} = \sum_{i \neq j} [(\vec{r}_i - \vec{r}_0) \times \vec{F}_{i,j}], \quad (3.47)$$

r_i - the radius-vector of the i -th atom, r_0 is the radius-vector of center of mass of the molecules, $F_{i,j}$ is the force acting on atom i from atom j . The obtained moment of forces vector M wrote on the projection on the WCS orts. Let's find the vector M projections on the OCS orts.

The OCS orts, are vectors (a_{11}, a_{12}, a_{13}) , (a_{21}, a_{22}, a_{23}) , (a_{31}, a_{32}, a_{33}) , let the desired vector, recorded in the projection on the OCS orts, has coordinates (M_{X3}, M_{Y3}, M_{Z3}) , then

$$(a_{11}, a_{12}, a_{13})M_{X3} + (a_{21}, a_{22}, a_{23})M_{Y3} + (a_{31}, a_{32}, a_{33})M_{Z3} = (M_X, M_Y, M_Z), \quad (3.48)$$

or matrix form

$$\begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} M_{X3} \\ M_{Y3} \\ M_{Z3} \end{pmatrix}. \quad (3.49)$$

Multiplying on the left by a matrix A , we get

$$A \cdot M = A \cdot A^T \cdot M_3, \quad (3.50)$$

or

$$M_3 = A \cdot M. \quad (3.51)$$

We write a sequence of calculations that perform rotation transformations.

1	$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, X, X_0$	The source data is generated when the molecules are created in the editor or when the previously saved data is loaded;	
2	$\vec{X}^{(init)} = A(\vec{X} - \vec{X}_0)$	Remember the coordinates of atoms in the "storage"; A - matrix of guiding cosines;	
3	$\lambda_0^0 = \frac{1}{2} \sqrt{1 + a_{11} + a_{22} + a_{33}}$ $\lambda_1^0 = \frac{a_{32} - a_{23}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}}$ $\lambda_2^0 = \frac{a_{13} - a_{31}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}}$ $\lambda_3^0 = \frac{a_{21} - a_{12}}{2\sqrt{1 + a_{11} + a_{22} + a_{33}}}$	Perform the initialization of the <i>Rodrigo-Hamilton</i> parameters, which corresponds to the current orientation of the molecules and is completely determined by the matrix A ;	(3.52)
4	$\vec{M} = \sum_{i,j} [\vec{r}_i \times \vec{F}_{i,j}]$	Calculate the forces acting on each atom and the moment of forces acting on the molecule in WCS;	
5	$\vec{M}_3 = A \cdot \vec{M}$	Find the projection of the moment vector of forces on the WCS ors;	
6	$\frac{d\omega_{X3}^r}{dt} = 0,9653 \times \frac{M_{X3}}{I_{X3}} - \frac{(I_{Z3} - I_{Y3})}{I_{X3}} \omega_{Y3}^0 \omega_{Z3}^0$ $\frac{d\omega_{Y3}^r}{dt} = 0,9653 \times \frac{M_{Y3}}{I_{Y3}} - \frac{(I_{X3} - I_{Z3})}{I_{Y3}} \omega_{Z3}^0 \omega_{X3}^0$ $\frac{d\omega_{Z3}^r}{dt} = 0,9653 \times \frac{M_{Z3}}{I_{Z3}} - \frac{(I_{Y3} - I_{X3})}{I_{Z3}} \omega_{Y3}^0 \omega_{X3}^0$	Solve the dynamic <i>Euler</i> equations by numerical method;	

7	$\omega_{x3}^0 = \omega_{x3}^\tau \quad \omega_{y3}^0 = \omega_{y3}^\tau \quad \omega_{z3}^0 = \omega_{z3}^\tau$	Remember the new initial conditions for the next step;
8	$\dot{\lambda}_0^\tau = -\frac{1}{2}(\omega_{x3}\lambda_1^0 + \omega_{y3}\lambda_2^0 + \omega_{z3}\lambda_3^0)$ $\dot{\lambda}_1^\tau = \frac{1}{2}(\omega_{x3}\lambda_0^0 + \omega_{z3}\lambda_2^0 - \omega_{y3}\lambda_3^0)$ $\dot{\lambda}_2^\tau = \frac{1}{2}(\omega_{y3}\lambda_0^0 + \omega_{x3}\lambda_3^0 - \omega_{z3}\lambda_1^0)$ $\dot{\lambda}_3^\tau = \frac{1}{2}(\omega_{z3}\lambda_0^0 + \omega_{y3}\lambda_1^0 - \omega_{x3}\lambda_2^0)$	Solve kinematic Euler equations by numerical method;
9	$\lambda_0^0 = \lambda_0^\tau \quad \lambda_1^0 = \lambda_1^\tau \quad \lambda_2^0 = \lambda_2^\tau \quad \lambda_3^0 = \lambda_3^\tau$	Remember the new initial conditions for the next step;
10	$A = \begin{pmatrix} 2(\lambda_0^2 + \lambda_1^2) - 1 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 2(\lambda_0^2 + \lambda_2^2) - 1 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 2(\lambda_0^2 + \lambda_3^2) - 1 \end{pmatrix}$	Calculate the rotation matrix (the matrix of guiding cosines);
11	$\vec{X} = A^T \vec{X}^{(init)} + \vec{X}_0$	Perform the rotation of all atoms for each molecule.
12	Back to position 3	

3.8 Runge-Kutta methods

Program uses two fourth-order *Runge-Kutta* methods for first-order differential equations (systems (3.42) and (3.44)) and for second-order differential equations (3.4).

For first-order differential equations, the *Runge-Kutta* method has the form

$$\begin{aligned}
 k_1 &= F(t, x) \\
 k_2 &= F\left(t + \frac{h}{2}, x + \frac{h}{2}k_1\right) \\
 k_3 &= F\left(t + \frac{h}{2}, x + \frac{h}{2}k_2\right) \\
 k_4 &= F(t + h, x + hk_3) \\
 x(t + h) &= x(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),
 \end{aligned} \tag{3.53}$$

where $F(t, x)$ – is the right part of the differential equation solved with respect to the derivative.

To testing the *Runge-Kutta* Method for first-order differential equations performed for the calculation scheme (3.53), which performs free rotation of a water molecule with the following parameters: the main values of the inertia tensor $I_X=1.810$, $I_Y=0.666$, $I_Z=1.144$, the initial velocities $\omega_X=0.0$, $\omega_Y=0.00001$, $\omega_Z=0.1$, the initial values of the quaternion are $\lambda_0=0.733663$, $\lambda_1=-0.087161$, $\lambda_2=-0.035699$, $\lambda_3=-0.672954$. The solution was built in Maple and MolRun; the comparison result is shown in Fig. 3.6, where, the solid curve shows the solution obtained by Maple, the dots show the solution obtained by MolRun.

The obtained solutions reproduce the dynamic effect known as the “*Dzhanibekov effect*” - periodic somersault, of a body with an asymmetric inertia tensor. The first figure shows the dynamics of the quaternion parameter (λ_0), which reflects the free rotation of the molecule. The second figure shows the dynamics of angular velocities, it can be seen that despite the very small initial value of the angular velocity $\omega_Y = 10^{-5}$, this perturbation is enough to the angular velocity ω_Z at a certain moment reversed the sign.

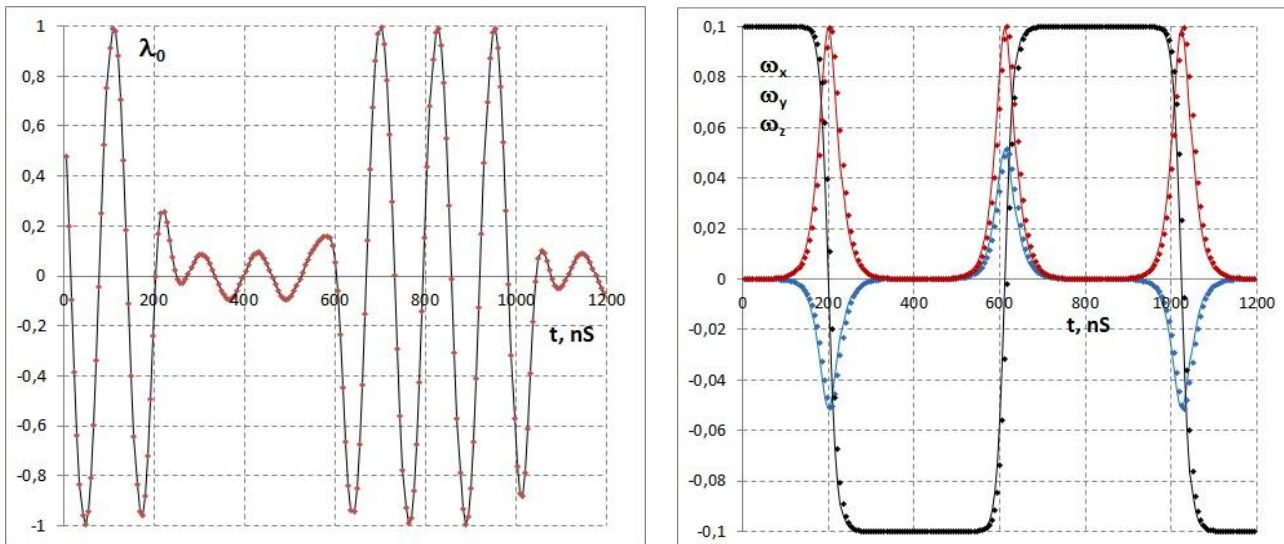


Fig. 3.6 Simulation results comparison of water molecules free rotation, the solid curve is the solution obtained by the program MolRun, point – solution obtained by the Maple program.

Dzhanibekov effect simulating hides an important feature - in the case of free rotation, the right-hand sides of $F(t, x)$ are independent of angles. In the general case, the moment of force depends on the body orientation, and when the molecules collide, the moment of force very much depends on the angles, while the calculation of the forces at the half-step of the *Runge-Kutta* method becomes very important. For this reason, it is necessary to verify the numerical implementation of the method for the case of a nonzero moment of forces. The result of comparing the calculations in the MolRun program and the modeling in the Maple package for the case of a constant moment of force $M_x = 10^{-5}$ $M_y = 0$, $M_z = 0$, is shown in Fig. 3.7.

Runge-Kutta method testing for second-order differential equations was performed for the following model. Let there be two Ne atoms, with the parameters of the interaction potential $\varepsilon = 0.03067$, $\alpha = 5.1$, $r_m = 3.086$, $m = 20.183$. To make the atoms move, we place them at the coordinates $(0,0,-2)$ and $(0,0,2)$, so the distance between atoms is 4 angstroms and assign fictitious charges $+0.1$ and -0.1 C. Particles will begin to attract each other until they “collide” with the interaction potentials.

Due to the symmetry of the problem, we will consider the coordinate of one atom $z = 2.0$. The Coulomb interaction acts in the direction of decreasing the coordinate of this atom, which means that the minus sign should be assigned to the force, the van der Waals force at a distance of 4 angstroms also acts in the direction of decreasing the coordinate, which means it also enters into the equation of motion with a minus sign, total, we have the equation of motion

$$\ddot{z} = \frac{0.9653}{m} (F_1(z) + F_2(z))$$

$$F_1(z) = -\frac{14.4073|q_1q_2|}{(2z)^2}$$

$$F_2(z) = -\frac{1}{r_m} \frac{6\varepsilon}{1 - \frac{6}{\alpha}} \left(-e^{\alpha \left(1 - \frac{2z}{r_m}\right)} + \left(\frac{r_m}{2z}\right)^7 \right)$$
(3.54)

The results are obtained using the following formulas that give a solution to a second-order differential equation

$$k_1 = F(x_n)h$$

$$k_2 = F\left(x_n + v_n \frac{h}{2}\right)h$$

$$k_3 = F\left(x_n + v_n \frac{h}{2} + k_1 \frac{h}{4}\right)h$$

$$k_4 = F\left(x_n + v_n h + k_2 \frac{h}{2}\right)h$$

$$\Delta x_n = v_n h + \frac{1}{6}(k_1 + k_2 + k_3)h$$

$$\Delta v_n = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
(3.55)

The variables x_n , v_n should be understood as vectors in the WCS; F is the total force applied to the center of mass of the molecule.

The calculation results in Maple and MolRun are shown in Fig. 3.8.

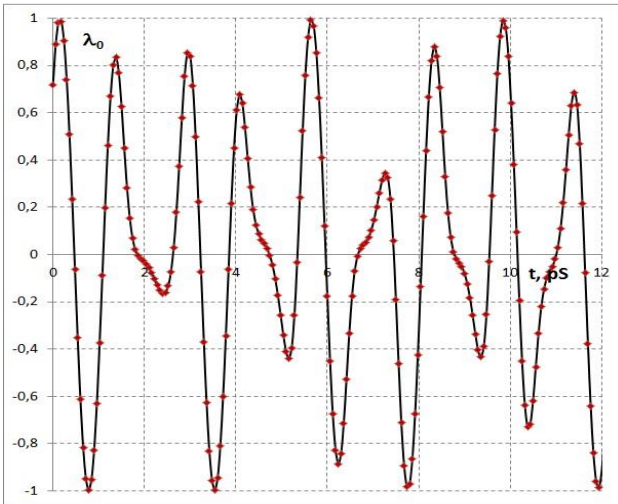


Fig. 3.7 The quaternion component λ_0 under the action of an external moment $M_x = 0.00001$

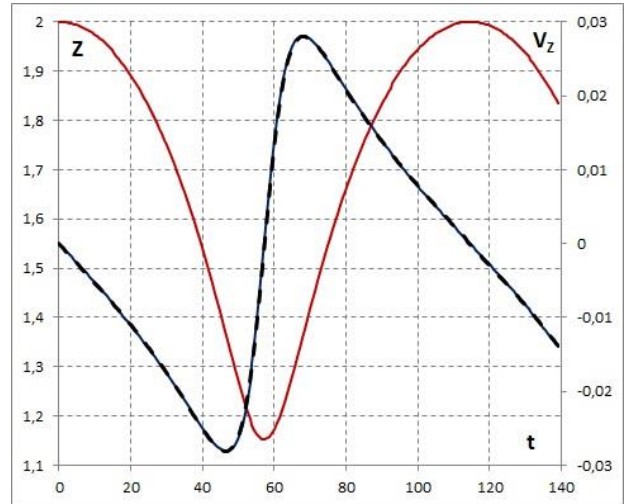


Fig. 3.8 The distance between the atoms (red line), speed (blue line).

3.9 The energy calculation

Energies of atomic scale are represented in a variety of units.

J/mol is a joule divided by the Avogadro number, that is, just a unit of energy transferred to the atomic scale. Convert this unit to electron volts

$$1 \frac{J}{mol} = \frac{(6.24 \times 10^{18})_{(eV)}}{6 \times 10^{23}} = 1.04 \times 10^{-5} eV.$$

A similar situation with the unit cal/mol, translate it into electron volts

$$1 \frac{cal}{mol} = \frac{(4.184 \times 6.24 \times 10^{18})_{(eV)}}{6 \times 10^{23}} = 4.35 \times 10^{-5} eV.$$

The atomic unit of energy (atomic unit), it is Hartree, is 27.211 eV.

To transfer the energy unit to the atomic scale, you can choose another divisor other than the Avogadro number, you can use the Boltzmann constant as such a divider, then the energy scale will be expressed in Kelvin. However, 1 Joule is too large a macroscopic value, therefore, we can express it in units of temperature 1 eV. First we translate the Boltzmann constant into units eV/K

$$1.38 \times 10^{-23} \times 0.625 \times 10^{19} = 8,62 \times 10^{-5} eV / K.$$

Now, divide 1 eV by the Boltzmann constant, expressed in eV/K

$$\left[\frac{\varepsilon}{k} \right] = \frac{1_{(eV)}}{(8.625 \times 10^{-5})_{(eV/K)}} = 1.159 \times 10^4 K.$$

To exclude doubts whether the factor depending on the number of degrees of freedom was lost during the conversion, we turn to [18, p.5] where it is indicated that the monograph uses the equality $1 eV = 1.1 \times 10^4 K$.

The program calculates the value of kinetic energy

$$E_k = \frac{1}{2} \sum m_{(amu)} \times (v_{x(A/10fs)}^2 + v_{y(A/10fs)}^2 + v_{z(A/10fs)}^2), \quad (3.56)$$

which can be converted to joules

$$(m_{(amu)} \times 1,6597 \times 10^{-27})_{(Kg)} \times (v_{(A/10fs)} \times 10^4)_{(m/s)}^2 = (m_{(amu)} \times v_{(A/10fs)}^2 \times 1,6597 \times 10^{-19})_{(J)}. \quad (3.57)$$

If we recall that $1 eV = 1,602 \times 10^{-19} J$, then we get

$$(m_{(amu)} \times v_{(A/10fs)}^2 \times 1,036)_{(eV)}. \quad (3.58)$$

We do the same with the kinetic energy of rotational degrees of freedom

$$E_r = \frac{1}{2} \sum (I_{xx(amuxA^2)} \times \omega_{\alpha(rad/10fs)}^2 + I_{yy(amuxA^2)} \times \omega_{\beta(rad/10fs)}^2 + I_{zz(amuxA^2)} \times \omega_{\gamma(rad/10fs)}^2). \quad (3.59)$$

Let us reduce the dimensions of the moment of inertia and angular velocity to the SI units

$$\begin{aligned} (m_{(amu)} \times 1.6597 \times 10^{-27})_{(Kg)} \times (x_{(A)} \times 10^{-10})_{(m)}^2 \times (\omega_{(rad/10fs)} \times 10^{14})_{(rad/S)}^2 = \\ = (I_{(amuxA^2)} \times \omega_{(rad/10fs)}^2 \times 1.6597 \times 10^{-19})_{(J)}. \end{aligned} \quad (3.60)$$

We got exactly the same conversion factor as for kinetic energy (3.57), which means that we hadn't been mistaken.

The potential energy of particles systems is considered as the sum of the potentials of all pair interactions.

$$E_p = \sum_{i < j} U(\mathbf{r}_i - \mathbf{r}_j). \quad (3.61)$$

The potential energy $U(r)$ is the sum of the interaction energy of a pair of atoms (2.16) and the energy of the Coulomb interaction if charges are assigned to atoms. There are no questions with formula (2.16); it gives the value of the interaction energy in units of eV.

For a pair of charges, the force was calculated in § 3.2 as

$$\frac{\Delta U_{(eV)}}{\Delta x_{(A)}} = \frac{q_{(e)}q_{(e)}}{x_{(A)}^2} \times 14,3993. \quad (3.62)$$

Then

$$U_{(eV)}(x) = 14,3993 \int_{\infty}^x \frac{q_{(e)}q_{(e)}}{x_{(A)}^2} dx = U_0 + 14,3993 \frac{q_{(e)}q_{(e)}}{x_{(A)}}. \quad (3.63)$$

3.10 The temperature calculation

The basic expression for the relationship between temperature and energy is the ratio [8]

$$\frac{1}{2} k_b T = \langle E_l \rangle. \quad (3.64)$$

By energy $\langle E_l \rangle$ we mean kinetic energy per one degree of freedom of one particle. If, in addition to translational degrees of freedom, rotational degrees of freedom are present, the total kinetic energy of N particles will be evenly distributed over all available three or six degrees of freedom

$$\langle E_k + E_r \rangle = \frac{3 \text{ or } 6}{2} N k_b T. \quad (3.65)$$

If the right-hand side of (3.65) can be left as is in the program code, in the absence of rotational degrees of freedom the corresponding contribution to energy will be zero, then the choice of the coefficient (3 or 6) is assigned to the user, the check “Rotation Energy (6N)” on the Physics tab Workspace windows.

Expressions (3.57), (3.60) give a representation of the kinetic energy in joules, whence we obtain the expression for the temperature in Kelvin.

$$T_{(K)} = \frac{2 \left(\langle E_k + E_r \rangle_{(eV)} \times 1.602 \times 10^{-19} \right)_{(J)}}{(3 \text{ or } 6) N k_b} = \frac{\langle E_k + E_r \rangle_{(eV)}}{(3 \text{ or } 6) N} \times 23209.03605. \quad (3.66)$$

This is the “kinetic” temperature [19, p. 198], which, generally speaking, was defined in kinetic theory for the state of thermodynamic equilibrium.

3.11 The density calculation

For the gas phase, the density is calculated for a cube with an edge CL, if a surface bounding a molecule or cluster is constructed, the density is calculated for a volume bounded by the surface.

$$\rho_{(kg/m^3)} = \frac{m_{(kg)}}{V_{(m^3)}} = \frac{(m_{(amu)} \times 1,6597 \times 10^{-27})_{(kg)}}{(CL_{(A)} \times 10^{-10})^3_{(m)}} = \left(\frac{m_{(amu)}}{CL_{(A)}^3} \times 1659,7 \right)_{(kg/m^3)} \quad (3.67)$$

or

$$\frac{(10^3 \times m_{(g)})_{(kg)}}{(10^6 \times V_{(cm^3)})_{(m^3)}} = \frac{m_{(g)}}{V_{(cm^3)}} \times 10^{-3}. \quad (3.68)$$

3.12 The dipole moment calculation

The dipole moment of the molecule is not involved in the solution of the equations of motion, so as not to waste processor time, the rotation of the vector of the dipole moment is not recounted at each step, but is calculated at time intervals ‘[Save To Analysis](#)’. The operation of rotating the vector by an angle, which is determined by the composition of the turns, is much more complicated than directly calculating the vector of dipole moment, therefore, if ‘[Dipole Moment Analysis](#)’ is checked, the program calculates the dipole moment of each molecule using the classical formula

$$\vec{\mu}_{(C \times m)} = \sum_i \vec{r}_i q_i = \sum \frac{(r_{(A)} \times 10^{-10})_{(m)} (q_{(e)} \times 1,6021 \times 10^{-19})_{(C)}}{3,34 \times 10^{-30}} = \sum (r_{(A)} q_{(e)} \times 4,79)_{(D)}, \quad (3.69)$$

The result of the calculations is presented in Debye.

3.13 The volume calculation

Suppose there is a distribution of atoms, you need to cover it with an elastic surface and calculate the internal volume bounded by this surface. Let the problem of finding atoms located on the outer layer be solved.

Let assume the distribution center of mass located inside the volume bounded by the outer layer of atoms. Let’s find the distribution center of mass, take it as the beginning of a spherical coordinate system and transform the outer layer atoms Cartesian coordinates to spherical ones.

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos\left(\frac{z}{r}\right) \\ \varphi &= \text{atan}_2\left(\frac{y}{x}\right) \end{aligned} \quad (3.70)$$

Here the standard function $\text{atan}_2(y, x)$ is used, which works taking into account the signs of the arguments and returns the angle in the range $-\pi \dots \pi$. The inverse transformation from a spherical system to a Cartesian one works correctly in the entire range of angles and has the form

$$\begin{aligned} z &= r \cos(\theta) \\ l &= \sqrt{r^2 - z^2} \\ x &= l \cos(\varphi) \\ y &= l \sin(\varphi) \end{aligned} \quad (3.71)$$

It would be tempting to construct a surface as a sum of generating functions F that provide the necessary distance from the distribution center to the atom located point. As we move away from the point where the atom located, the function decreases

$$R(\theta, \varphi) = \sum A_i F(\theta, \theta_i, \varphi, \varphi_i). \quad (3.72)$$

The function R must satisfy the following conditions. At the atom location angular coordinates (θ_i, φ_i) , the function returns the specified distance from the origin

$$r_i = R(\theta_i, \varphi_i). \quad (3.73)$$

Since the coordinate φ must define a closed curve, it is necessary to additionally require

$$R(\theta_i, -\pi) = R(\theta_i, \pi). \quad (3.74)$$

The first condition leads to a system of linear algebraic equations with respect to unknown coefficients A_i

$$\begin{pmatrix} r_1 \\ \dots \\ r_n \end{pmatrix} = \begin{pmatrix} F_{1,1} & \dots & F_{1,n} \\ \dots & \dots & \dots \\ F_{n,1} & \dots & F_{n,n} \end{pmatrix} \cdot \begin{pmatrix} A_1 \\ \dots \\ A_n \end{pmatrix}. \quad (3.75)$$

The condition (3.74) in the model (3.72) cannot be fulfilled, which leads to a gap at the surface closure, but does not prevent an estimate of the volume. Since, by the condition, the generating functions decrease as they move away from the location of the atom, we can expect that the matrix F will contain many values close to zero. To solve System of linear equations with such matrices, the Gauss method is used with the selection of the largest element [20], which is implemented in the program.

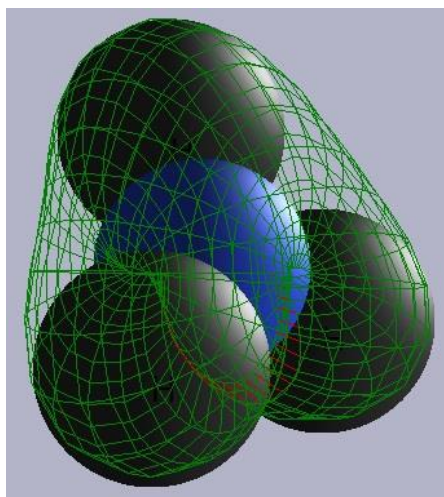
Functions of the following type were studied as generating functions:

$$R = \sum \frac{A_i}{1 + \sqrt{(\theta - \theta_i)^2 + (\varphi - \varphi_i)^2}}, \quad (3.76)$$

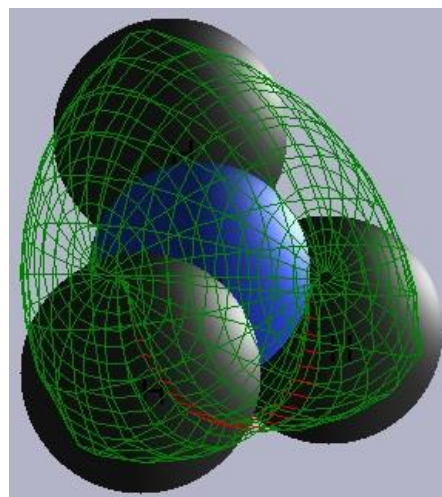
$$R = \sum \frac{A_i}{1 + (\theta - \theta_i)^2 + (\varphi - \varphi_i)^2}. \quad (3.77)$$

The selection of the generating function (3.76) or (3.77) is made by the ‘[Workspace\Volume\Function check mark](#)’. The surfaces constructed by generating functions (3.76), (3.77) are shown in Fig. 3.9-3.11, where V_0 denotes a volume equal to the sum of the volumes of spheres given by atomic radii, which is taken as 100%, the volume limited by the constructed surfaces is denoted as V_1 and V_2 for different models.

$$R = \sum \frac{A_i}{1 + (\theta - \theta_i)^2 + (\varphi - \varphi_i)^2} \qquad R = \sum \frac{A_i}{1 + \sqrt{(\theta - \theta_i)^2 + (\varphi - \varphi_i)^2}}$$

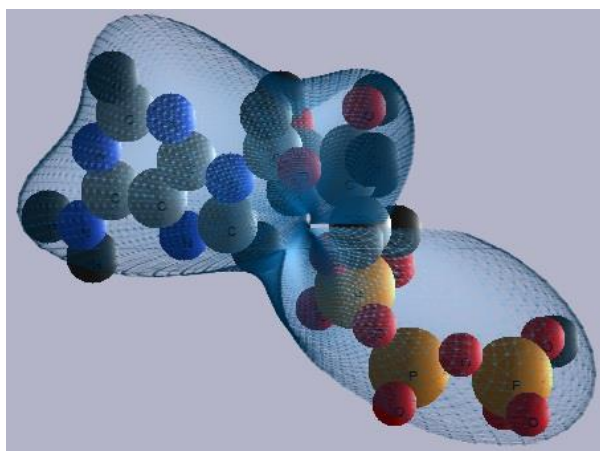


$$V_0=7,9 \text{ \AA}^3; V_2=9,4 \text{ \AA}^3 (+18\%)$$

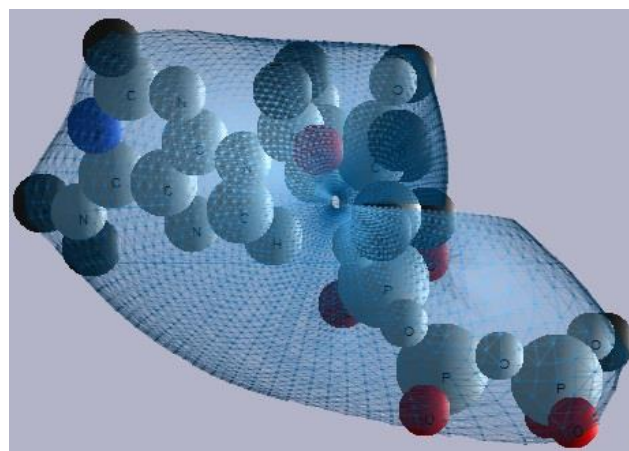


$$V_0=7,9 \text{ \AA}^3; V_1=9,3 \text{ \AA}^3 (+17\%)$$

Fig. 3.9 Approximating surface for an ammonia molecule (a gap in the place of crosslinking is highlighted in red $\varphi=-\pi, \pi$). A grid of 25x25 elements.

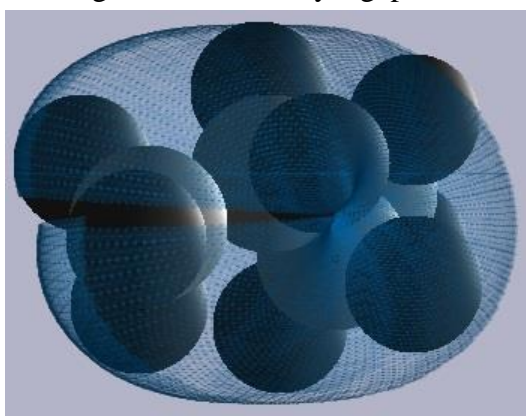


$$V_0=105,5 \text{ \AA}^3; V_2=490,9 \text{ \AA}^3 (+365\%)$$

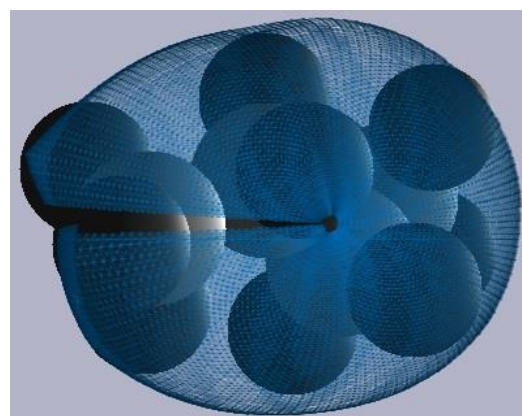


$$V_0=105,5 \text{ \AA}^3; V_1=565,4 \text{ \AA}^3 (+435\%)$$

Fig. 3.10 The approximating surface for the ATP molecule, a grid of 100x100 elements, the crosslinking area is shown by a gap.



$$V_0=33,3 \text{ \AA}^3; V_2=69,5 \text{ \AA}^3 (+109\%)$$



$$V_0=33,3 \text{ \AA}^3; V_1=69,9 \text{ \AA}^3 (+110\%)$$

Fig. 3.11 The approximating surface for the butane molecule, a grid of 100x100 elements, the crosslinking area is shown by a gap.

To calculate the volume, the same grid is used as for surface visualization. The volume bounded by the surface (3.72) is calculated as the sum of trapezoids with vertices in the center of mass of the distribution of atoms

$$V = \sum_{\Theta} \sum_{\varphi} \left| \begin{array}{ccc} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{array} \right|, \quad (3.78)$$

where (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) are the Cartesian coordinates of the triangle, the point $(0, 0, 0)$ is the center of mass, straight brackets denote the modulus of the determinant.

To locate the atoms forming the outer layer of a molecule or a cluster following algorithm are used. The procedure takes a list of atoms to be processed, calculates the center of mass coordinates, taking atoms mass as single. Next, the angular coordinates of the pair of tested atoms are calculated (θ_1, φ_1) , (θ_2, φ_2) , and the angular dimensions of the tested atoms are calculated.

$$\alpha_1 = E_s \cdot \arctg\left(\frac{R_1}{r_1}\right), \quad \alpha_2 = E_s \cdot \arctg\left(\frac{R_2}{r_2}\right), \quad (3.79)$$

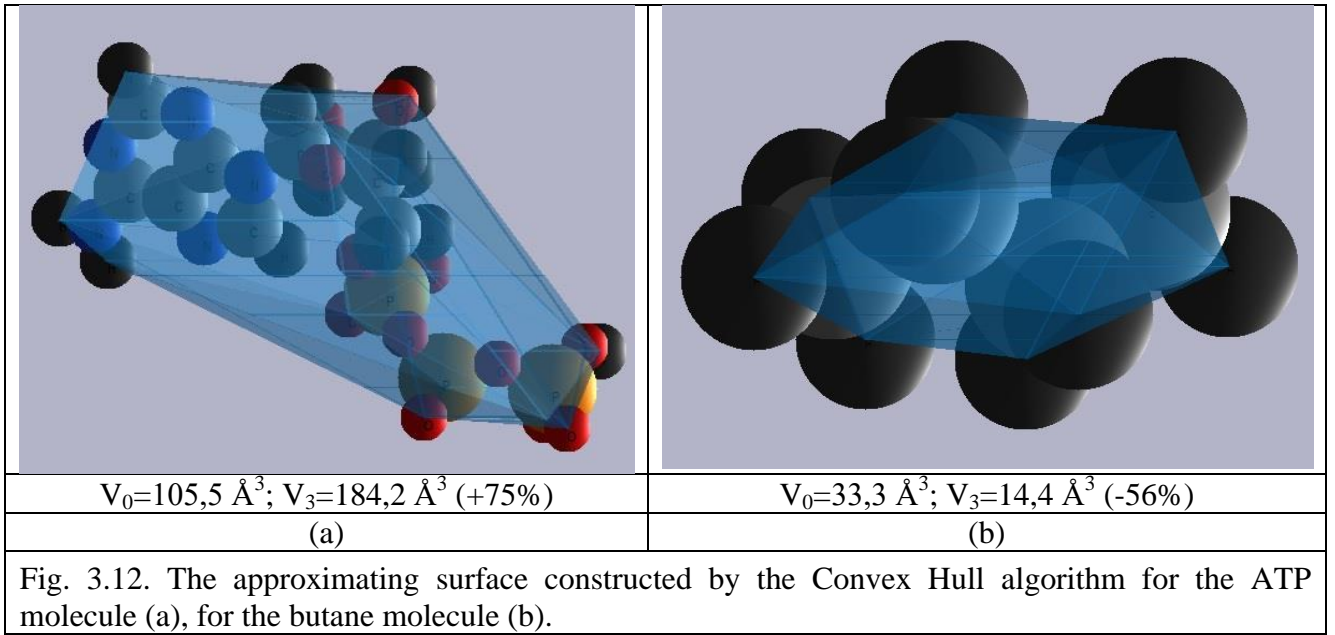
where R is the radius of the atom, r is the distance from the center of mass to the atom, E_s is the parameter that determines the sensitivity of the internal atom search procedure. The E_s parameter is editable on the [Workspace\Volume tab\Atom list\External atoms sensitivity](#) parameter. Next, the angular dimensions of a pair of atoms are compared. If the intervals of angular coordinates

$$\begin{aligned} & [\theta_1 - \alpha_1, \theta_1 + \alpha_1] \text{ и } [\theta_2 - \alpha_2, \theta_2 + \alpha_2] \\ & [\varphi_1 - \alpha_1, \varphi_1 + \alpha_1] \text{ и } [\varphi_2 - \alpha_2, \varphi_2 + \alpha_2] \end{aligned} \quad (3.80)$$

overlap at the corners θ , φ , then an atom with a smaller distance r is removed from the compared pair of atoms. In order for the result of the procedure to be visible on the screen, it is necessary to set the '[Workspace\Volume mark\Atom list\Fade](#)' internal atoms, then the deleted atoms will be displayed in white. This function is not reversible, since it replaces colors in the main data array of molecules.

A smooth surface looks nice, but it requires a lot of computing resources, so it is not suitable for tasks where the volume needs to be calculated continuously. There is a fast algorithm *Convex Hull* [20] that wraps a point cloud with a surface constructed from plane elements. This algorithm is implemented in the program and is enabled by the '[Workspace\Volume\Surface\Convex Hull](#)' check mark. In the three-dimensional case, the algorithm is cumbersome and has no relation to the of the simulated process's physics, so the description of the software implementation is not given here. For comparison, the volumes of molecules calculated using the Convex Hull algorithm are shown below. The volume bounded by the surface is calculated as the sum of the volumes of tetrahedra with a common vertex.

The Convex Hull algorithm implemented in the program was conceived to calculate the volume of a collective of molecules, therefore, the centers of atoms are chosen as surface constructed points, which leads to underestimated volume values if the volume of a single molecule is calculated.



3.14 The pressure calculation

To calculate the pressure, the *Clausius virial* Ω is used, which is described in detail in [22, 23],

$$\Omega = -\frac{1}{2} \left\langle \sum_{i=1}^N (x_i F_{xi} + y_i F_{yi} + z_i F_{zi}) \right\rangle \quad (3.81)$$

A certain (3.81) virial satisfies the equality

$$\langle E_k \rangle = \Omega. \quad (3.82)$$

Since the virial is defined through forces, it becomes possible to isolate pressure as the flow of a force field along the outer surface bounding the particle cluster. The force acting on the molecule is divided into the force acting on the outer surface and the forces acting on the other $N-1$ particles. Thus, the virial splits into three virials *total virial*, *external virial (wall virial)*, *internal virial*

$$\Omega^{tot} = \Omega^{ext} + \Omega^{int}. \quad (3.83)$$

Which ultimately leads to the formula [23, p.144, formula (8.26)]

$$PV = Nk_b T + \frac{1}{3} \left\langle \sum_i r_i F_i^{(int)} \right\rangle. \quad (3.84)$$

The virial has the dimension of energy, but is not as familiar a physical quantity as speed or energy. To deal with the sign of the virial term, we will write the virial sum for all particles through the sum of interacting pairs of particles. The easiest way to do this is by describing the interaction for three particles. There are only two forces acting on each of the particles

$$\sum_{k=1}^3 F_k r_k = (F_{12} + F_{13})r_1 + (F_{21} + F_{23})r_2 + (F_{31} + F_{32})r_3. \quad (3.85)$$

Group the members

$$F_{12}r_1 + F_{21}r_2 + F_{13}r_1 + F_{31}r_3 + F_{23}r_2 + F_{32}r_3, \quad (3.86)$$

using Newton's third law, for pairs of forces, we can write $F_{12} = -F_{21}$, where from

$$F_{12}(r_1 - r_2) + F_{13}(r_1 - r_3) + F_{23}(r_2 - r_3) = \sum_{i=1}^3 \sum_{i < j} (r_i - r_j) F_{ij}. \quad (3.87)$$

Returning to (3.84) we get

$$PV = Nk_B T + \frac{1}{3} \left\langle \sum_{i=1}^N \sum_{i < j} (r_i - r_j) F_{ij}^{(\text{int})} \right\rangle. \quad (3.88)$$

It remains to rewrite (3.88) in program units, denoting units of measurement in the lower index (see (3.1))

$$P_{\left(\frac{N}{m^2}\right)} \left(V_{(A^3)} \times 10^{-30} \right)_{(m^3)} = Nk_B T + \frac{1}{3} \left\langle \sum_{i=1}^N \sum_{i < j} \left(F_{(eV/A)} \times 1,6021 \times 10^{-9} \right)_{(N)} \left((r_{(A)} - r_{(A)}) \times 10^{-10} \right)_{(m)} \right\rangle \quad (3.89)$$

or

$$P_{\left(\frac{N}{m^2}\right)} = \frac{N \times 1,380\,649 \times 10^{-23} \times T_{(K)}}{\left(V_{(A^3)} \times 10^{-30} \right)_{(m^3)}} + \frac{1}{3} \left\langle \sum_{i=1}^N \sum_{i < j} F_{(eV/A)} (r_{(A)} - r_{(A)}) \right\rangle \times \frac{1,6021 \times 10^{11}}{V_{(A^3)}} \quad (3.90)$$

or

$$P_{(Pa)} = \frac{NT_{(K)}}{V_{(A^3)}} 1,380\,649 \times 10^7 + \frac{1}{3} \left\langle \sum_{i=1}^N \sum_{i < j} F_{(eV/A)} (r_{(A)} - r_{(A)}) \right\rangle \times \frac{1,6021 \times 10^{11}}{V_{(A^3)}} \quad (3.91)$$

Let's illustrate the contribution of the virial member to pressure by an elementary example. Consider the behavior of two Ne atoms with charges of different signs located on the same axis. At the zero moment of time, the particles are at rest, starting to move towards each other under the influence of Coulomb attraction. In this case, the virial has a positive value. Then, the repulsive forces exceed the attractive forces, at this point we have a maximum of kinetic energy and the virial becomes negative. After the collision, the particles fly apart until the forces of attraction stop them.

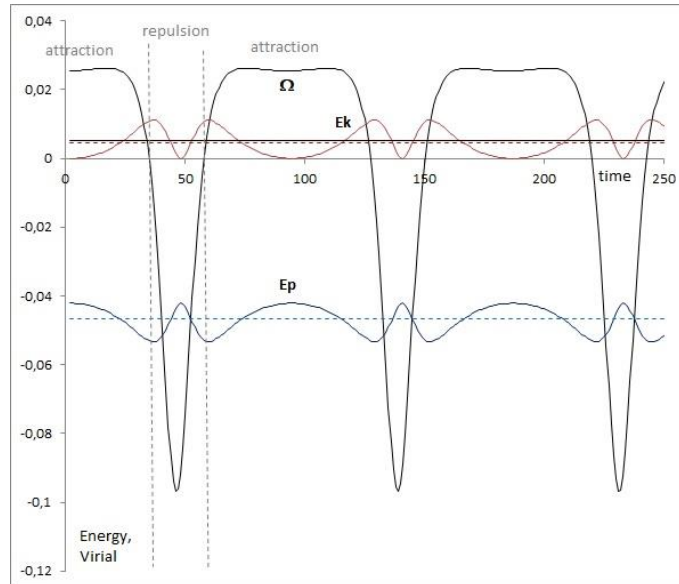


Fig. 3.13 Virial Ω , kinetic energy E_k , potential energy E_p for a pair of colliding Ne atoms with charges of different signs located on the same axis.

Fig. 3.13 shows a graph of kinetic E_k and potential E_p energy, their average values (dotted lines), which are calculated over the oscillation period, and the value of the *Clausius* virial entering (3.88).

The average value of the virial is shown by a black straight line, which practically coincides with

the red dotted line - the average kinetic energy, that is, the virial theorem (3.82) is fulfilled.

Fig. 3.13 also makes it clear that the time scale on which the instantaneous values of the virial must be calculated should be at least less than the time at which the velocity reversal occurs during the collision of particles.

Let's return to the formula (3.88), the value Ω shown in Fig. 3.11 and the virial sum included in (3.88) have different signs, which means that as long as the particles are mostly in the region of mutual attraction, the virial term will give a minus sign contribution to the total pressure (3.88), when the particles are mostly compressed to the 'repulsion' region, Fig. 3.13, the virial sum will give a plus sign contribution to the total pressure of the system.

4 Program Management

4.1 Screen projection

To display the three-dimensional world described in the world coordinate system, a procedure is required that displays the coordinates of the WCS on the coordinates of the screen plane. The most universal method is projected onto a plane having an arbitrary position in space. This method allows to freely fly in the space of WCSs and observe the picture from any point and from any angle.

Let the vector V_0 , describes the position of the observer, the vector V_1 describes the direction of view, the vectors V_2, V_3 , specify the unit vectors of the axes on which the plane of the screen projection is stretched. Q is the vector defining the point of the central projection, P is the coordinate of the point for which it is necessary to construct the central projection (Fig. 4.1). All vectors are given in the coordinates of the WCS (X, Y, Z).

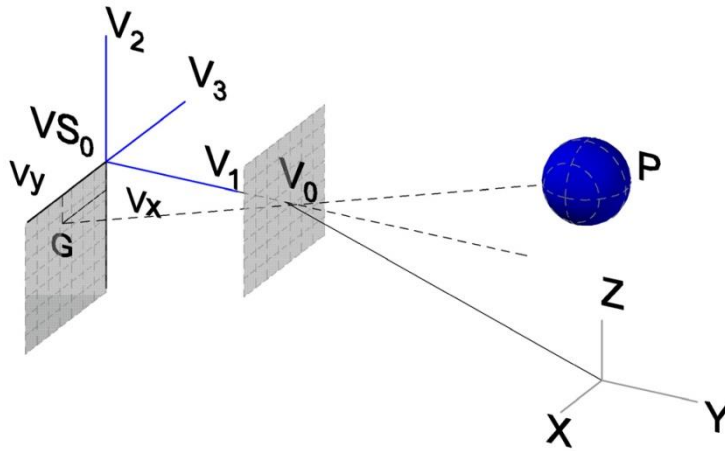


Fig. 4.1 World coordinate system (X, Y, Z) and the screen projection plane (V_2, V_3).

Let's start with the simplest, move the projection plane away from the point V_0 , where the observer is located. Step back to the distance h , in the direction of the vector $-V_1$

$$VS_0 = V_0 - V_1 \cdot h. \quad (4.1)$$

A straight line passing through the center of the projection will pierce the projection plane at point G . Let's construct the equation of the PG line in parametric form. To get the vector PG , you need to subtract the vector P from the vector G , multiplying this vector by the parameter, we get a set of points belonging to the line PG

$$G = V_0 + \tau \cdot (G - P). \quad (4.2)$$

Now let's build the projection plane. The equation of the plane can be obtained by requiring that the vector perpendicular to vectors V_2, V_3 has zero length

$$\begin{vmatrix} x - V_{0x} & y - V_{0y} & z - V_{0z} \\ V_{2x} & V_{2y} & V_{2z} \\ V_{3x} & V_{3y} & V_{3z} \end{vmatrix} = 0. \quad (4.3)$$

Let's rewrite the equation of the plane in the form

$$K_x x + K_y y + K_z z + K_0 = 0, \quad (4.4)$$

then, from (4.3) and (4.4)

$$K_x = \begin{vmatrix} V_{2y} & V_{2z} \\ V_{3y} & V_{3z} \end{vmatrix} \quad K_y = -\begin{vmatrix} V_{2x} & V_{2z} \\ V_{3x} & V_{3z} \end{vmatrix} \quad K_z = \begin{vmatrix} V_{2x} & V_{2y} \\ V_{3x} & V_{3y} \end{vmatrix} \quad K_0 = -K_x V_{0x} - K_y V_{0y} - K_z V_{0z}. \quad (4.5)$$

Note that it is easy to shift the constructed plane to the point V_0 , for this it is enough to change the free term in equation (4.4)

$$K_x(V_{0x} + \tau L_x) + K_y(V_{0y} + \tau L_y) + K_z(V_{0z} + \tau L_z) + K_0 = 0, \quad (4.6)$$

The plane to which the point V_0 belongs divides space into two regions, if some test point belongs to the plane the equation with a free term (4.6) gives zero, if the point lies on one side of the plane the expression (4.4) with a free term (4.6) is greater than zero, on the other – less. Moreover, when the plane rotates, the areas do not change places. This is used to determine the visibility of objects, a condition of visible objects is $K_x x + K_y y + K_z z + K_0 > 0$.

It remains to find the intersection of the line and the plane. It is very fortunate that the line PG is given in the parametric form (4.2), we denote the vector $L = V_0 - P$, and substitute the coordinates of the line in the equation of the plane (4.4)

$$K_x(V_{0x} + \tau L_x) + K_y(V_{0y} + \tau L_y) + K_z(V_{0z} + \tau L_z) + K_0 = 0. \quad (4.7)$$

Expressing the parameter τ , we obtain

$$\tau = -\frac{K_x V_{0x} + K_y V_{0y} + K_z V_{0z} + K_0}{K_x L_x + K_y L_y + K_z L_z}. \quad (4.8)$$

Now the point G satisfies the equation of the line (4.2) and the equation of the plane (4.4). Find the coordinates of the point G in the basis given by the vectors V_2, V_3 , that is, in the coordinates of the screen plane. The projection of a vector onto a line given by a vector is the scalar product of these vectors divided by the module of the vector defining the line. In our case, the last vector is normalized to unity, whence we get

$$\begin{aligned} V_x &= (G_x - V_{0x})V_{2x} + (G_y - V_{0y})V_{2y} + (G_z - V_{0z})V_{2z}, \\ V_y &= (G_x - V_{0x})V_{3x} + (G_y - V_{0y})V_{3y} + (G_z - V_{0z})V_{3z}. \end{aligned} \quad (4.8)$$

The coordinates found are the screen coordinates of the point P projected from the WCS onto the screen plane.

The coordinates in the projection plane are obtained using the central projection, which means that the laws of perspective work automatically, the parameter determining the reduction of distant objects is h , which defines the center of the projection. If the center of the projection is far removed from the plane of view, the central projection approaches to a parallel projection.

4.2 Ray tracing

Since the simulated world consists mainly of spheres, it is not difficult to implement the ray tracing method, at least in a simple form, without taking into account refraction and reflection. The program implements a method of reverse ray tracing (from the screen plane to objects) with an algorithm for calculating *Lambert-Phong-Blinn* illumination.

Let the coordinates of the point of the screen plane be known V_X, V_Y , construct a straight line passing through a point with coordinates V_X, V_Y and the point of the central projection given by the vector V_0 (Fig. 4.1). The coordinates of the first point will be set by the vector

$$\vec{G} = \vec{V}_2 \cdot V_X + \vec{V}_3 \cdot V_Y + \vec{V}S_0. \quad (4.10)$$

The equation of the desired line, let's call it a direction of view line, is convenient to represent in parametric form

$$\vec{V} = (\vec{V}_0 - \vec{G})\tau + \vec{G} \quad (4.11)$$

The equation of the intersection of a line (4.11) and a sphere of radius R representing an atom centered at a point with coordinates (x_0, y_0, z_0) will take the form

$$(G_x + \tau(V_{0x} - G_x) - x_0)^2 + (G_y + \tau(V_{0y} - G_y) - y_0)^2 + (G_z + \tau(V_{0z} - G_z) - z_0)^2 = R^2. \quad (4.12)$$

Let's rewrite (4.12) in a simple way

$$\tau^2(b_x^2 + b_y^2 + b_z^2) + 2\tau(b_x a_x + b_y a_y + b_z a_z) + a_x^2 + a_y^2 + a_z^2 - R^2 = 0, \quad (4.13)$$

where $b_x = V_{0x} - G_x$, $a_x = G_x - x_0$, and the same way for y, z .

If the quadratic equation (4.13) has two solutions, we find the smallest parameter τ , which determines the nearest point of intersection of the direction of view line with the sphere. Denote the coordinates of the found intersection point (P_x, P_y, P_z) .

The normal to the surface of the sphere, is given by a straight line passing through a point on the sphere and the center of the sphere

$$n_x = P_x - x_0, \quad n_y = P_y - y_0, \quad n_z = P_z - z_0. \quad (4.14)$$

The direction of view is set by the guiding vector of the straight line (4.11)

$$V_x = -V_{0x} + G_x, \quad V_y = -V_{0y} + G_y, \quad V_z = -V_{0z} + G_z. \quad (4.15)$$

After normalization of vectors (4.14), (4.15), the cosine of the angle between them is found

$$\cos(\alpha) = n_x V_x + n_y V_y + n_z V_z \quad (4.16)$$

Next, a vector is calculated, known as the 'halfway' vector, which is the difference between the direction of view vector and the vector \vec{L} defining the light source.

$$H_x = V_x - L_x, \quad H_y = V_y - L_y, \quad H_z = V_z - L_z. \quad (4.17)$$

After normalization of the vector, the cosine of the angle between the vector \vec{H} and the normal to the surface is considered

$$\cos(\theta) = H_x n_x + H_y n_y + H_z n_z \quad (4.18)$$

Now there are all the necessary values to determine the color components (C_{RED} , C_{GREEN} , C_{BLUE}) for a pixel with coordinates V_x, V_y

$$\begin{aligned} C_{RED} &= 255 \cdot (S_{RED} (K_{DIFFUSE} \cos(\alpha) + K_{STRAY}) + K_{REFLECT} \cos(\theta)^7), \\ C_{GREEN} &= 255 \cdot (S_{GREEN} (K_{DIFFUSE} \cos(\alpha) + K_{STRAY}) + K_{REFLECT} \cos(\theta)^7), \\ C_{BLUE} &= 255 \cdot (S_{BLUE} (K_{DIFFUSE} \cos(\alpha) + K_{STRAY}) + K_{REFLECT} \cos(\theta)^7). \end{aligned} \quad (4.19)$$

The values S_{RED} , S_{GREEN} , S_{BLUE} represent the color components of the surface of the sphere depicting the atom, which are editable on the 'Atom' tab of the 'Edit' window (Fig. 4.8). The exponent for $\cos(\theta)$ is responsible for the size of the glare, it is an empirical value prescribed in the program code.

Ray tracing when constructing a surface bounding a group of atoms (§3.13) is performed for the elements of the plane, which are obtained by dividing the surface into triangles. A grid is

stretched on the surface with a uniform pitch along the angular variables. For each grid cell, a point is plotted in the center of the cell, which is a common point for the four triangles. For each of the four triangles, a ray tracing procedure is called, at the input of which there are coordinates of three points specified by vectors T_1 , T_2 , T_3 . Let the common point of all triangles be point T_3 . The vector specifying the direction of view for the four triangles is assumed to be the same and equal to $V=T_3-V_0$.

Let's construct two vectors $V_1=T_2-T_1$, $V_2=T_3-T_1$, which can be considered as the basis of the plane given by the points T_1 , T_2 , T_3 , then the normal to the plane is $n=V_1 \times V_2$. After normalization of the vectors, the cosine of the angle between the direction of view and the normal to the surface is calculated by the formula (4.16), then the 'halfway' vector and the cosine of the angle between the direction to the light source and the 'halfway' vector are calculated by the formulas (4.17), (4.18).

Taking into account the transparency, the shading of the inner area of the triangle is calculated as

$$\begin{aligned} C_{RED} &= (1 - K_{TRANSPARENT}) C1_{RED} + K_{TRANSPARENT} C2_{RED}, \\ C_{GREEN} &= (1 - K_{TRANSPARENT}) C1_{GREEN} + K_{TRANSPARENT} C2_{GREEN}, \\ C_{BLUE} &= (1 - K_{TRANSPARENT}) C1_{BLUE} + K_{TRANSPARENT} C2_{BLUE}, \end{aligned} \quad (4.20)$$

where $C1_{RED}$, $C1_{GREEN}$, $C1_{BLUE}$ are the color components of the plane element calculated by the formulas (4.19), $C2_{RED}$, $C2_{GREEN}$, $C2_{BLUE}$ are the color components of the drawn scene stored in the virtual screen memory, $K_{TRANSPARENT}$ is the transparency coefficient set on the 'Workspace\Ray tracing' form (Fig. 4.6). The calculation of the color components (4.21) is performed for each pixel, this requires using a fast-acting procedure for accessing the virtual screen, the program code uses access to the Bitmap array through direct access to memory (C code notation):

`ptr = (byte*)Bitmap->ScanLine[BmpMaxYd2-sm].`

4.3 Program management

The main screen (Fig. 4.2) is the result of constructing the central projection (4.8) for the all atoms.

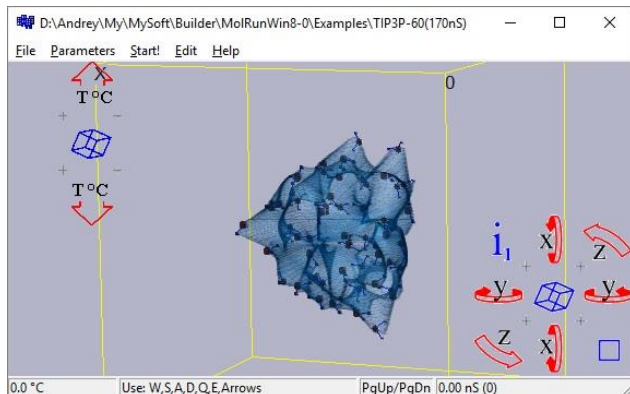


Fig. 4.2. The main program window.

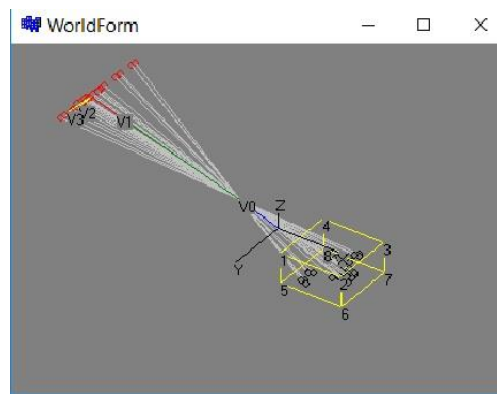


Fig. 4.3. The relative position of the projection plane and the cube.

The free flight of the projection plane is carried out by the arrows and keys W, S, A, D, Q, E. The position of the projection plane with respect to the cube can be viewed in the 'Parameters\Work Space' window (Fig. 4.4). The possibility of free flight of the projection plane allows the loss of a

molecules from the field of view, if this happens, the ‘World’ window will help you find the molecules back.

At the bottom right and top left of the main window (Fig. 4.2) there are navigation areas that change their appearance when you click on the central part of the navigators. The PgUp, PgDn keys allow you to sort through the molecules, the selected molecule can be edited. If you select the menu item ‘Edit’. In editing mode, all molecules except the selected one will disappear and the editor window will open. While editing the selected molecule, it remains possible to freely change the position of the projection plane in space.

4.4 The ‘WorkSpace’ window

The main parameters responsible for the calculation of the molecular ensemble dynamics are collected on the tabs of the ‘Work Space’ window. The window contains the tabs ‘Load’, ‘View’, ‘Physics’, ‘Molecules’.

The ‘Load’ tab is a text report about loading process of parameters describing an ensemble of molecules.

The ‘View’ tab (Fig. 4.4) contains the parameters responsible for displaying the ensemble of molecules.

After the coordinates of the atom are obtained by the formula (4.8), all atoms are sorted by distance in the direction of view (vector V_1). Scene rendering starts from the farthest atom, this part of the algorithm can be eliminated by unchecking ‘Z buffered’. The following is the part of the algorithm that displays the atom in the form of a filled circle, the ‘Draw ball’ checkmark, the part of the algorithm responsible for displaying the bond between the atoms is marked the ‘Draw bonds’ checkmark, and finally, the ‘Atom Name’ checkmark is responsible for the code that labeled the atom’s name.

Two checkboxes ‘Analysis background’ and ‘Potential background’ allow you to display the contents of the ‘Analysis’ and ‘Power potential’ windows as the background for the molecules on the main screen. This feature is provided for creating videos.

The checkbox ‘Calculation Report’ enables the mode of displaying internal variables on the graphical screen when computational errors occur.

In order for the calculation process to be accompanied by recording frames in a video file, just check the ‘Save AVI when Start’ checkbox and select the ‘Frame Time’ option. The file name and location of the video will match the name and location of the description file of the downloaded scene (*.mlc).

In the process of constructing the central projection, the scaling factor is determined by two parameters, the distance from the center projection point to the object d_1 and the distance from the center projection point to the view plane d_2 , font scaling is performed according to the formula $FontSize = 3 + (d_2/d_1) * FontScale$. ‘Font Scale’ is available for editing in the window of the same name.

The ‘Mouse wheel’ contains two more parameters that can be changed using the mouse wheel. This is the offset to the projection center ‘Central projection Point’, which is the parameter h from formula (4.1) and the common factor ‘Scale Factor’, by which all coordinates in the projection plane are multiplied.

The parameters circled by the general outline ‘Info’ are information blocks, the display of which is activated by clicking on the corresponding navigation area i_1, i_2, i_3, i_4 , located in the lower right corner of the main window (Fig. 4.2). The filling of the block is determined by the parameters

selected on the ‘View’ tab. Information blocks i_1 , i_2 relate to the whole scene, and blocks i_3 , i_4 relate to the selected molecule.

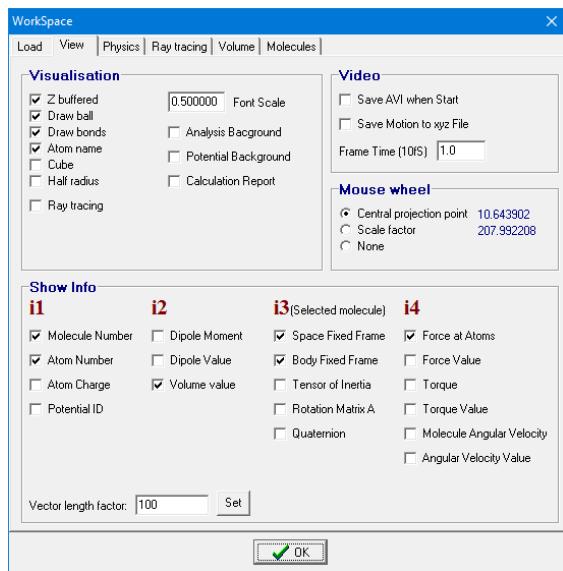


Fig. 4.4 The window ‘WorkSpace’, tab ‘View’.

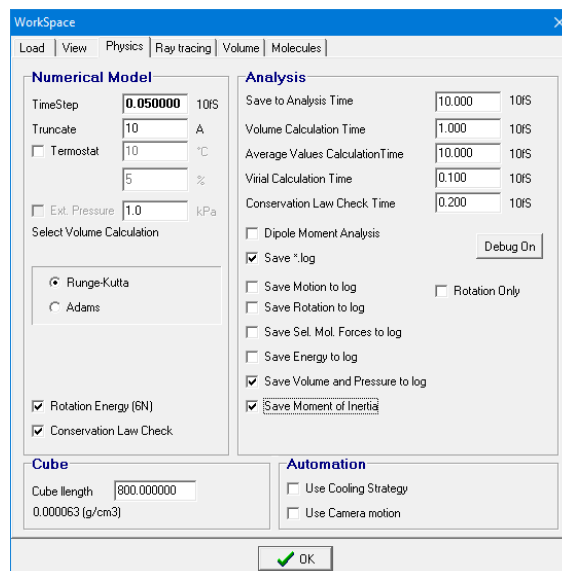


Fig. 4.5 The window ‘WorkSpace’, tab ‘Physics’.

The ‘Physics’ tab, fig. 4.5 contains parameters related to the physical model of the described scene. The main parameter that determines the dynamics of the calculation is ‘TimeStep’, this is a step of the algorithm for the numerical integration of the equations of motion. As a numerical integration algorithm, one can choose a fourth-order *Runge-Kutta* algorithm or an implicit fourth-order *Adams* method.

Comparing the work of the *Runge-Kutta* algorithm with the implicit *Adams* method, I could not make out the differences between them. As a comparison criterion, the accuracy of conservation of the total energy of the system during the collision of two model particles in a strongly varying potential (2.17) was used. Both methods of the 4th order give the same accuracy and have the same computational cost.

The ‘Truncate’ field specifies the distance of half of the edge of the virtual cube associated with the center of mass of each molecule. Molecules whose centers of mass is located within this cube participate in the total force acting on the molecule. The action of molecules outside the cube is ignored.

The ‘Termostat’ checkbox includes an algorithm, such as *Berendsen*, which is a global thermostat that scales all molecules [20], the velocity vector of each molecule retains direction, scales according to the target temperature, while the effect on velocity is set by a parameter expressed as a percentage.

The ‘Ext. Pressure’ checkbox is available only if the *Volume* tab, press *Volume* button calculation mode is selected. This mode includes external force field acting on each molecule. The external field is given by a quadratic potential and is known as *harmonic confinement*. The system becomes non homogeneous, the target pressure corresponds to the center of the accumulation of molecules. Such a barostat algorithm is not standard and is under discussion.

Flag ‘Rotation Energy (6N)’ must be reset for monatomic molecules see formula (3.66).

The ‘Conservation Law Check’ checkbox enables an algorithm that calculates the total

moment of the amount of movement. In the process of compression of the system by an external force field, an uncompensated moment of forces arises, leading to the unwinding of the system. When a total amount of movement occurs, a counter-torque will be formed to prevent unwinding.

The ‘Analysis’ parameter group contains the time intervals that calling computational procedures.

‘Save to Analysis Time’ is the time interval after which data is recorded that can be tracked as graphs in the Analysis window.

‘Volume calculation Time’ is the time interval after which the surface covering the molecules is constructed and its volume is calculated.

‘Average Values Calculation Time’ is the time interval after which the average values of energy, temperature, pressure are calculated.

‘Virial Calculation Time’ is the time interval after which the instantaneous value of the *Clausius* virial is calculated.

‘Conservation Law Check Time’ is the time interval after which the total kinetic moment is calculated.

The ‘Debug Off/On’ button makes available the fields that control the recording of the calculation process parameters in the file. All variables are written to the file in program units (3.8), time in units of 10fS, distance in angstroms, etc.

In addition, in Debug On mode, the hotkey ‘K’ becomes available, which opens a window named ‘Keyhole’, which displays the current values of key program variables. A copy of the contents of the ‘Keyhole’ window is saved in a file named KHhh.mm.SS.ss.txt, where hh.mm.SS.ss is the system time in the format hh - hours, mm - minutes, SS - seconds, ss – milliseconds. If the runtime error occurred in the thread performing the calculation, the option to call the ‘Keyhole’ window should remain operational, since it is implemented in the thread processing the events of the main window, which allows you to save the current values of the calculated parameters for analyzing the causes that caused the error.

The ‘Save Motion to log’ checkbox enables recording the coordinates of the centers of mass of molecules and the *Rodrigue-Hamilton* parameters in a binary file with the extension ‘xyz’. The file format is described in §4.9. To play the motion stored in the ‘xyz’ file, the utility is used xyzPlayer.exe, which loads scene parameters from a file with the *.mlc extension and motion path from a file with the ‘xyz’ extension.

Each log file is recorded in text form, a list of variables is listed in the file header, the time interval the recording is performed corresponds to the ‘Save To Analysis Time’ variable.

The parameter ‘Cube length’, sets the size of the cube which keeps the molecules are placed.

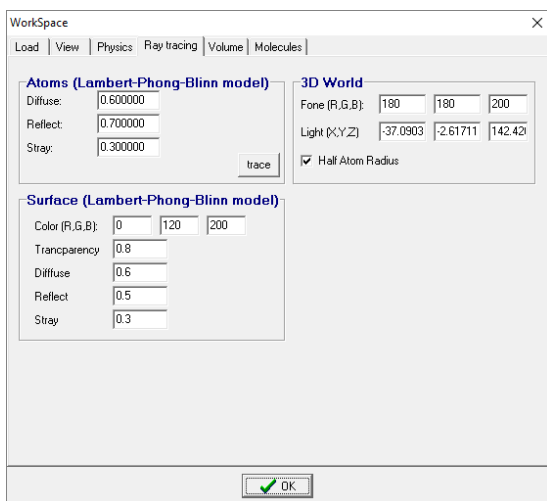


Fig. 4.6 'WorkSpace' window, 'Ray tracing' tab.

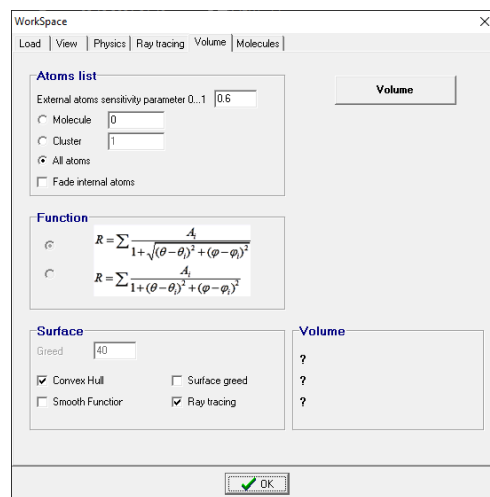


Fig. 4.7 The 'WorkSpace' window, the 'Volume' tab.

The 'Ray tracing' tab (Fig. 4.6) contains parameters that control ray tracing, described in §4.2. By default, ray tracing is disabled (check mark on the 'View' tab), because it reduces the calculation performance. The frame 'Atoms (Lambert-Phong-Blinn model)' contains the coefficients K_{DIFFUSE} , K_{REFLECT} , K_{STRAY} , which determine the realistic visualization of atoms. Similarly, the frame 'Surface (Lambert-Phong-Blinn model)' contains coefficients for constructing a realistic surface covering a set of atoms.

The '3D World' frame contains the components of the background color of the main window, the coordinates of the light source and the 'Half atom radius' check mark, which reduces the radius of the atoms by 2 times when the ray tracing mode is selected.

The 'Volume' tab (Fig. 4.7) contains parameters that affect the calculation of the volume of a molecule or a cluster. The 'Atoms list' frame contains parameters defining the selection of outer layer atoms of the molecule or cluster, which were described in §3.13. To force the selection of clusters, you can click on the 'Cluster Analysis' button on the 'Molecules' tab.

The 'Surface' frame defines the method of surface construction described in §3.13.

The 'Molecules' tab contains buttons that allow you to add molecules from another file to the scene, delete a molecule, clone a selected molecule by assigning random coordinates to it.

4.5 The 'Edit' window

The editor window contains the tabs 'Atoms', 'Bond', 'Force', 'Molecule', 'Debug'.

The 'Atoms' tab (Fig. 4.6.) lists all the selected molecule atoms.

First, the parameters related the whole molecule:

- the molecule name, the number of atoms, the total molecule mass;
- the center of mass velocity;
- the angular velocity included in equation (3.38);
- the center of mass current coordinates;
- the inertia tensor main values;
- the main axis direction of inertia I_1 ;
- the main axis direction of inertia I_2 ;
- the main axis direction of inertia I_3 .

The next is a listing of each atoms parameters:

Periodic table atom number, the atom name, the empty field;

- the atom coordinates in the world coordinate system;
- the atom mass, the radius, the charge;
- R, G, B colors (0 ÷ 255) component.

Mass,Radius,Charge	35.453000	0.970000	0.000000	Color	31	240	31	Atom 2 Name:	63	BR
Atom Position:			11.078224	-1.286794	-1.763903					
Mass,Radius,Charge	79.904000	1.120000	0.000000	Color	166	41	41	Atom 3 Name:	49	I
Atom Position:			8.729388	-1.920749	-3.957186					
Mass,Radius,Charge	14.820000	2.000000	0.000000	Color	166	117	115	Atom 4 Name:	1	H
Atom Position:			10.529416	0.044333	-3.800599					
Mass,Radius,Charge	1.007970	0.790000	0.000000	Color	60	60	60	Atom 5 Name:	6	C
Atom Position:			2.428518	-1.959827	-6.967748					
Mass,Radius,Charge	12.011150	0.910000	0.000000	Color	144	144	144	Atom 6 Name:	63	CL
Atom Position:			1.952444	-0.672830	-5.868533					

Fig. 4.8 The 'Edit' window, 'Atoms' tab.

C1r	0(C)	1(CL)	2(BR)	3(I)	4(H)	5(C)	6(CL)	7(BR)	8(I)
0(C)	X	X	X	X	X	X	X	X	X
1(CL)	S	X	X	X	X	X	X	X	X
2(BR)	S	0	X	X	X	X	X	X	X
3(I)	S	0	0	X	X	X	X	X	X
4(H)	S	0	0	0	X	X	X	X	X
5(C)	0	0	0	0	0	X	X	X	X
6(CL)	0	0	0	0	0	S	X	X	X
7(BR)	0	0	0	0	0	S	0	X	X
8(I)	0	0	0	0	0	S	0	0	X
9(H)	0	0	0	0	0	S	0	0	0

Fig. 4.9 The 'Edit' window 'Bonds' tab.

The tab 'Bonds' (Fig. 4.9) shows the matrix of bonds of the edited molecule. This window also used to edit the distances between atoms and angles between selected directions.

The program recognizes the atom by clicking the mouse in the main window. The selected atom is displayed in the main window with a red circle with a crosshair (Fig. 4.10), in the list of atoms (Fig. 4.8), the selected atom is displayed with a red field in the left column of the table, in the bonds table (Fig. 4.9), the selected atom is displayed with a darkened row of column. You can also select an atom by clicking on the Atom row in the left column of the atom table (Fig. 4.8).

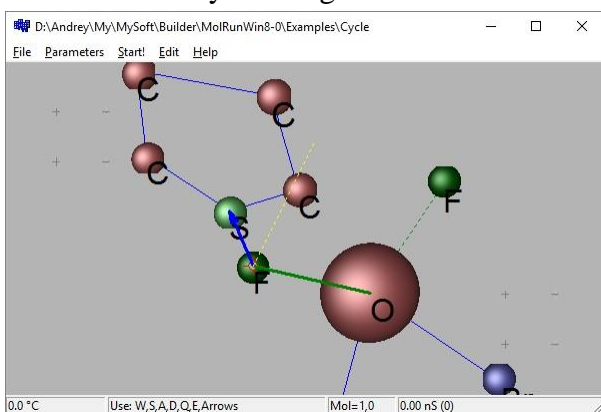


Fig. 4.10 Editable molecule.

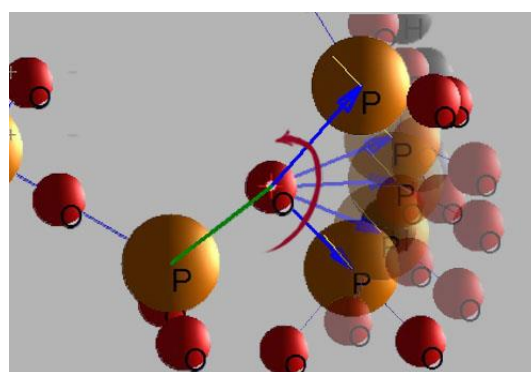


Fig. 4.11 Change of the valence angle.

To change the length of the bond or the angle between the bonds, you need to select vectors using the right and left mouse buttons by clicking on the fields of the bonds table (Fig. 4.9). The selected vectors are displayed in the main window (Fig. 4.10). If the vectors have a common point and it coincides with the selected atom, the 'Valence Angle' parameter is activated and it becomes possible to change the angle for the atom pointed at by the arrow, or the group of atoms pointed at by the arrow (Fig. 4.11). The axis of rotation is displayed as a yellow dotted line.

To change the torsion angle, it is necessary that the blue and green vectors are located on different sides of the twisted link, as shown in Fig. 4.11. The axis of rotation is the selected bond, which is displayed in red.

If only the blue vector is selected, the ‘Distance’ field is activated and it becomes possible to change the bond length to the atom indicated by the blue arrow, or to a group of atoms.

You can cancel the selection of vectors intended for editing by clicking on the upper-left corner of the bonds table, marked as ‘Clr’.

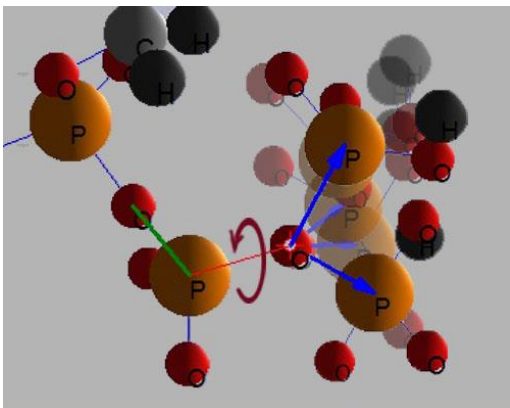


Fig. 4.12 Change of the torsion angle.

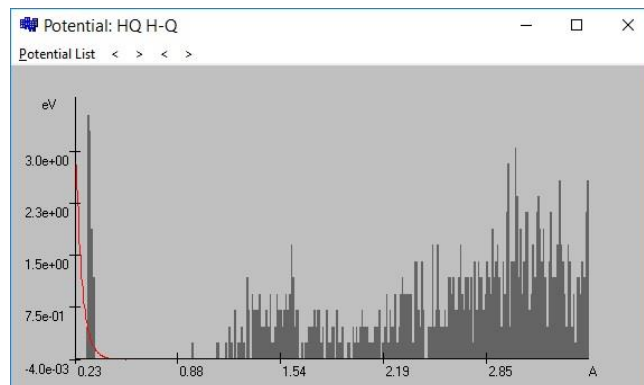


Fig. 4.13 The 'Power Potential' window.

The ‘Force’ tab shows a list of the interaction potentials parameters for the atoms that make up the editable molecule, the ‘Force’ tab is shown in Fig. 2.14.

The ‘Molecule’ tab is shown in fig. 2.2 and is described in §2.1.

The ‘Debug’ tab allows you to explore the rotation matrix, and is described in §3.6.

4.6 The window ‘Power potential’

The ‘Power Potential’ window (Fig. 4.13) allows you to select the interaction potential of a pair of atoms (red line) which displays a histogram of interatomic distances for the selected pair of atoms. The arrows at the top of the window allow you to change the distance interval displayed along the X axis.

4.7 The window ‘Analysis’

‘Analysis’ window, Fig. 4.10 is used to display graphs of temperature (3.61), kinetic energy of translational degrees of freedom (3.52), kinetic energy of rotational degrees of freedom (3.54), potential energy (3.58) and total energy, which is considered as the sum of all the listed energies.

The time scale of all graphs is represented in pS, using the notation introduced in §3.1 from (3.3) we obtain

$$t_{(pS)} = 0.01 \cdot t_{(10fS)}.$$

The parameters to describe the ensemble of molecules are calculated at time intervals specified by the ‘Save to Analysis’ parameter, which is located on the ‘Physics’ tab of the ‘WorkSpace’ window.

The ‘Average Time’ parameter sets the time during which the sum of the energies of all

degrees of freedom is accumulated to calculate the temperature.

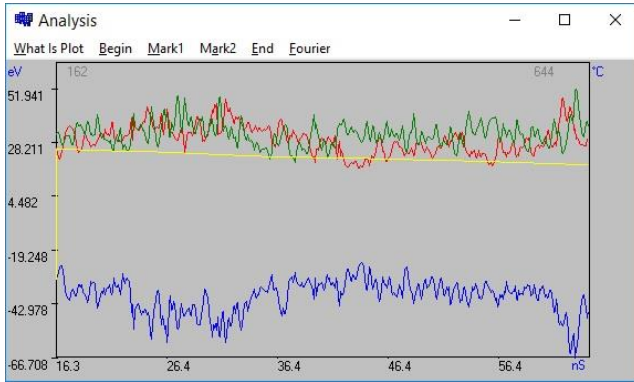


Fig. 4.14 The 'Analysis' window.

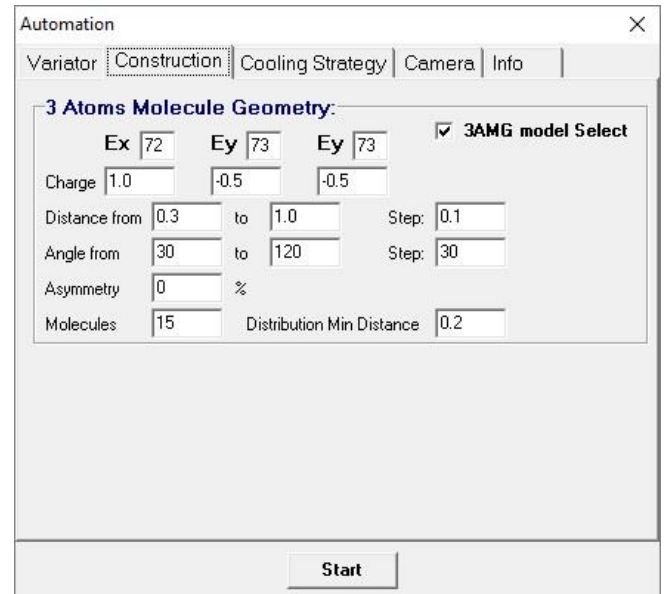


Fig. 4.15 The 'Automation' window, 'Constructor' tab.

4.8 The 'Automation' window

The 'Automation' window is an add-on control for the molecular dynamics program. Here you can automatically generate molecules, set the parameters providing condensation of the molecular crystal, set the parameters of the automatic movement of the camera. The 'Automation' window contains the tabs 'Variator', 'Constructor', 'Cooling Strategy', 'Camera', 'Info'.

The 'Constructor' tab (Fig. 4.15) contains the parameters of the automatic generator of charged spatial structures. The spatial structure generation begins with the '3AMG' (3 Atoms Molecule Geometry) method. This method builds in space 3 atoms with a given angle α between atoms and given distances d between atoms. The reference plane defined by three atoms can be considered as the basis for further atomic growths.

The construction is performed as follows. The first atom is placed at a point with random coordinates (x_0, y_0, z_0) . For the second atom, we construct a unit vector with random coordinates (x_1, y_1, z_1) , and depart from the first atom in the direction of a random unit vector by a distance d . To find the coordinates of the third atom (x_2, y_2, z_2) , one must satisfy the equation

$$x_1 x_2 + y_1 y_2 + z_1 z_2 = \cos(\alpha), \quad (4.9)$$

The desired vector (x_2, y_2, z_2) , we assume it to be unity, describes a cone with the axis of symmetry given by the vector (x_1, y_1, z_1) . Due to the arbitrary orientation of the first vector, we can choose any solution, however, we cannot assign arbitrary values to two coordinates and find a third, because it is difficult to guess the values at which the equation has a solution. We expand the space of our searches as follows. We introduce a straight line perpendicular to the first vector. Points located on a line with coordinates (x_3, y_3, z_3) must satisfy the equation

$$x_1 x_3 + y_1 y_3 + z_1 z_3 = 0. \quad (4.9)$$

If the coordinate z_1 is not equal to zero, we can choose arbitrary values x_3, y_3 , and construct a

solution

$$x_3 = 1 \quad y_3 = 1 \quad z_3 = -\frac{x_1 + y_1}{z_1} . \quad (4.10)$$

We normalize the vector (x_3, y_3, z_3) by one and consider the vectors (x_1, y_1, z_1) and (x_3, y_3, z_3) as a local orthonormal basis in which the desired unit vector (x_2, y_2, z_2) has the coordinates $\cos(\alpha)$ for the first axis and $\sin(\alpha)$ for the second axis. Having retreated, at these distances in the direction of the basis vectors and adding them, we obtain the coordinates of the vector (x_2, y_2, z_2) :

$$\begin{aligned} x_2 &= x_1 \cos(\alpha) + x_3 \sin(\alpha) , \\ y_2 &= y_1 \cos(\alpha) + y_3 \sin(\alpha) , \\ z_2 &= z_1 \cos(\alpha) + z_3 \sin(\alpha) . \end{aligned} \quad (4.11)$$

In the direction of the found vector (x_2, y_2, z_2) we step back by a distance d .

The 'Construction' tab also has the Asymmetry parameter, expressed as a percentage, this parameter allows you to add asymmetry to the distances between atoms, the distance to the second atom is calculated as $d(1 - \text{Asymmetry}/100)$, to the third atom, as $d*(1 + \text{Asymmetry}/100)$.

Of course, in WCSs, all vectors that specify the coordinates of atoms are also shifted in the direction of the vector (x_0, y_0, z_0) .

There are three atoms on the 'Constructor' tab, clicking on which opens the periodic table (Fig. 4.12), in the first period, Ex, Ey atoms are added, to which you can assign arbitrary properties.

Below are the charges 'Charge', attributed to the point at which the atom is located. Below is the 'Distance' parameter (d in formulas (4.9 - 4.11)), which will run through the set values within the given limits with the step 'Step'.

Next is the 'Angle' parameter (α in formulas (4.9 - 4.11)), which runs over the values within the given limits with the step 'Step'.

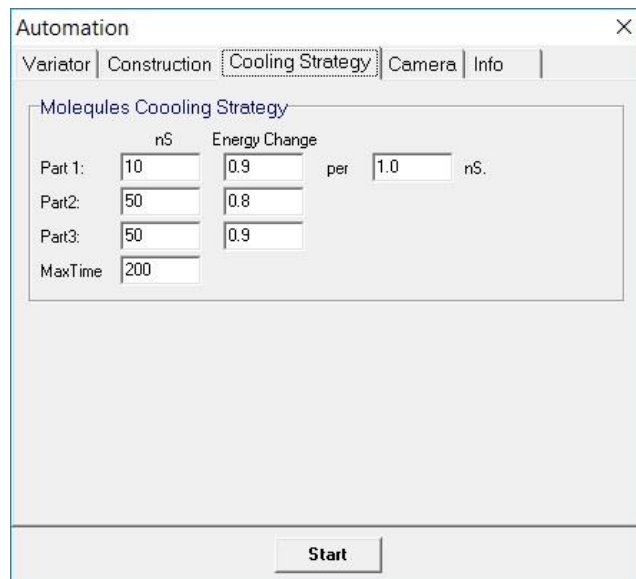
The last line sets the number of molecules 'Molecules' and the 'Distribution Min Distance' parameter, which indicates that if the molecules are randomly located in the cube, the distance between the nearest ones should not be less than the specified value. If this condition is not satisfied, the program will build a new distribution until it exhausts the maximum number of attempts.



Mendeleev's Table not Periodic Tadle

H	Ex	Ey						He		
Li	Be	B	C	N	O	F	Ne			
Na	Mg	Al	Si	P	S	Cl	Ar			
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	
Cu	Zn	Ga	Ge	As	Se	Br	Kr			
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	
Ag	Cd	In	Sn	Sb	Te		Xe			
Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	
Au	Hg	Tl	Pb	Bi	Po	At	Rn			

Fig. 4.16 Periodic tabel.



Automation

Variator | Construction | **Cooling Strategy** | Camera | Info

Molecules Cooling Strategy

nS Energy Change

Part 1: per nS.

Part 2:

Part 3:

MaxTime

Start

Fig. 4.17. The window 'Automation', 'Cooling Strategy' tab.

When you click the ‘**Start**’ button in the ‘**Automation**’ window, the program will automatically create a scene from the constructed molecules, start the algorithm for solving dynamics equations, and the ‘**Cooling Strategy**’, ‘**Camera**’ tabs parameters will take effect.

The ‘**Cooling Strategy**’ tab, Fig 4.17, contains a parameters that sets the total simulation time ‘**Max Time**’ and parameters that control the condensation of molecules. Over the time intervals Part1, Part2, Part3, the linear and angular velocities are multiplied by the coefficient K, denoted as ‘**Multiply**’, the multiplication is performed at the time interval specified after the ‘**per**’ prefix.

The ‘**Camera**’ tab contains the period during which the type of movement selected from the drop-down window is valid and the time interval over which the camera’s movement is applied.

The ‘**Info**’ tab contains a text report on the process of generating an ensemble of molecules.

If the checkbox ‘**Save AVI when Start**’ is checked before starting the program from the ‘**Automation**’ window, the program will automatically generate scenes with various molecules and save the video file with the process of their condensation.

Once ‘**Max Time**’ is reached, the program stops the simulation of the mechanic’s equations and proceeds to the analysis of the condensed state. First, molecular clusters are determined; for this, a list of all pairs of atoms is constructed, the distance between which is less than the minimum point of the potential interaction energy r_m (2.16). Then an arbitrary atom is taken from the resulting list and the construction of a tree of atoms having a connection with it begins, each atom assigned to the tree is deleted from the original list. The procedure for constructing trees continues until a single atom remains in the list of coupled pairs of atoms. As a result, we get lists of atoms bound in clusters.

For each cluster found, the center of mass, the inertia tensor (as a characteristic of the asymmetry of formation), the dipole moment vector, and the dispersion of the coordinate distribution, which is called gyration radius in [27], are calculated and calculated as

$$Rg = \frac{1}{m_c} \left[N_c \sum_i^{N_c} (m_i r_i)^2 \right]^{1/2}.$$

Since we already have the inertia tensor for the cluster, the following quantity is calculated as a parameter of a similar gyration radius

$$Rg = \frac{1}{N_c} \left[\sum_i^{N_c} (x_i - x_o)^2 + (y_i - y_o)^2 + (z_i - z_o)^2 \right]^{1/2}, \quad (4.25)$$

where the index i runs through all the coordinates of the atoms of the cluster, the index “0” marks the coordinates of the center of mass of the cluster. All calculated parameters are saved in a text file with the *.log extension and the name matching the scenario filename.

4.9 File structures

The ensemble of molecules is represented on the disk by two text files with the same names: a file with the *.mlc extension and a file describing interaction potentials with the *.ptn extension.

The *.mlc file reads line-by-line information, lines starting with a semicolon are comments and ignored by program.

Below is an example of a file, the contents of a text file are shown in bold, the comments shown italic blue.

; **Object identifications:** - *comment;*

; - comment;;
 3 – the number of molecules in the scene;
 0.000200 – step of numerical integration, nS;
 0.100000 – Save to Analysis, nS;
 2.000000 – Average Time, nS;
 2.000000 – Volume Time, nS;
 2.000000 – Virial Time, nS;
 ;Box parameters: - comment;
 3.800000,0.300000,-1.199998 – cube center;
 15.000000 – cube edge length;
 Reflect – molecules are reflected from the walls of the cube; possible values are Reflect, Thru;
 ;Camera: - comment;
 9.091769,-0.054743,-17.832886 – vector V0;
 -0.342939,0.022544,0.939087 – vector V1;;
 0.939356,0.010324,0.342790 – vector V2;
 -0.003449,0.999686,-0.024814 – vector V3;
 12.354342 – indent for a central projection point;
 207.992208,0.100000 – general scale (Scale Factor), font scale (Font Scale);
 ;===== Molecule: 1 ===== - comment;
 Water – name of the molecule;
 5- number of atoms in a molecule
 1.635943,0.184179,-0.366123 – coordinates of the 1st atom;
 1.333735,0.767041,-1.120677 – coordinates of the 2st atom;
 2.433070,0.531541,-0.860432 – coordinates of the 3st atom;
 2.118074,-0.212104,0.134254 – coordinates of the 4st atom;
 0.873951,0.054424,-0.160248 – coordinates of the 5st atom;
 Atom name reference: - obligatory field;
 8,12.000000,O, , - atom number, atom mass, symbol (up to 3 symbols separated by commas);
 1,1.000000,H, , - atom number, atom mass, symbol (up to 3 symbols separated by commas);
 1,1.000000,H, , - atom number, atom mass, symbol (up to 3 symbols separated by commas);
 0,0.000000,Q, , - atom number, atom mass, symbol (up to 3 symbols separated by commas);
 0,0.000000,Q, , - atom number, atom mass, symbol (up to 3 symbols separated by commas);
 0.000000,0.300000 – atomic charge, atomic radius;
 RGB: 200,100,100 - the color with which the circle is painted;
 0.230000,0.100000 – atomic charge, atomic radius;
 RGB: 100,100,200- the color with which the circle is painted;
 0.230000,0.100000 – atomic charge, atomic radius;
 RGB: 100,100,200- the color with which the circle is painted;
 -0.230000,0.000000 – atomic charge, atomic radius;
 RGB: 0,100,0- the color with which the circle is painted;
 -0.230000,0.000000 – atomic charge, atomic radius;
 RGB: 0,100,0- the color with which the circle is painted;
 Center of gravity: - obligatory field;
 1.671295,0.250624,-0.455327- coordinates of the center of mass of the molecule;
 1.810111,0.665863,1.144248 – main values of the inertia tensor of the molecule;
 0.027588,0.796402,0.604139 – direction of the main axis of inertia;
 -0.302893,-0.569296,0.764303- direction of the main axis of inertia;
 0.952626,-0.204075,0.225519- direction of the main axis of inertia;
 Initial value: - obligatory field;

```

-0.009220,-0.009631,0.009766 - components of the velocity of the molecule;
0.014353,-0.039785,0.045832- components of the angular velocity of the molecule;
14.000000 - the mass of the molecule;
1- attribute of an active (mobile) molecule;
0SSZZ - link matrix S-Single, Z-dot, 0-no link;
00000
00000
00000
00000
;===== Molecule: 2 =====
...

```

The *.ptn file reads line-by-line information, lines starting with a semicolon are considered comments and are ignored.

Below is an example of a *.ptn file, the contents of a text file are shown in bold, the comments shown italic blue.

The first 10 characters are a text comment, then the parameters are read using the C language pattern "% i,% i,% lf,% lf,% lf".

```

; Atom-atom potential: - comment;
; - comment;
6 - number of potentials;
OH O-H 8,1,0.000000,0.000000,0.000000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).
HH H-H 1,1,0.000000,0.000000,0.000000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).
OQ O-Q 8,0,0.000000,0.000000,0.000000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).
QQ Q-Q 0,0,0.000000,0.000000,0.000000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).
HQ H-Q 1,0,0.004000,5.000000,0.561000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).
OO O-O 8,8,0.003280,5.000000,3.479000 - atom number 1, atom number 2,  $\epsilon, \alpha, r_m$  from (2.17).

```

Information about atoms and pair interaction potentials is stored in the Atom.cfg text file. Information from this file is displayed in the 'Mendeleev's Table not Periodic Table' window. Each file line contains the following data: table row number, table column number, 3 characters of the atom's name, atom number, atom mass, atom radius 1 (for metals - metal, for non-metals - covalent), atom radius 2 (for metals - metal, for non-metals - covalent), R, G, B components of the color by which the atom is displayed, parameters of the pair interaction potential ϵ, α, r_m . The program uses the value of the second atomic radius of the atom.

```

1 1 H 1 1.00797 0.79 0.32 60 60 60 0.000 0.000 0.000
8 1 He 2 4.0026 0.49 0.93 217 255 255 0.000880 5.400 2.869
1 2 Li 3 6.939 2.05 1.23 204 128 255 0.000 0.000 0.000
2 2 Be 4 9.0122 1.4 0.9 194 255 0 0.000 0.000 0.000
3 2 B 5 10.811 1.17 0.82 255 181 181 0.000 0.000 0.000
4 2 C 6 12.01115 0.91 0.77 144 144 144 0.000 0.000 0.000

```

The trajectory file has the extension 'xyz' and contains the following C language data structures

```

typedef struct MOTION {
int N;

```

```
double X0,Y0,Z0;  
double I0,I1,I2,I3; } MOTION;
```

where, N is the number of the molecule, variables $X0$, $Y0$, $Z0$ are the coordinates of the center of mass of the molecule, variables $I0$, $I1$, $I2$, $I3$ are the *Rodrigue-Hamilton* parameters. For each time step defined by the *FrameTime* parameter, MOTION-type structures are recorded for all molecules, so a molecule with the number 0 is a sign of the beginning of a new time frame.

4.10 The xyzPlayer.exe utility

To play the trajectory file recorded in the 'xyz' file format, a utility is provided xyzPlaer.exe. The main window of the program is shown in Fig. 4.18, the navigation control panel is shown in Fig. 4.19.

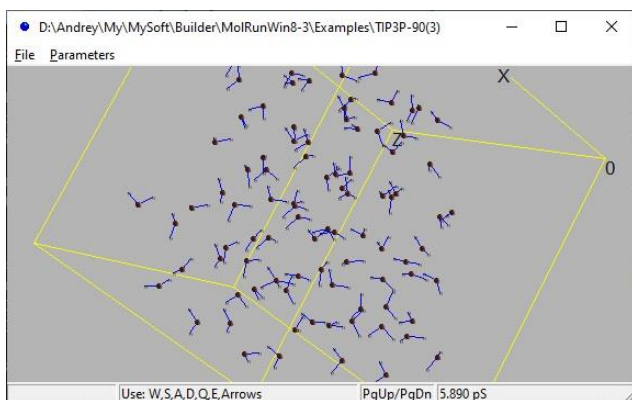


Fig. 4.18 Main window xyzPlayer.exe.

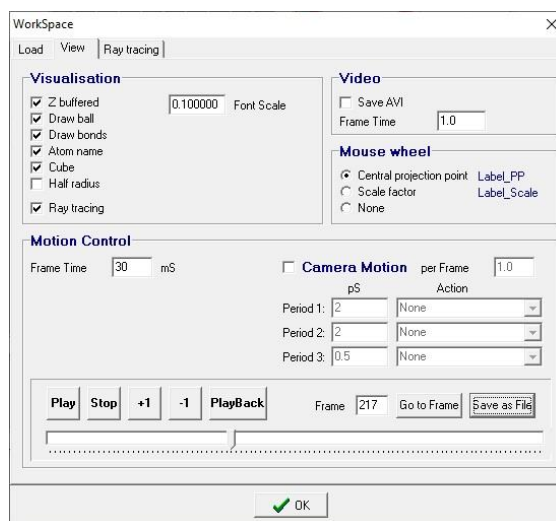


Fig. 4.19 Navigation through the trajectory file.

To download the molecular workspace, you need two files, a scene description file with the extension *.mlc and a trajectory description file with the extension *.xyz.

The controlling interface is the same as in the main program, the 'View' tab contains an additional area of 'Motion Control' (Fig. 4.19). On the 'Motion Control' tab there are buttons for navigating through the frames 'Play', 'Stop', 'Play Back', '+1', '-1'. There is also a 'Go to Frame' button to force a transition to a given frame and a 'Save as File' button that saves the data of the selected frame to a text file so that you can make sure there are no errors when interpreting the binary file of trajectories.

The 'Camera Motion' checkbox allows you to adjust the movement of the 'camera' during the playback of the trajectory file.

The 'Save AVI' checkbox, as before, allows you to save the movement in the 'avi' format file. Calculations of physical parameters (temperature, energy, etc.) are not provided in the xyzPlayer utility.

Calculations of physical parameters (temperature, energy, etc.) are not provided in the xyzPlayer utility.

5 Test of the program

5.1 Conservation of energy test

One of the most easily verified results is the total conservation energy test for NVE ensemble. The total energy conservation is checked separately for translational motion and rotational motion. The results are shown in Fig. 5.1a, b.

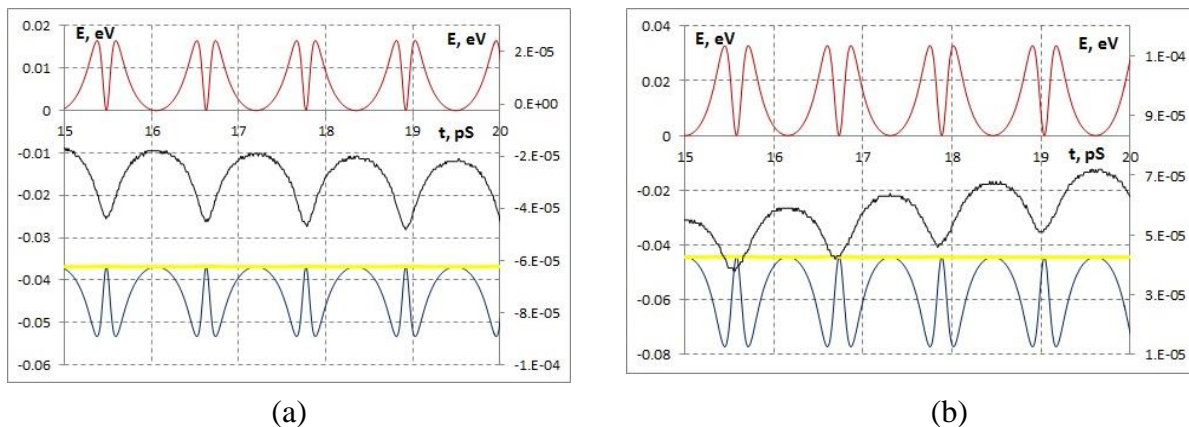


Fig. 5.1 Kinetic energy (red) potential energy (blue) total energy (yellow), the difference between the total energy at the initial moment of time and the instantaneous value of the total energy (black). The black line values are shown on the additional axis on the right side. (a) - translational motion (*2atom.mlc*). (b) - rotational motion.

The model problem of the two atoms collisions in the one-dimensional case was already used in §3.8. Two Ne atoms are separated by a distance of 4 Å, have charges of +0.1 and -0.1 C, interaction potential parameters $\varepsilon = 0.03067$, $\alpha = 5.1$, $r_m = 3.086$, $m = 20.183$. A pair of atoms oscillates indefinitely, the result of calculating the kinetic, potential and total energy is shown in Fig. 5.1 a, b, the step of numerical integration is $0.0005 \times 10 \text{ fS}$, the red line shows the kinetic energy, the blue line, the potential energy, the yellow line, the total energy. The black line shows the difference between the energy at the initial time and the instantaneous total energy, the values are shown on the additional axis on the right side. At time 20 pS, the energy difference is $-4.8 \times 10^{-5} \text{ eV}$, which is 0.13% of the total energy at the initial time.

The group of equations described rotational motion is studied similarly. The model problem is a pair of linear two atoms molecules. The molecules are in the same plane, the lines connecting the of each of the atoms are parallel and spaced 4 Å apart. Atoms located on one side are charged with +0.1 and -0.1 C, the parameters of the interaction potential are the same as before $\varepsilon = 0.03067$, $\alpha = 5.1$, $r_m = 3.086$. Atoms on the other side of the molecule are “balances” that have zero charges and do not interact with each other. One of the molecules is stationary, and the other can freely rotate relative to the center of mass. The main values of the inertia tensor 280.0, 280.0, 280.0, assigned in such a way that the oscillation period coincides with the oscillation period of the first problem. The numerical integration step is $0.0005 \times 10 \text{ fS}$. The system performs infinite oscillation but now the rotation is described by the chain of equations given in Table 3.1.

The described model task is stored in the *rotate1XY.mlc* file, in order to exclude the translational components of the movement, it is necessary to check the ‘Parameters\Workspace\Physics Rotation Only’ checkbox, the option becomes available after clicking the ‘Debug On/Debug Off’ button.

The result of solving equations describing rotational motion is shown in Fig. 5.1 b. the red line shows kinetic energy, the blue line, potential energy, the yellow line, the total energy. The black

line shows the difference between the total energy at the initial moment of time and the instantaneous value of the total energy. At time $t = 20$ pS, the energy difference is -7.2×10^{-5} eV, which is -0.16% of the total energy at the initial time.

In both cases, after approximately 20 collisions of a pair of atoms, the total energy error does not exceed 0.2%. On the other hand, in both cases the total energy drift is visible, due to the instability of the numerical scheme when calculating in the vicinity of a strong change in the pair interaction potential, i.e., in a collision.

The real world objects are necessarily part of something larger, which can be considered as a thermostat, so NVT ensemble is closer to reality. Fortunately, the average values in the limit $N \rightarrow \infty$ for both ensembles coincide, however, the higher moments and in particular fluctuations for different ensembles are different [28].

5.2 Helium heat capacity

The inert gases pair interaction potentials of are well known, for helium $\varepsilon = 0.000880$, $\alpha = 5.4$, $r_m = 2.869$ (Table 2.1.). The program has the ability to heat or cool the system, heat 58 He atoms at different speeds (*Helium58-2.mlc* file), approximate the dependence of kinetic energy on time (the total energy will grow at the same rate) and the dependence of temperature on time by linear functions of the form $E(t) = E_0 + K_E t$, $T(t) = T_0 + K_T t$ then, given that the volume of the system is fixed

$$C_V = \left(\frac{\partial E}{\partial T} \right)_V = \frac{\partial E}{\partial t} / \frac{\partial T}{\partial t} = \frac{K_E}{K_T} \quad (5.1)$$

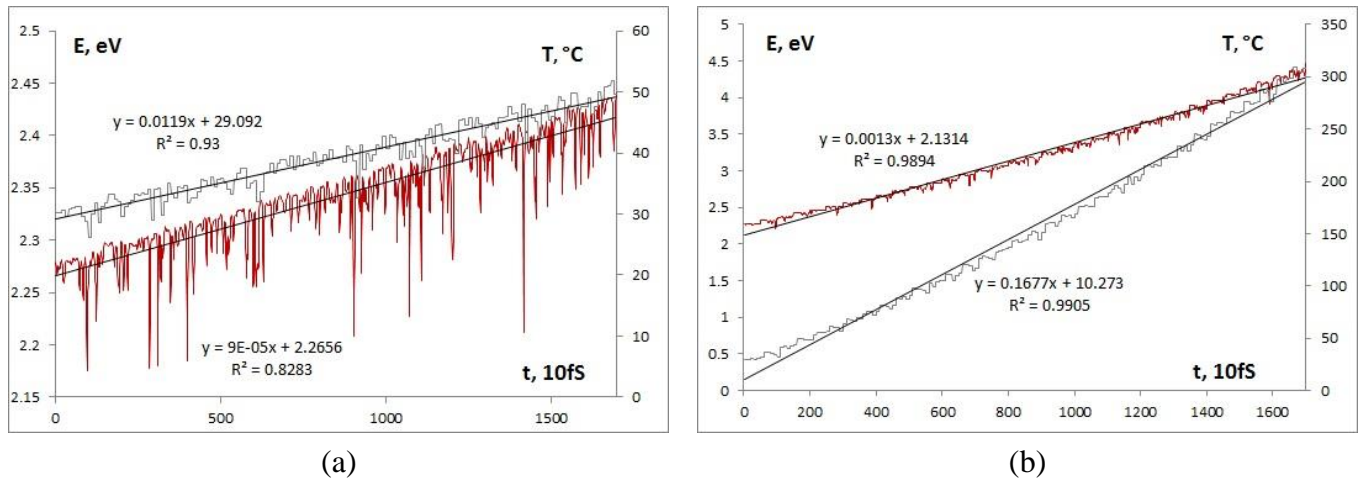


Fig. 5.2. Heating of 58 He atoms at different rate and approximation temperature grows.

To recalculate the heat capacity to standard units [$J \times \text{mol}^{-1} \times K^{-1}$], the energy in electron volts must be converted into Joules, divided by the number of particles in the simulated system N and multiplied by the Avogadro number

$$C_V = \left(\frac{K_E}{K_T} \frac{N_A}{N} \times 1.6021 \times 10^{-19} \right) \frac{J}{K \times \text{mol}} = 12.581 \quad (5.2)$$

The value given in (5.2) corresponds to the graph of Fig. 5.2 (a), for modeling Fig. 5.2 (b) we have $C_V = 12.895$.

The experimental value [20]

$$C_V = 1.51 \times R = 12.555 \text{ J} \times \text{mol}^{-1} \times \text{K}^{-1} \quad (5.3)$$

The first case, the relative deviation is 0.2%, in the second 2.7%.

The above definition of isochoric heat capacity is a solution to the problem "head-on". To find the heat capacity usually use the ratio [20]

$$\langle \delta H^2 \rangle_{NVT} = k_b T^2 C_V, \quad (5.4)$$

where H is the Hamiltonian of the particle system. The relation (5.4) is valid only for the canonical ensemble. The issue of recalculation of fluctuations during the transition from the canonical to the microcanonical ensemble was solved in [20]. The formula for calculating the heat capacity is given from [25, formula (2.82)]

$$C_V = \frac{3}{2} N k_b \left(1 - \frac{2 \langle \delta E_k^2 \rangle_{NVE}}{3 N k_b^2 T^2} \right)^{-1} \frac{N_A}{N} \quad (5.5)$$

Formula (5.5) is left in the form in which it is given in [25], a multiplier is added to the right, bringing the heat capacity to the molar.

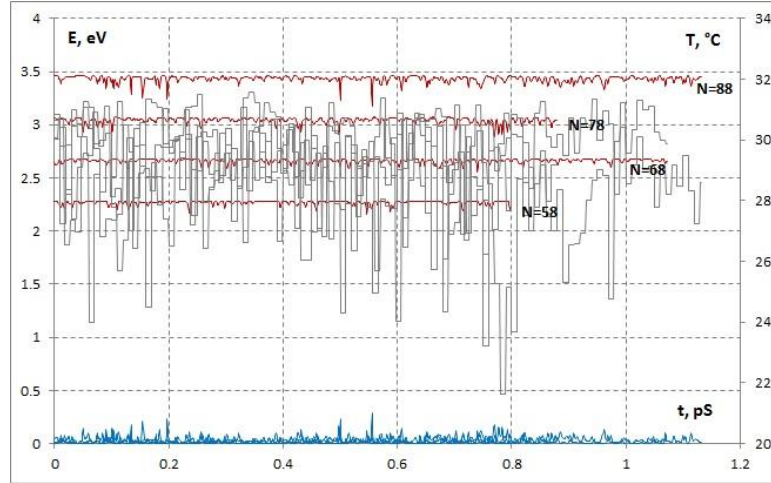


Fig. 5.3 The kinetic energy fluctuations (red), potential energy (blue), temperature (gray) for 58, 68, 78, 88 He atoms.

The energy fluctuations are expressed in electron volts, which must be converted to Joules

$$C_V = \frac{3}{2} N k_b \left(1 - \frac{2 \langle \delta E_k^2 \rangle_{NVE} \times (1.6021 \times 10^{-19})^2}{3 N k_b^2 T^2} \right)^{-1} \frac{N_A}{N}$$

The calculation results of the isochoric heat capacity for an ensemble of N He atoms are shown in Table 5.1.

Table 5.1

N	$\langle \delta E_k^2 \rangle_{NVE}$, (eV)	C_V , ($\text{J} \times \text{mol}^{-1} \times \text{K}^{-1}$)	Error, (%)
58	0.000324207	12.540	-0.1
68	0.000354571	12.536	-0.2
78	0.000710230	12.585	0.2
88	0.000813129	12.586	0.2

So, the isochoric heat capacity C_V was calculated via microscopic model of the molecule coincides with the experimental value (5.3) with an error at the level of tenths of a percent.

6 Conclusion

The given results shows that the program correctly reproduces the heat capacity of inert gases, which means that all the calculation schemes implemented in the program code do not contain errors.

The next step would be to see to practically significant tasks application. However, most of them are cross-disciplinary and involves the interaction of specialists with different areas of specialization.

The work was done exclusively on enthusiasm, by one person, so that all the code placed in one head, therefore, it will not be difficult to modify the program for a close engineering task.

The author is open to any questions concerning the application of the program code.

andreyvn68@mail.ru

<https://www.researchgate.net/profile/Andrey-Nazarkin>

Nazarkin Andrey

30.03.2023

References

1. *William H. Elliott, Daphne C. Elliott* **Biochemistry and Molecular Biology**. Oxford University Press; 4th edition March 25, 2009.
2. *Yu.G. Pryadko V.G. Kravaeva* **Theoretical mechanics: Geometry of masses**. Chelyabinsk South Ural State University, 2002.
3. *Malyshev A.I., Maksimova G.M.* **Fundamentals of vector and tensor analysis for physicists**. Moscow, Nizhny Novgorod, 2012.
4. *Ilya G. Kaplan* **Intermolecular Interaction**. John Wiley & Sons. Ltd. 2006.
5. *Gevorgyan A.A., Maleev A.V., Potekhin K.A.* **The use of the model of atom-atomic potentials in the study of the features of the interaction of molecules in crystals**.
6. *A.S. Fedorov, P.B. Sorokin, P.V. Avramov, S.G. Ovchinnikov* **Modeling of properties, electronic structure of a number of carbon and non-carbon nanoclusters and their interaction with light elements**. Novosibirsk, Publishing house of SB RAS, 2006
7. *G.V. Dedkov* **Effective rules for combining interatomic interaction potentials and forces**. JETP Letters, 1997, Vol. 23, №16
8. *J. Willard Gibbs* **Elementary Principles in Statistical Mechanics**. Charles Scribner's Sons, 1902
9. *V.F. Zhuravlev* **Foundations of Theoretical Mechanics**. M: FIZMATLIT, 2001.
10. *H. Goldstein* **Classical Mechanics**. Addison-Wesley, Reading, 2nd edition, 1980.
11. *V.N. Branets I.P. Shmyglevsky* **Application of quaternions in problems of orientation of a rigid body**. M. Nauka, 1973
12. *Golubev Yu.F.* **Algebra of quaternions in kinematics of a solid body**. Preprints of M.V. Keldysh IPM 2013.
13. *A.A. Yablonsky* **Course of theoretical mechanics. Part II Dynamics**. M. HSE, 1966.
14. *P. Appell* **Theoretical Mechanics**. M. FIZMATLIT, 1960
15. *F.L. Chernousko, L.D. Akulenko, D.D. Leshchenko* **Evolution of solid body motions relative to the center of mass**. M. Izhevsk, Izhevsk Institute of Computer Research, 2015.
16. *E. A. Balakina, E. B. Kuznetsov* **On numerical integration of kinematic Euler equations**. J. Comput. Math. Math. Phys., 41:11 (2001), 1623–1629
17. *E. B. Kuznetsov* **On one approach to the integration of Euler's kinematic equations**. J. Comput. Math. Math. Phys., 38:11 (1998), 1732–1738
18. *V.P. Krainov* **Qualitative Methods of Physical Kinetics and Hydrodynamics**. American Institute of Physics, 1992
19. *Morozov A.E.* **Introduction to plasmodynamics**. M. FIZMATLIT, 2006

20. *E. A. Volkov* **Numerical Methods**. Taylor & Francis, 1990
21. *Joseph O'Rourke* **Computational Geometry In C**. Cambridge University Press, 1998
22. *Lindsay R.B.* **Introduction to physical statistics**. John Wiley & Sons, 1941
23. *Yiannis N. Kaznessis* **Statistical Thermodynamics and Stochastic Kinetics**. Cambridge, 2012
24. *Irina V.A., Silaev P.K.* **Numerical methods for the physicist**. Moscow-Izhevsk: Institute of Computer Research, 2004 (B II TOMAX)
25. *G. Hall, J. M. Watt* **Modern Numerical Methods of Ordinary Differential Equations**. Oxford University Press, 1975
26. *Giovanni Bussi* **A Fully Anisotropic Formulation of Stochastic Cell Rescaling**. Academic Year 2020/2021
27. *Susanne Pfalzner, Thomas K. Henning* **N-Particle-Simulations of Dust Growth**. Icarus 141, 388–398 (1999).
28. *Allen M.P. Tildesley D.J.* **Computer Simulation of Liquids**. Clarendon Press Oxford, 1991
29. *A. I. Anselm,* **Bases of Statistical Physics and Thermodynamics**. M. Nauka, 1973.
30. *L.D. Landau, E.M. Lifshitz* **Statistical Physics**. vol. 5 Elsevier, 2013.
31. *J. Z. Lebowitz, J. K. Percus, L. Verlet* **Ensemble Dependence of Fluctuation with Application to Machine Computation** Phys. Rev. Vol.153, 1, 1967.
32. *Vargaftik N.B., et all* **Handbook of thermal conductivity of liquids and gases**. Boca Raton: CRC Press, 1994.