

Program aztotMD
(PC version)

Manual

Computer program azTotMD.exe (in the following called **The Program**) is intended for simulation by classical molecular dynamics (MD) method and by non-constant force field molecular dynamics.

System requirements

The Program can work under operating systems of Windows™ family: Windows XP, Windows 7, Windows 10.

1 The Program capabilities

The Program has been developed for simulation of different atomistic systems from molecular liquids to ionic crystals. The simulation can be performed in microcanonical (NVE) or canonical (NVT) ensembles in boxes with rectangular periodic conditions. Nosé-Hoover thermostat is used for temperature control in NVT case.

1.1 Interactions in the system (a force field)

Intermolecular interactions can be specified by short-range pair potentials (in the following called **pair potentials**) and by Coulombic forces. A pair potential is interaction energy of two particles of specified types as a function of distance. For example, it is possible to define a pair potential between H^+ and O^{2-} ions, and for each such ion pair energy of interaction will be calculated. Thus, pair potentials act on **every** specified pair. Coulombic interaction is set automatically after defined nonzero ion charges. For Coulombic forces and energy calculation a value of permittivity must be specified. The Ewald summation technique is applied for Coulombic interaction calculations.

Intramolecular interactions can be specified by “bonded pair interactions” (in the following called **bonds**). Bonds act between concrete (specified by the number not a kind) particles. For example, one can specify bonds between H^+ and O^{2-} ions in the same water molecule. Then ions in the same molecule will interact via bonds and ions belonging to different water molecules will interact via pair potentials. Like pair potentials, a bond is also interacting

energy as a function of a distance between two particles. The Program supports also valence angle potentials which depend on coordinates of three particles.

1.1.1 Pair potentials

The Program supports the next functional forms of pair potentials (energy, U , as a function of distance between interacting particles, r):

Lennard-Jones potential,

$$U = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \text{ parameters: } \varepsilon \text{ and } \sigma;$$

Buckingham potential,

$$U = A \exp(-r/\rho) - \frac{C}{r^6}, \text{ parameters: } A, \rho \text{ and } C;$$

Born-Mayer-Huggins potential,

$$U = A \exp[B(\sigma - r)] - \frac{C}{r^6} - \frac{D}{r^8}, \text{ parameters: } A, B, \sigma, C \text{ and } D;$$

potential from Stafford et al. paper [Staff] for molten halides (“746” potential):

$$U = \frac{A}{r^7} - \frac{B}{r^4} - \frac{C}{r^6}, \text{ parameters: } A, B \text{ and } C.$$

Parameters of pair potentials must be specified in a file **field.txt** (see section 2.2 «[field.txt](#)») in the above order. All parameter values mean in angstroms, electron-volts and derived units.

1.1.2 Intramolecular interactions (bonds)

As pair potentials, bonds are also energy (U) of two bonded particles as a function of distance (r) between them. At present version The Program supports harmonic and Morse functional form of bonds:

$$U = k(r - r_0)^2, k \text{ and } r_0 \text{ are parameters.}$$

$$U = D(1 - \exp[-\alpha(r - r_0)])^2 + C, \text{ parameters: } D, \alpha, r_0 \text{ and } C.$$

Parameters for different bonds are kept in a file **field.txt** (see section 2.2 «[field.txt](#)»), but a list of bonded atoms is presented by a **bonds.txt** file. For comfort, The Program can automatically create bonds based on interatomic distances with considering periodic conditions. This saves users from necessity of manual specifying all bonds in a simulated system, see section 2.2 «[field.txt](#)».

1.1.3 Valence angles

At present version The Program supports harmonic cosine functional form of valence angle potential:

$U = k(\cos \theta - \cos \theta_0)^2$, k – is a spring constant and θ_0 is an equilibrium angle (user must specify $\cos \theta_0$ as a parameter value).

Parameters for different valence angles are kept in a file **field.txt** (see section 2.2 «**field.txt**»), but a list of atoms constitute valence angles is presented by a **angles.txt** file.

1.1.4 External electric field

The Program can simulate evolution of a system under external electric field. In this case an additional force acts on charged particles and a projection of this force onto x-axis (F_x) is equal to:

$$F_x = -q \frac{\Delta E_x}{a},$$

and energy (U) of a charged particle in this external field is equal to:

$$U = qx \frac{\Delta E_x}{a}.$$

where ΔE_x is a voltage drop along x-axis, a is a box size in a x-direction, x - x-coordinate of a particle and q is its charge. The voltage drop assumed to be linear along x-axis.

1.2 Non-constant force field molecular dynamics

Non-constant force field molecular dynamics (NCFFMD) in The Program expresses by two possibilities: an electron transfer between particles and bonds recombination.

An electron transfer is implemented by algorithms presented in [JCC]. During an electron transfer a particle-donor give an electron to a particle-acceptor. As a result the donor transforms into own oxidized form and the acceptor – into reduced one. User directives set which particles can be donors or acceptors, their oxidized and reduced forms and maximal distance of an electron transfer (see section 2 «**Input files**»).

Bonds recombination is breaking old bonds and formation new ones. Bonds are automatically broken then their lengths achieve a critical value specified by a user. New bonds form then energy of a system before and after bond creation is equal. The Program can change particles kind after bond broking or creation. All corresponding directives are specified by a user in a file **field.txt** (see section 2 «**Input files**»).

1.3 Simulation results

As a result of The Program work one can obtain structural information: a system configuration (coordinates of all particles) and radial distribution functions (RDF). One can calculate coordination number (ligancy) by integration of RDF:

$$n_{AB} = 4\pi \frac{N_B}{V} \int_{r_1}^{r_2} g(r) r^2 dr,$$

where n_{AB} is a coordination number of A particle by B particles, N_B is a total number of B particles in a system, V is a volume of a system, $g(r)$ is a radial distribution function of A-B pair, r_1 and r_2 are distances interval for the coordination number determination, usually, they are points of minimums, surrounding of the first maximum on RDF curve.

Also one can obtain dynamical properties of a system: mean square displacement (MSD) for every kind of particles and a number of particles (and electrons in the case of electron transfer option) passed through the Oyz edge of a simulation box. Self-diffusion coefficients can be calculated by a slop of MSD dependences on time, according to the Einstein relation:

$$D = \frac{1}{2d} \frac{d \langle |r(t) - r(t_0)|^2 \rangle}{dt},$$

where D is a self-diffusion coefficient, d is a dimensionality of a system (3 for 3d space), the second multiplier is a time derivative of MSD (just a slop of MSD as a function of time in the case of its linearity).

Moreover, one can obtain a system statistic: temperature, total energy and etc. A list of output files with abovementioned information is given in [section 3 «Output files»](#). In output files all length values are give in angstroms (Å, 10^{-10} m), time – in picoseconds (ps, 10^{-12} s), energy – in electron-volts (eV, 1.6×10^{-19} J), temperature – in kelvins (K).

2 Input files

The Program requires three mandatory text-files in own directory: *control.txt*, *field.txt* and *atoms.xyz*. For bond and valence angle predifinition it is possible to use two text-files: *bonds.txt* and *angles.txt*. All these files can be created and edited in any text editor.

2.1 control.txt

This file contains main program directives: a timestep value, an ensemble type, temperature, a thermostat constant, directives for the Ewald summation, electron transfer directives and etc. Parameters must be specified after a directive keyword through space(s) or(and) tabulation(s). A full list of directives is given in the following table (symbols N and f mean an integer or a fractional number, correspondingly; 1/0 means unit or zero):

directive	meaning	comment
timestep f	set a value of timestep equal to f picosecond (ps)	a typical value is about 10^{-3} – 10^{-4} ps.
nstep N	set a number of steps (cycles) of MD equal to N	
nequil N	set a number of steps for initial relaxation of a system as N	
temperature f	set temperature equal to f kelvin's degree	this parameter is ignored in NVE ensemble (directive NVT set as zero)
permittivity f	set relative permittivity equal to f	typically, 1.0. This parameter has meaning only if a system contains charged particles.
alpha f	parameter for Ewald summation	This parameter has meaning only if a system contains charged particles.
kx N	A number of k-vectors along x-axis	
ky N	A number of k-vectors along y-axis	
kz N	A number of k-vectors along z-axis	
nvt 1/0	1 – for NVT ensemble, 0 – for NVE	
tautemp f	set thermostat relaxation time equal to f ps	has meaning only for NVT simulation
gaussian 1/0	set initial velocities according to Gaussian distribution	
maxRDF f	maximal distance for radial distribution function calculation (in angstroms)	
dRDF f	set radial distribution step width as f angstroms	recommended value 0.05
frRDF N	collect statistic for calculation of radial distribution function every N timesteps	
eJump N f eq/min/metr f_l	use electron transfer: positive N : how many times per timestep negative N : every N^{th} timestep f – maximal hopping distance	examples: eJump -5 3.5 metr eJump 1 3.5 eq 1e-6

	Hopping conditions: eq – energy equality; min – with energy decreasing; metr – according to Metropolis rule. f_l – permissible energy uncertainty (for equality)	
Ux f	set electric voltage drop along Ox axis as f volt	for simulation of a system behavior in external electric field, for example, for electrical conductivity study
hist N	output history.dat file every N timesteps	
revcon N	output instant configurations every N timesteps	
stat N	output stat.dat file every N timesteps.	

The Program ignores other words and they can be used as comments. Example of **control.txt** file:

```

timestep 0.0005 ps
nstep 6000
nequil 1000
eqfreq 10000
temperature 833.0

permittivity 1.0

// EWALDS:
alpha 0.5
kx 3
ky 15
kz 15

// THERMOSTAT
nvt 1
tautemp 0.5
gaussian 0

//RDF OUTPUT
maxRDF 10.0 // maximum RDF distance output
dRDF 0.05 // width of RDF step
frRDF 50

// e transfer
eJump -5 3.5 eq 5e-5

// external field
Ux 0

//OUTPUT
hist 200
revcon 2000
stat 100

```

2.2 field.txt

This file contains all information about particle's kinds and their interactions. The structure of this file is strongly specified: at first, it is necessary to point the number of particle's types in a system (line *spec N* where N is the number of species), then describe each of them in separate lines. The line consists of a name of a particle's type (not more than 7 characters, using Latin letters, numbers, minus or underscore), its mass (in a.m.u.), its charge (in *e*), its ability to be an electron donor (1 – yes, 0 – no), its ability to be an electron acceptor (1 – yes, 0 – no) and its own energy (in eV). The last three properties are necessary for non-constant force field MD (electron transfer), for classical MD they may be pointed at random.

After specifying all particle's types **red-ox** keyword has to be written with the number of oxidant and reductant types (can be equal to zero). Then all particle's types which can be an electron donor or acceptor must be listed. Each line consists of name of particle's type, name of its oxidized form and then name of its reduced form. In the case of absence a keyword **null** must be pointed. This section is obligatory only for the electron transfer routine, otherwise it is enough to write **red-ox 0**.

The next is a section with short-range pair interactions (pair potentials). This section starts with a keyword **vdw** and the number of pair potentials in a system. Then every pair potential is described in a separate line. At first, names of interacted particles are pointed and then a keyword which determines a functional form of the pair potential. The Program supports the next pair potentials (a keyword – a pair potential):

- **lnjs** – Lennard-Jones pair potential;
- **p746** – potential from Stafford work [Staff];
- **bmhs** – Born-Mayer-Huggins pair potential;
- **bkhm** – Buckingham pair potential.

After specifying of the pair potential type, one needs to point a cutoff radius (The Program supports different cutoffs for different potentials) and parameters of the pair potential (see Section 1.1.1 «Pair potentials»). All parameter values mean in angstroms, electron-volts and derived units.

The next section describes intramolecular (bonded) interactions. It starts from **bonds** keyword and the number of bond's types. Then every bond's type is described in a separate line. At first, one need to specify a serial number (from 1, without a pass), then names of a bonded particles pair and possibility of the bond breaking (0 – can not break, 1 – can). If one points 1, one has to point a distance of breaking (in angstroms) and particle's types, in which initial particles transform after the bond breaking. Particle's types after breaking must be specified in the same

order as the initial types. Then keyword of functional form for bonded interaction must be pointed (harmonic potential - **harm** keyword, Morse potential – **mors** keyword) and parameters of this bonded interaction (see Section 1.1.2 «Intramolecular interactions (bonds)»). All parameter values mean in angstroms, electron-volts and derived units.

Keyword **autobonding** allows to set initial bonded interaction in a system automatically. After **autobonding** keyword one must specify the number of particle's pair types for which this routine is applied an maximal distance (in Å) for automatically bonding of particles.

The next section describes possibility of new bonds formation and starts from **linkage** keyword and the number of particle's types which can form bonds. Advice: it is enough to specify only one particle types from a pair capable of bonding. It is better to specify more rare type from the pair to optimize The Program calculations. Then type of particle capable of bonding, the number of particle types with which the bond can be formed and then pair particle type – bond index (from **bonds** section) must be listed in separate lines. All bonds in **autobonding** and **linkage** sections must be previously defined in **bonds** section. Bonds from **linkage** section must possess ability of breaking (the value of the flag is equal to 1).

The section starting from **angles N** keyword describes valence angle potentials. Then every potential is described in a separate line. At first, one need to specify a serial number (from 1, without a pass), then the name of central atom constitutes the valence angle. Then keyword of functional form for valence angle must be pointed (harmonic cosine potential - **hcos** keyword) and parameters of this bonded interaction (see Section 1.1.3 «Valence angles»).

If simulation includes a creation of new bonds, it is possible to create new valence angles too. To do this one needs to declare **angle-forming** section. It starts from **angle-forming** keyword and the number of species which can form valence angle after their creation. Then each line contains the name of such species and valence angle identifier (specified in **angles** section).

Directives **angle_list** and **bond_list** force The Program to read lists of valence angles and bonds from files **angles.txt** and **bonds.txt**, correspondingly. These files start from the number of angles(bonds) and then each triplet(pair) of atoms and potential type identifier is listed. Note, that for valence angle the first number – is the number of central atom in angle.

Example of **field.txt** file:

```
spec 3
Cu    63.546    0.5    1    0    1.1
Cu2   63.546   1.002  0    1    1.1
Cl-   35.0     -0.5    1    0    0.0
red-ox 2
Cu    Cu2    null
Cu2   null   Cu
vdw  4
Cu    Cu     p746    8.0    0.055977    0.0    0.0
```



```

Cl- Cu p746 8.0 61.76188 6.230535 0.0
Cl- Cl- p746 8.0 1908.315 12.46107 83.0735
Cu2 Cl- bmhs 8.0 1.82919 2.95187 1.81622 202.2764
-473.94615
bonds 3
1 Cu Cl- 1 2.7 Cu Cl- harm 10.5 1.0
2 Cl- Cl- 1 5.0 Cl- Cl- harm 1.5 3.0
3 Cu2 Cl- 1 3.0 Cu2 Cl- harm 5 2.2

bond_list 1

angles 1
1 O(c) hcos 200.0 -0.85081

angle_list 1

linkage 2
Si+ 1 O(t) 1
B+ 1 O(t) 3
angle_forming 1
O(c) 1

```

2.3 atoms.xyz

This file contains initial coordinates of all particles in a system. The file structure must correspond to *.xyz format (https://en.wikipedia.org/wiki/XYZ_file_format), which is supported by many molecular editors, for example, Diamond software (<http://www.crystalimpact.com/diamond/Default.htm>). Also, there are some free editors, for example, Avogadro (<http://avogadro.cc/>). Despite *.xyz extension, the file is text, in fact, and can be edited by any text editor. The first line of the file contains the number of particles, the second – box parameters (the first number is 1, then 3 numbers: sizes in X, Y and Z directions). Note, in original *.xyz format the second line is reserved for a comment and ignored by most part of molecular editors, but The Program reads box parameters from it. The next lines contain names of particles and their Cartesian coordinates in angstroms. Note, it is not recommended to use + symbol in particles names. Names of particles from this file must be presented in **field.txt** file. All coordinates must be positive and inside of the box.

An example of **atoms.xyz** file (only several first lines are pointed):

```

400
1 5.950000 40.103720 40.103720
Cu 0.584874 7.034183 0.128431
Cu2 4.268840 9.420375 0.094972
Cu2 4.911667 24.926207 39.798957
Cu2 1.022435 15.745150 38.199232
Cu2 0.799887 30.615409 5.461493
Cu 4.189181 32.015996 31.446103
Cl- 0.042140 0.821895 0.836985
Cl- 3.017035 2.470509 1.486856
Cl- 5.131349 10.105318 38.570523
Cl- 3.871618 21.783697 35.749538

```

3 Output files

3.1 rdf.dat

A textfile. It contains radial distribution function (RDF) for all particles pairs in a system. The first column contains a distance between particles of a pair in Å, the others – RDF values (dimensionless) for all pairs. For plotting of RDFs and their treatment one may open this file with any table editor, for example, Origin or MS Excel.

3.2 revcon.xyz

This file contains a final configuration of a system (particle's coordinates in angstroms). It is completely similar to **atoms.xyz** input file. One can rename this file to atoms.xyz to start simulation from coordinates obtained in the previous simulation. The file can be opened as any text editors as special programs for atoms structure visualization like Diamond or Avogadro software.

3.3 stat.dat

A text file. It contains statistic about system parameters: energy (in eV), temperature (K) and etc. For plotting and treatment of these data one may open this file with any table editor, for example, Origin or MS Excel.

3.4 msd.dat

A textfile. It contains mean square displacement (MSD) for every particle's type and the number of these particles passed through an Oyz edge of a box in positive and negative directions. For plotting and treatment of these data one may open this file with any table editor, for example, Origin or MS Excel.

3.5 jumps.dat

A textfile. It appears only in the case of application the electron transfer routine. The first five columns contain serial number of timestep, time (in ps), total amount of electron hops and amounts of hops through Oyz edge of a box in positive and negative directions. Other columns contain amount of electron hops between different donor-acceptor pairs. For plotting and treatment of these data one may open this file with any table editor, for example, Origin or MS Excel.

3.6 revbonds.txt

A textfile. This file contains a final list of bonds.

3.7 revangles.txt

A textfile. This file contains a final list of valence angles.

4 Warning and error messages

Some errors or warnings can appear during The Program run. The Program stops after an error appearance, an error message starts from ERROR word. In some cases, The Program detects some problem situation and informs about this but continues to work. Such messages start from WARNING word.

message	meaning	solutions
ERROR! Can't open file 'field.txt'	impossible to open field.txt file	verify, that field.txt file exists in The Program directory
ERROR: Unknown species in red-ox section:	there is unknown particle type in the red-ox section of field.txt file.	verify, that particle names in field.txt file corresponds to particles names in red-ox section of the same file
ERROR: Unknown OxForm in red-ox section:	unknown oxidized form in red-ox section	see above
ERROR: Unknown RedForm in red-ox section	unknown reduced form in red-ox section	see above
ERROR! Unknown atom type in vdw-line:	unknown particle type in vdw section	verify, that particle names in field.txt file corresponds to particles names in vdw section of the same file
ERROR! Unknown potential type in vdw-line	unknown type of pair potential in vdw section	correct pair potential keyword in vdw section of field.txt file. Possible variants: lnjs, bkhn, p746 and bmhs. <u>See section 1.1.1 «Pair Potentials»</u>
WARNING! no Van-der-Waals interaction	There are no short-range pair interactions in a system. This is not an error, but, as a rule, a MD simulation uses pair interactions	set pair potentials in vdw section of field.txt file
Can't open file 'atoms.xyz'	impossible to open atoms.xyz	verify, that atoms.xyz file

	file	exists in The Program directory
Error! Unknown box type	incorrect box type	at presence, only 1 value is supported (atoms.txt file, the first number of the second line). This value corresponds to a box with a rectangular geometry
ERROR! unknown atom type in atoms.xyz	there are particle name in atoms.xyz file which is not specified in field.txt file	verify, match of particle names in these two files. All particle names in atoms.xyz file must be specified in field.txt file
ERROR! Can't open file 'control.txt'	impossible to open control.txt file	verify, that control.txt file exists in The Program directory
ERROR: timestep must be declared in control.txt file!	integration timestep is not specified in control.txt file	point timestep f directive in control.txt , where f is a value of timestep in picoseconds.
ERROR!!!: no 'nstep' or 'timesim' directives in control.txt file!	the number of steps or simulation time are not specified in control.txt file	point nstep or timesim directives in control.txt file
WARNING: no 'nequil' or 'timeequil' directives in control.txt file - there is no equilibration period!	the number of equilibration steps or equilibration time are not specified in control.txt file. MD simulation will run without initial equilibration.	point nequil or timeequil directives in control.txt file
WARNING: temperature was not defined in control.txt file!	temperature is not specified in control.txt file.	set desired temperature with temperature directive for simulation in NVE ensemble
WARNING: alpha was not defined in control.txt file!	alpha parameter for Ewald summation is not specified in control.txt file.	set alpha parameter for a system with charged particles
WARNING: kx / ky / kz was	kx / ky / kz parameter for	set kx / ky / kz parameter for a

not defined in control.txt file!	Ewald summation is not specified in control.txt file.	system with charged particles
WARNING: permittivity was not defined in control.txt file. Set as 1.0!	permittivity is not specified in control.txt file, a default value of 1.0 is used.	set this value explicitly, if it is necessary
ERROR: thermostat relaxation time (tautemp) was not defined in control.txt file!	thermostat relaxation time is not specified in control.txt file for a constant temperature simulation	use tautemp directive to point relaxation time
WARNING: maxRDF was not defined in control.txt file!	a maximal distance for RDF plotting is not specified in control.txt file	use maxRDF directive to point a maximal distance for RDF plotting
ERROR: dRDF was not defined in control.txt file!	a distance step for RDF plotting is not specified in control.txt file	use dRDF directive to point a distance step for RDF plotting
ERROR: frRDF was not defined in control.txt file!	a frequency of RDF collection is not specified in control.txt file	use frRDF directive to point a frequency of RDF collection
WARNING: rElec was not defined in control.txt file!	a distance of electron transfer is not specified in control.txt file	use rElec directive to point a distance of electron transfer
ERROR: dEjump is not specified	an energy uncertainty for electron transfer is not specified in control.txt file	use dEjump directive to point an energy uncertainty for electron transfer
WARNING: maxAcc is not specified. Set to default value, 6	the maximal number of acceptors for one donor is not specified in control.txt file, the default value of 6 is used	use maxAcc directive to point the maximal number of acceptors for one donor, if it is necessary
WARNING: stat directive is not specified, default value of 1000 is used	a frequency of statistic collection is not specified in control.txt file, the default value of every 1000 timesteps is used	use stat directive to point a frequency of statistic collection, if it is necessary
ERROR nvtsacle is NAN	integration error	verify initial coordinates, it is

		possible that some particles are too close and obtain too much acceleration, verify pair potentials and integration timestep
WARNING: max acceptor reaches:	the maximal number of electron acceptors is reached.	increase the maximal number of electron acceptors by maxAcc directive (the default value is 6)
ERROR: Unknown species in bonds declaration	unknown particle type in the bonds section of field.txt file	verify, that particle names in field.txt file corresponds to particles names in bonds section of the same file
ERROR: Unknown potential type in bonds declaration	an unknown type of intramolecular interaction	at the moment only harmonic potential is supported, harm keyword
ERROR: Unknown species in autobonding declaration	an unknown particle type in autobonding section of field.txt file	verify, that particle names in field.txt file corresponds to particles names in autobonding section of the same file
WARNING: a maximal number of bonds exceeded	the maximal numbers of bonds is reached	
ERROR: Unknown species in linkage definition	an unknown particle type in linkage section of field.txt file	verify, that particle names in field.txt file corresponds to particles names in linkage section of the same file
ERROR: Unknown bond number in linkage definition!	unknown bond number in linkage section of field.txt file	verify, numbers of bonds in bonds section of field.txt file

References

[Staff] A.J. Stafford, M. Silbert, J. Trullas, A. Giro, Potentials and correlation functions for the copper halide and silver iodide melts: I. Static correlations. J. Phys.: Condens. Matter. 2, 6631-6641 (1990) <https://doi.org/10.1088/0953-8984/2/31/016>

[JCC] A.A. Raskovalov, A new extension of classical molecular dynamics: an electron transfer algorithm. J. Computational Chem, 38, 926-932 (2017) <http://dx.doi.org/10.1002/jcc.24755>