The Lucretius™ Molecular Dynamics (MD) simulation package is comprised of a number of individual source code files, all of which are necessary in order to compile the appropriate computer program necessary to perform accurate, atomistic simulations of chemically realistic systems.

The following files are necessary to compile the Lucretius™ program for MD computation:

|  |  |
| --- | --- |
| lucretius.f | Contains all principal logic for the MD simulation. |
| params.h | Contain the necessary system specific parameters for simulation .  (Generated by SystemGenerator™) |
| dimensions.h | Contains the dimensions of program data structures. |
| commons.h | Contains common block definitions for subroutine data transfer. |

Simulation of each particular system requires compilation of the Lucretius™ package for each specific system to be simulated with all four of the above files. Of the preceding files, only params.h is system specific.

In addition to the above files required to compile, four additional files are needed to simulate a system:

|  |  |
| --- | --- |
| ff.dat | Force field parameterization for the simulated system. |
| coords.inp | Initial system configuration. |
| connectivity.dat | Molecular connectivity data for the simulated system. |
| mdrun.params | Runtime variables for control of simulated system. |

For each of the above input files, a line which begins with an \* (asterisk) is considered to be a comment line, and is ignored by the parser used by Lucretius™.

Following are in-depth descriptions for the system specific files which are employed in the Lucretius™ simulation package.mdrun.params

The mdrun.params file is used to set all of the possible run-time control variables which are used to control the manner in which Lucretius™ performs the specified simulation. System state variables, simulation ensembles, and global interaction controls are set via this file.

Line 1

**\* ---- Ensemble control variables**

\* NVE (microcanonical) NVT(canonical) NPT(isothermal-isobaric)

*.false. .false. .true.*

*NVE* : Boolean

*NVT* : Boolean

*NPT* : Boolean

Select the ensemble under which the simulation will run: microcanonical (NVE), canonical (NVT), or isothermal-isobaric (NPT). Only one value should be set to *true*.

Line 2

**\* ---- Dispersive interaction and integration control**

\* Cutoff Rad. Max "Drift" Delta Number of steps

*10.5 1.0 0.5 2000000*

*Cutoff Radius (Rc)* : Double precision

*Max Drift (DMax)* : Double precision

*Delta ()* : Double precision

*Number of steps (n)* : Integer

The *cutoff radius* represents the distance (in Å) at which the dispersion interactions are truncated to zero. The appropriate pressure correction term due to this truncation is included within the simulation. Corrections to the energy due to truncation are only included in the finished statistics of the simulation, but are not utilized internally in energy calculations during the simulation run.

The *max drift* represents the cumulative maximum displacement (in Å) since the previous neighbor list calculation allowed before a new neighbor list calculation is required.

The overall length of the neighbor list is therefore directly proportional to *R­c+DMax*.

*Delta* represents the shortest integration time step within the system.

The *number of steps* is the total number of integration time steps for the simulation run.

For a multiple timestep integrator, *delta* (D) represents the time step of the short time processes, while the *number of steps* refers to the total number of full (multi-)timestep cycles performed. Thus, the total time of the run is *t=SinD* where *Sl* is the number of small time steps which comprise a single long time step (see line 12), and *n* is the *number of steps* from the mdrun.params input file.

Line 3

**\* ---- Thermostat and Barostat**

\* Temperature(K) Pressure (atm) Num\_NHC OmegaQ (1/fs) OmegaW (1/fs)

*298.00 1.00 1 0.01 0.0005*

*Temperature (T)* : Double precision

*Pressure (P) :* Double precision

*Num\_NHC (NNHC) :* Integer

*OmegaQ (wQ) :* Double precision

*OmegaW (wW)* : Double precision

*Temperature* is the nominal system temperature in Kelvin. For the NVT and NPT ensembles, this is the temperature maintained by the thermostat, while for the isokinetic ensemble this is the value at which the overall temperature of the system is fixed via velocity rescaling (see Line 8).

*Pressure* is the nominal system pressure, in atmospheres. For the NPT ensemble, this is the pressure maintained by the barostat.

*Num\_NHC* is the number of Nosé-Hoover chains to use in the thermostat calculation. A value of 1 here represents the classic Nosé-Hoover thermostat, while a value greater than one is used to implement the Nosé-Hoover Chains methodology of Martyna et. al.[[1]](#endnote--1) which gives improved approach to the canonical ensemble for small and/or stiff systems.

*OmegaQ* () represents the frequency of the Nosé-Hoover thermostat mass. The related mass is where N is the number of degrees of freedom, R the universal gas constant, and T the nominal temperature of the system.



*OmegaW* () represents the frequency of the Anderson-Hoover barostat mass. The related mass is , with N, R, and T having the same meaning as for the thermostat.



Line 4

**\* ---- Statistic collection and calculation control**

\* Initialize(fs) Average(fs) Pressure(fs) Energy Split(fs) Coord Output(fs)

*30.0 150.0 3.0 1000000.0 1000.0*

*Initialize* : Float

*Average*  : Float

*Pressure* : Float

*Energy Split*  : Float

*Coord Output* : Float

*Initialize* represents the time (in femtoseconds) at the beginning of the run during which simulation is performed but no statistics are collected. This equilibration time is not calculated independently of the total simulation time; thus if one wishes 1ns of simulation time with a 100ps initialization, one should set the number of steps to account for a total of 1.1ns of total time.

*Average* represents the time window (in femtoseconds) over which quantities are averaged to generate the *fort.65* statistics tracking file.

*Pressure* is the time window (in femtoseconds) between calculation of the pressure in the simulation system.

*Energy Split* determines the time window (in femtoseconds) for calculation of in-depth energy contribution statistics for both intra- and intermolecular contributions to the overall system energy. Since this subroutine recalculates the total energy of the system, which is the single most expensive calculation in the simulation loop, it should generally be used only for debugging as needed.

*Coord Output* dictates the timespan (in femtoseconds) between output of the instantaneous coordinates and velocities of the atoms in the system. This output file contains the necessary state information to restart a run from the output state, and is output in the file coords.out.

Lines 5 and 6

**\* ---- Trajectory file settings**

***\* ---- Trajectory Output Flags (Filenames: fort.XX)***

\* Box Dipole(76) Coordinates(77) Stress Tensor(78) Velocities(79)

*.false. .true. .false. .false.*

\*

***\* ---- Trajectory Output Frequency (femtoseconds)***

\* Dipole Time Coordinate Time Stress Time Velocity Time

*100 1000 9 1000*

*Box Dipole* : Boolean

*Coordinates* : Boolean

*Stress Tensor* : Boolean

*Velocities* : Boolean

*Dipole Time*  : Float

*Coordinate Time*  : Float

*Stress Time*  : Float

*Velocity Time*  : Float

The *Box Dipole* flag toggles output of the aggregate sum over the dipoles of each individual molecule in the simulated system, yielding the total average *instantaneous* molecular dipole of the system at the current time, with the *Dipole Time* variable controlling the frequency (in femtoseconds) of this output.

The *Coordinates* flag toggles output of the instantaneous position vector for all atoms within the system, while the *Coordinate Time* variable controls the frequency (in femtoseconds) of this output.

The *Stress Tensor* flag toggles output of the instantaneous stress tensor for the total system, while the *Stress Time* variable controls the frequency (in femtoseconds) of this output.

The *Velocities* flag toggles output of the instantaneous velocity vector for all atoms within the system, while the *Velocity Time* variable controls the frequency (in femtoseconds) of this output.

All trajectory files are output as binary, unformatted files in order to reduce space. The filenames are named, generically, fort.XX, where the XX suffix is that listed after the names of the flag variables (e.g. the Coordinates file is output as fort.77).

Line 7

**\* ---- System Adjustment and Debugging**

\* Resize Box? Target Box Size Output Additional Information?

*.false. 35.74 .false.*

*Resize Box* : Boolean

*Target Box Size* : Double precision

*Output Additional Information* : Boolean

The *Resize Box* flag determines whether or not the system should be shrunk or enlarged to the *Target Box Size*. If *Resize Box* is true, the system will undergo an **affine** transformation of the system boundary **and all the atomic positions within the system** over the entire simulation time dictated by the integration control variables. It is therefore important to ensure that the system changes at a reasonable time rate to ensure proper maintenance of molecular connectivity.

The *Output Additional Information* flag toggles the printout of a brief set of statistics for every outer integration step to **stdout**. The included statistics are the current outer integration step, box size, pressure, temperature, and total energy of the system.

Line 8

**\* ---- Interaction Control**

\* Calc Intermolecular? Exclude 1-4? Isokinetic Ensemble?

*.true. .false. .false.*

*Calc Intermolecular* : Boolean

*Exclude 1-4* : Boolean

*Isokinetic Ensemble* : Boolean

The *calc intermolecular* flag toggles whether or not intermolecular interactions are calculated for the simulation. Setting this flag to *.false.* disables **all** intermolecular interactions within the system.

The *exclude 1-4* flag toggles whether or not repulsion/dispersion interactions are calculated between dihedral atoms separated by 3 bonds.

The *isokinetic ensemble* flag determines whether or not the overall temperature of the system is constrained by rescaling the momentum of the overall system to reproduce the nominal temperature at every integration step. If set to true, the temperature is rescaled to the value set in Line 3.

Line 9

**\* ---- 1-4 Interaction Scaling**

\* Reduce 1-4(Exc. mu-mu) Red. 1-4 Factor Reduce Q-mu 1-4 Red. Q-mu Factor

*.false. 0.0 .true. 0.8*

*Reduce 1-4 (Exc. mu-mu)* : Boolean

*Red. 1-4 Factor* : Double precision (0.0-1.0)

*Reduce Q-mu* 1-4 : Boolean

*Red. Q-mu Factor* : Double precision (0.0-1.0)

The *Reduce 1-4 (Exc. mu-mu)*  flag determines whether or not contributions to the non-bonded energy due to dihedral pair interactions are reduced for the dispersion, static charge-static charge, and static charge-induced dipole term contributions to energies between atoms attached by 3 intermediate bonds (torsionally linked atoms). The induced dipole-induced dipole terms are excluded from reduction.

The *Red. 1-4 Factor* determines the amount of reduction applied to the torsionally linked atoms if the *Reduce 1-4* *(Exc. mu-mu)* flag is set to true. The factor should be between 0.0 and 1.0, inclusive, and determines the amount of reduction applied. (e.g. 0.80 corresponds to an 80% reduction)

The *Reduce Q-mu 1-4* flag determines whether only the static charge-induced dipole contribution term of the dihedral pair interactions is reduced.

The *Red. Q-mu Factor* value determines the amount of reduction applied to the torsionally linked atoms if the *Reduce Q-mu 1-4* flag is set to true. This factor should be between 0.0 and 1.0, inclusive, and determines the amount of reduction applied. (e.g. 0.80 corresponds to an 80% reduction)

The *Reduce 1-4* *(Exc. mu-mu)* flag and *Reduce Q-mu 1-4* flags are not mutually independent. Specifically, if *Reduce 1-4 (Exc. mu-mu)* is set to true, then, in addition to reduction of repulsion/dispersion terms for the 1-4 atoms, the staic charge-static charge and static charge-induced dipole interactions will be reduced by *Red 1-4 Factor* as well. If *Reduce Q-mu 1-4* is set, the repulsion/dispersion terms will not be reduced, but the static charge-induced dipole interaction will be. If both are set, then the static charge-induced dipole contribution will be reduced first by the *Red. 1-4 Factor*, and the subsequent reduced value will be reduced again by *Red. Q-mu Factor*. In general this is probably not the desired outcome, so only one of these flags should be set at a time.

Line 10

**\* ---- Bond constraints (SHAKE Algorithm)**

\* Constrain Bonds Tolerance (SHAKE)

*.true. 1.d-14*

*Constrain bonds* : Boolean

*Tolerance* : Double precision

The *Constrain Bonds* flag determines whether or not the SHAKE algorithm is used to rigidly constrain all bonds to the rest value provided in the *ff.dat* file.

The *Tolerance (SHAKE)* parameter is used to dictate the maximum average squared bond length differential which indicates convergence of the SHAKE algorithm and subsequent acceptance of the constrained atomic positions.

Line 11

**\* ---- Long range electrostatic calculation (Ewald sum)**

\* Electrostatic Calculation Type Alpha Maximum K Vector

*Ewald 8.0 5*

*Electrostatic Calculation Type* : Character\*12

*Alpha* (a) : Double precision

*Maximum K Vector (kmax)*  : Integer

The *Electrostatic Calculation Type* is a non-control label which identifies the type of electrostatic calculation used in the integrator. The type actually used within the code is hard-coded at compilation time and cannot be changed via changing the label. *Ewald* represents a standard () Ewald summation for the long range portion of the electrostatic sums. The real-to-inverse switch distance is taken to be the cutoff radius as specified in line 2.



*Alpha* represents the raw screening length parameter for the Ewald summation. It is modified as , where is the size of the periodic cell before utilization in the Ewald summation.



The *Maximum K Vector* (kmax) parameter controls the maximum magnitude of the lattice vector offset in the Ewald summation. For lattice offset vectors , where are the real-space lattice vectors, the Ewald summation is carried out on periodic images up to the boundary of .



Line 12

**\* ---- Integrator Parameters**

\* Integrator Type Medium Steps Large Steps Medium Timestep Cutoff

*Multiple3 3 6 6.5*

*Integrator type* : Character\*12

*Medium Steps (Sm)* : Integer

*Large Steps (Sl)* : Integer

*Short Cutoff (Rs)* : Double precision

The *integrator type* is a non-control label which identifies the characteristics of the integrator being employed. The included integrator for this codebase is referenced as

**Multiple3** – A multiple timestep integrator which breaks the systematic interactions of the simulation into three different time scales: *short* (bonds, bends, and improper torsions), *medium* (torsions and short-range nonbonded interactions), and *long* (full-range nonbonded interactions, electrostatics, and any extended lagrangian integrations within the system).

For the Multiple3 integrator, the following variables are defined:

The inherent time step of the integrator () represents the inherent time scale of the smallest time step. *tshort=D*

*Step Med* controls the number of small steps in a single medium time step: *tmed=SmD*

*Step Large* controls the number of small steps in a single large time step: *tlarge=SlD*

*Short Cutoff (Rs)* is the cutoff radius (in Å) for the short-ranged nonbonded interactions which are calculated at the medium time steps.

Line 13

**\* ---- Polarization parameters**

\* Q-mu mu-mu | Polarization Tol. Tapering Taper Start Epsilon(RF)

*Ewald RF/Taper* | *1.0d-7 .true. 9.5 1.0*

*Q-mu mu-mu* : Character\*19

*Polarization Tol.* : Double precision

*Tapering* : Boolean

*Taper Start (RTS)* : Double precision

*Epsilon (RF)* : Double precision

The *Q-mu mu-mu* label is a non-control label that indicates the interaction model for static charges with induced (many body) dipoles (*Q-mu*) and the interaction model for induced-dipoles with other induced dipoles (*mu-mu*). The models used for each subtype of the charge interactions are as follows:

**Charge-Dipole Interaction Types**

**Ewald** Interactions between static charges and many-body induced dipoles are calculated using the Ewald method.

**Dipole-Dipole Interaction Types**

**RF/Taper** Interactions between induced dipoles are calculated directly, using a screened point-charge representation, for all atoms within the cutoff radius (see line 2) via summed pairwise interactions. Tapering of the interaction to yield smooth forces at the cutoff is included as an option. Polarization of all atoms is calculated via an iterative process. Interaction with points outside of the spherical cutoff radius are calculated via a reaction-field approximation.

For the RF/Taper dipole interactions, the following variables are defined:

The polarizability tolerance [*Polarization Tol.*] is the maximum allowed difference (in Å2e2) between iterated solutions of the polarizabilities of the atoms below which the atoms are considered to have self-consistently converged to the actual polarization state.

The *Tapering* variable serves as a flag to control whether or not tapering of the dipole-dipole interaction occurs. If tapering is set to *.true.* then the dipole-dipole potential is tapered starting at *RTS* and going to zero at the cutoff radius from line 2 (*Rc*).

The *epsilon (RF)* parameter controls the permittivity of the reaction field around the dipole moment, which serves to mimic the character of the dielectric medium outside the cutoff radius. A value of 1.0 is representative of vacuum outside the dipole radius.

ff.dat

The ff.dat file is comprised of seven sections that contain the necessary information to completely specify the force field determining inter-atomic interactions between the species present in the simulation. The ff.dat file controls both non-bonded and bonded interactions, including dispersion, charge-charge interactions, polarizability, bonds, bends, torsions, and improper torsions.

Section 1

**Number of non-bonded types**

The first data line in the ff.dat file looks similar to the following:

**\* Number of repulsion-dispersion types (T\_RD)**

9

The number of repulsion-dispersion types (TRD) is the number of atomic species with unique repulsion-dispersion interactions within the system. This number should be less than or equal to the parameter *maxtypes* in the params.h file.

Section 2

**Non-bonded parameters**

Non-bonded interactions are represented by the combination of a repulsion-dispersion interaction, a coulomb interaction term representing the two-body electrostatic potential, and a many-body polarizability term:









The lines following the number of dispersion types declaration set out the interaction parameters for the dispersion interactions. At a minimum, there must be at least *TRD* lines, each of which contain a full set of dispersion parameters for the indicated type. A full set of repulsion-dispersion parameters for a single repulsion-dispersion type contains the following terms:

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**\* Label A B C D Mass Notes**

\* Charge Induced polarizability

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*Cm* ***108283.00 3.64050 560.53*** *0.00* ***12.011*** *exp-6 |8. Cm*

Parameter Units Variable Type

*Label* None Character\*3

*A*  kcal/mol Double precision

*B*  Å-1 Double precision

*C* Å6·kcal/mol Double precision

*Mass* g/mol Double precision

The *label* is a 3 character tag used to denote the specific atom type for future use in the coordinates files.

The *A* and *B* parameters collectively define the repulsive barrier for the non-bonded interactions between two atoms of the labeled types, while the *C* parameter describes the attractive tail of the potential. The atomic *mass* is also specified here. The remainder of the line after the *mass* specification may be utilized for notes about the species described. The parameter labeled *D* is no longer utilized in the current forcefield, and should be set to 0.0.

In addition to the dispersion parameters, it is also possible to specify one or a number of charge permutations for the base atom. In order to specify a charge permutation associated with a set of dispersion parameters, the lines following the description of the dispersion parameters are appended with a line for each different charge species present. These charge description lines are required to start with at least one space, after which the static *charge* and *induced polarizability* of the species is listed:

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

\* Label A B C D Mass Notes

**\* Charge Induced polarizability**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cm 108283.00 3.64050 560.53 0.00 12.011 exp-6 |8. Cm

***-0.0929 0.6500*** *|Nitrogen bonded methyl carbon in im+, pyr+*

***-0.0764 1.0500*** *|Terminal methyl carbon in alkyl chain*

Parameter Units Variable Type

*Charge*  *(qi)* e Double precision

*Induced polarizability (αi)* Å3 Double precision

The *charge* represents the total static charge of the given species, while the *induced polarizability* represents the relative ability of local electric field gradients to further polarize the charge at the atom. The remainder of the line is available for notes regarding what the particular charge species represents.

The above snippet of the ff.dat file defines two atom types. Both have dispersion parameters that represent a methyl carbon. The first atom type has a static charge of -0.0929e, an induced polarizability of 0.6500 Å3, and represents an ammonium bonded methyl group while the second has a static charge of -0.0764e, an induced polarizability of 1.0500 Å3, and represents a carbon bonded methyl group. Both atom types share the label Cm, with the different charge types being represented in the coords.inp file by the addition of a two-integer charge index to differentiate the different charge species (see the description of coords.inp for further information).

In addition to combining rules, it is possible to specify non-default cross-terms to better describe the interaction between two different atom types. The non-default cross-terms immediately follows the dispersion parameter section and has the following format:

\* non-default cross terms

**\*Label1\_Label2 A B C D**

\* ------------------------------------------------------------------

*Cm \_C 102702.04 3.63127 539.82 0.00*

*Cm \_N+ 52447.87 3.93375 261.55 0.00*

*…*

*N \_O= 23978.15 3.45339 446.39 0.00*

*!*

Parameter Units Variable Type

*Label1\_Label2* None Character\*7

*A*  kcal/mol Double precision

*B*  Å-1 Double precision

*C* Å6·kcal/mol Double precision

Lucretius™ parses the appropriate cross-term interaction by using the value of *Lbl1\_Lbl2* to form the appropriate index into the interaction matrix. As such, it is critical that both *Lbl1*\_*Lbl2* be exactly 7 characters in length, with the first three characters comprising one full label (including any trailing spaces necessary to reach 3 characters total) from the preceding section, the fourth character is the underscore (“\_”), and the fifth through seventh characters comprising a second, differing full label from the preceding section. The values of *A*, *B*, and *C* have the same meanings and units as in the preceding section, except here they dictate the interaction between two atoms of differing types. As in the homoatomic section, the D parameter is no longer used and should always be set as 0.0. The non-default cross term section is terminated by a line containing a single ! as the first character on the line.

Section 3

**Bonding Potentials**

The bond energy between any two contiguous atoms employs a harmonic oscillator model:



The first line of the bond parameterization subsection of the ff.dat file dictates the number of unique bond types (TBond) present in the system:

**\* Bonds**

*14*

The remaining TBond lines in the bonded energy subsection describe the individual bonds in ascending order, indexed from 1 to TBond. Each line consists of a (informative only) label, as well as three parameters representing the bond interactions. Mapping of atoms to a specific bond type is accomplished in the connectivity.dat file.

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**\* Label\_ij K\_Bond R0 R0 (constrained)**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*C 12Cm 1 618.000 1.520 1.520 |Methyl carbon to ethyl carbon*

*H 0 Cm 1 655.000 1.100 1.100 |Hydrogen to methyl carbon*

*C 12N+ 1 739.000 1.472 1.472 |Carbon to imidazolium nitrogen*

Parameter Units Variable Type

*Label\_ij* None Character\*10

*KBond* kcal/mol·Å2 Double precision

*R0* Å Double precision

*R0 (constrained)* Å Double precision

The bond label, *Label\_ij*, is a non-control text field which serves to identify the species of atoms involved in the referenced bond.

The bond parameters actually used within the simulation calculation begin with *KBond*, which represents the spring constant for the harmonic oscillator model, while *R0* represents the associated average bond length.

If the SHAKE algorithm is used to rigidly constrain the bonds, then *R0 (constrained)* is utilized for the rigid bond length instead.

Any additional data beyond the value for *R0 (constrained)* is discarded and can therefore be used for additional notes about the defined bonds, as long as at least one space is present after the numerical value for *R0 (constrained)*.

Section 4

**Bending Potentials**

The bend energy between any three contiguous atoms employs a harmonic potential about the bend rest angle:



The first line of the bend parameterization subsection dictates the number of unique bending potentials (TBend) defined within the system:

**\* Bends**

*23*

The remaining TBend lines in the bend potential subsection describe the individual bending interactions in ascending order, indexed from 1 to TBend. Each line consists of a (informative only) label and two parameters providing the bend energy constant () as well as the nominal bend angle (q0). Mapping of atoms to a specific bend types is accomplished in the connectivity.dat file.



\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**\* alpha\_beta\_gamma K\_Bend Theta0 Linear**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*H 1 Cm 0 C 0 86.00 110.500 F | from C4,C5 alkanes*

*H 0 Cm 1 H 0 77.00 107.700 F | terminal methyl group*

Parameter Units Variable Type

*Alpha\_beta\_gamma* None Character\*15

*KBend* kcal/mol·rad2 Double precision

*Theta0 (0)* degrees Double precision

The label *alphabetagamma* is a non-control label which serves to identify the species involved in the bending potential.

The bending potential parameters are comprised of *KBend* which specifies energy constant for the harmonic bending potential and *0* which specifies the reference angle for a bend containing atoms of species α, β, and γ in that order. The *Linear* option is currently not used for calculation. Any extra data after *0* is therefore discarded, and can therefore be used to provide additional notes about the bends defined.

Section 5

**Torsional Parameters**

The torsional energy between any four contiguously bonded atoms is specified by:



The first line of the torsional energy subsection of ff.datprovides the number of unique torsional interactions (TTorsion) within the system:

**\* Torsions**

21

The remaining TTorsion lines in the torsional energy subsection describe the individual torsional potentials in ascending order, indexed from 1 to TTorsion. Each line consists of an informative label indicating the identity of the four species involved in the torsional paramaterization, followed by the total number of terms in the torsional expansion, and the value of for each of these expansion terms. The maximum number of terms possible in the expansion is set at compilation time via the params.h file. Mapping of specific atoms to a torsional interaction type is accomplished via the connectivity.dat file.



**\* alpha\_beta\_gamma\_delta nprms K0 K1 K2 K3 K4...**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*H 0 Cm 0 C 12N+ 1 4 0.00 0.00 0.00 -0.30 |Imidazolium H-C-C-C*

*H 0 Cm 0 C 0 H 0 4 0.00 0.00 0.00 -0.30*

*Cm 0 C 12N+ 1 Cc 0 3 0.00 0.00 0.43 |Imid+. Cm-C-N+-C*

Parameter Units Variable Type

*alpha\_beta\_gamma\_delta* None Character\*20

*nprms (n)* None Integer

*K0, K1, K2, K3, …, Kn* kcal/mol Double precision

The label *alpha\_beta\_gamma\_delta* is a non-control label which identifies the species of the four atoms for which the torsional parameters provided apply. The definition of the torsional energy term begins with *nprms* which is the number of potential energy terms utilized to build the complete torsional energy curve. Following this are *n* double precision values representing the appropriate force constants for the potential energy expansion, with at least a single space between each constant. Any extra data beyond the last force constant is disgarded, and can therefore be used for further notes about the defined torsions.

Section 6

**Planar Deformations**

The energy term utilized for planar deformations is:



For this energy term, *j* represents the out-of-plane atom, and the only necessary forcefield parameter is the energy constant,



As in all previous subsections, the first line of this subsection denotes the number of unique out-of-plane interactions (TDeform):

**\* Out-of-plane deformations**

*4*

The next NDeform lines provide requisite labels and the necessary energy constant for each unique interaction type, indexed from 1 to TDeform:

**\* alpha\_beta\_gamma\_delta K\_abgd**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

\*

*Cc 8 N+ 1 Cc 9 C 12 65.0*

*N+ 1 Cc 8 Cc 8 H 2 57.0*

Parameter Units Variable Type

*alpha\_beta\_gamma\_delta*  None Character\*20

*K\_abgd (Kabgd)* kcal/mol⋅rad2 Double precision

The label *alpha\_beta\_gamma\_delta* is a non-control label which identifies the species which must be constrained in a planar fashion. The force constant, *Kabgd* determines the stiffness of the constraint that attempts to maintain planarity of the molecule. *The force constant Kabgd should always be positive. A value less than 1.0e-5 is used to indicate a planar constraint on the molecule, and the deformation will be forced to planarity.* Any extra data beyond the force constant is ignored.

Section 7

**Lone Pair Definition**

The final subsection of the ff.dat file contains the necessary details to define the lone pair dummy atoms within the system. For specific systems (e.g. water) the geometric requirements to obtain proper orientation due to local electronic interactions (e.g. hydrogen bonding) is not sufficiently accounted for by simple charge or dipole moments located at the center of the atom in question. For this reason, Lucretius™ implements the ability to explicitly describe lone pairs in one of three geometries. These lone pairs are treated as dummy force centers with rigidly constrained geometries. Since the dummy atoms are rigidly constrained, energy for any bonded interactions is identically zero by construction, and resultant forces from non-bonded interactions are transferred to the host atoms used to form the lone pair geometry.

There are three types of lone pair geometries currently implemented in Lucretius. Each type is represented by three values representing the geometric specification of the lone pair location:

**Type 1** lone pairs represent the lone pair with a single force center parallel to the bisector of angle A1-A2-A3. For such lone pairs, the length lB represents the distance between the atom to which the lone pair is attached and the center of the dummy force center representing the lone pair.

**Type 2** lone pairs have two force centers attached to the central atom, extending out of the plane of the bend A1-A2-A3. Here, the distance from the central angle atom to the line which connects the center of the two dummy force centers is lB, while the distance from the plane of the atom to the force centers, perpendicular to the plane is lC. The overall distance from the atom to the force centers is therefore .

**Type 3** lone pairs have two force centers attached to the central atom, extending in the plane of the bend A1-A2-A3. Here, the distance from the central angle atom to the line which connects the center of the two dummy force centers is lB, while the distance from the force center to the imaginary line formed by the bisector of angle A1-A2-A3 is lC. The overall distance from the atom to the force centers is again . ***Please note that type 3 lone pairs are currently still experimental.***

The lone pair subsection of the ff.dat file begins with the number of uniquely specified lone pair geometries (TLP):

**\* Lone Pair definitions**

*2*

The remaining TLP lines provide labels, the relevant charge type, and the necessary parameters to define the lone pairs in the system, indexed from 1 to TLP:

\* Bend Atoms Charge Index r\_A l\_B l\_C Type A1 A2 A3

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

H 1 O 1 H 1 6 0.50 -0.23808 0.00000 1 3 4 3

Cc 1 O 2 Cc 1 7 0.50 0.409060 0.505140 2 6 5 6

Parameter Units Variable Type

*Bend Atoms*  None Character\*20

*Charge Index* None Integer

*rA* None Double precision

l*B* Å Double precision

l*C* Å Double precision

*Type* None Integer

A1 None Integer

A2 None Integer

A3 None Integer

The label *Bend Atoms* is a non-control variable which identifies the species forming the bend upon which the lone pairs are rigidly fixed.

The *charge index* is an integer which contains the numeric index of the atom type to be considered a lone pair. This number is the index of the defined atom type which is used to represent the lone pair, where the index is derived by counting all previous charge combinations. For example, in the following snippet of a ff.dat file, the bolded, underlined numbers on the left represent the charge index of the related atomic species. Note that if a repulsion-dispersion type does not have explicit charges and polarizability, both quantities are assumed to be zero. In such a case, the affected atom *still counts as a charge species* (as in the first labeled charge species below):

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**\* Label A B C D Mass Notes**

\* Charge Induced polarizability

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**1.** *C 97431.00 3.62220 519.87 0.00 12.011 exp-6|6. C*

H 5352.30 4.36460 22.59 0.00 1.008 exp-6|1. H

**2.***0.0576 0.3500 |1, H\*-C Bonded to C 8,*

**3.** *0.1169 0.3500 |1, H\*-C Bonded to Cm 2,*

Lp 0.00 0.00000 0.00 0.00 0.000 lj126|Lone pr

**4.** *-1.1074 0.0000 |6, force center in SWM4 H2O Bonded to O*

**5.** *-0.2200 0.0000 |1, ether group Bonded to O,*

The next three parameters are used to fix the location of the lone pair relative to the molecular geometry of the atoms of the related bend:

*rA* – Relative proportion of the vector from one terminal atom (A1) to the other (A3) along which the line intersecting the central atom of the bend (A2) is drawn. This line, in turn, serves as the direction in which the length lB is extended. For current implementations, this value should be set to 0.5 to approximate the bisection of the A1-A2-A3 angle.

l*B* – The (projected) distance along the line (plane) containing the bisector from the central atom of the affected bend. To locate the lone pair exterior to the bend, as in dimethoxyethane, l*b* should be positive. To locate the lone pair interior to the bend, as in TIP4P water, l*b*should be negative.

l*C* – In a type 1 lone pair, lC should be zero. For a type 2 lone pair, lC represents the perpendicular distance from the plane containing the atoms A1-A2-A3 to the dummy force centers (one at - lC the other at lC). For a type 3 lone pair, lC represents the perpendicular distance from the bisector of A1-A2-A3 to the dummy force centers (one at - lC the other at lC).

The *LP Type* specifies whether the lone pairs are represented by a dummy atom with a single lobe in the plane of the bend (*type 1*), by two dummy atoms which form a tetrahedrally coordinated structure with the related central bend atom (*type 2*), or by two dummy atoms which form a planar structure with the related central bend atoms (*type 3*).

Finally, the A1, A2, and A3 parameters correspond to the *charge indices* (see above) for the atom types of the A1, A2, and A3 atom positions in the lone pair geometry. This allows for both error checking to ensure that lone pairs are placed on the proper atoms, as well as the unambiguous specification of location in type 3 lone pairs, where the construction geometry differs from the bonding connectivity.

coords.inp/coords.out

The coords.inp and coords.out file both have the same format. These files represent a complete summary of the instantaneous positions and velocities of all the atoms in the system in a plain ASCII format, as well as containing information about the system cell size and the state of any extended lagrangian variables within the system. Since both files have identical formats, we will refer to them as simply “coordinate files”.

The coordinate files have three major portions:

1. An informational header which may store a text description of the system simulated
2. A list of positions and velocities, one set of each for every force center in the system
3. An extended system which provides the definition of the simulation cell at a minimum, and may also provide extended coordinates for thermostat and barostat variables.

Section 1

The coordinate files begin with a header comprised solely of an arbitrary number of comment lines (lines whose first character is an asterisk ) which are generally used to provide useful information about the system.

Section 2

Following the introductory header is a section comprised of Nfc blocks (where Nfc is the number of individually defined simulation force centers), with each block comprised of two lines with the following format:

**\* X Y Z Atom Charge**

**\***  *V\_X V\_Y V\_Z*

\*-----------------------------------------------------------------------

\*

***22.538988 25.630641 10.683374 Cm 2***

*-24.740393 -235.265295 -106.378173*

22.583869 24.214686 11.234227 C 1

-157.029719 -392.865083 -497.809885

22.744302 24.144902 12.695788 N+ 1

275.429794 -242.863373 -537.171476

Parameter Units Variable Type

*X*  Å Double precision

*Y*  Å Double precision

*Z* Å Double precision

*Atom Type (Atom)* None Character\*3

*Charge Subtype (Charge)* None Integer

*VX* m/s Double precision

*VY* m/s Double precision

*VZ*  m/s Double precision

*Line 1*

*X, Y,* and *Z* provide the Cartesian coordinates which describe the location of the relevant force center. The valid range of these values is from 0.0 to *BX, BY,* and *BZ*, respectively, where *Bi* is the length of the simulation box in the *i* direction.

After the Cartesian coordinates, the *atom type* label associated with the proper species in the ff.dat file is provided, as well as the relevant charge subtype within that species. Valid values for charge subtype are from 1 to NQ-Label where NQ-Label is the total number of charge subtypes for the species of type *label*.

*Line 2*

*VX*, *VY*, and *VZ* provide the *X*, *Y*, and *Z* components of the velocity of the referenced atom.

Section 3

At the end of the coordinate files is an additional section which contains the necessary information to specify the simulation cell and any extended variables necessary for simulation in ensembles other than the microcanonical.

\*-----------------------------------------------------------------------

\* **Box (Angstroms)**

35.1971066497367

**\* Xlogs (Thermostat position)**

-2.18539477964628

**\* Vlogs (Thermostat velocity)**

1.494671827159197E-004

**\* Glogs (Thermostat force)**

2.931502988086876E-006

**\* Xlogv (Barostat position)**

-5.366430103976744E-004

**\* Vlogv (Barostat velocity)**

-6.873090614127863E-006

**\* Glogv (Barostat force)**

7.293176555254246E-008

Parameter Units Variable Type

*Box (LBox)*  Å Double precision

*Xlogs*  Å Double precision

*Vlogs* m/s Double precision

*Glogs* Double precision

*Xlogv* Å Double precision

*Vlogv* m/s Double precision

*Glogv* Double precision

The size of the box is given in the section marked *Box*. If the system is cubic, a single value is listed, while an orthorhombic system will have three values representing the box length in the *X*, *Y*, and *Z* directions respectively.

*Xlogs, Vlogs,* and *Glogs* represent the variables necessary to track the thermostat evolution in an NVT or NPT simulation. Each section will have NNHC values *(see* mdrun.params*)* representing the thermostat “position”, “velocity”, and “force” of the indexed Nosé-Hoover chain temperature coupling to the virtual “heat bath”. At the beginning of a run, or after equilibration, it is generally recommended to set all of these variables to 0.0 in order to “erase” the thermostat history before generating data for a production run. Otherwise, these values are generally not modified except through the simulation itself.

*Xlogv, Vlogv,* and *Glogv* represent the variables necessary to track the barostat evolution in an NPT simulation. Here, each section has either one (cubic) or three (orthorhombic) values which house the associated “position”, “velocity”, and “force” of the Nosé-Anderson barostat mass which is used to control the pressure of the system. Like the thermostat variables, it is recommended to set all the barostat associated quantities to 0.0 at the beginning of a production set of simulation runs and to otherwise not modify these values outside of the normal process of running a simulation in the NPT ensemble.

connectivity.dat

The connectivity.dat file provides the molecular connectivity information necessary to completely specify the atoms, bonds, bends, torsions and deformations of each molecular species within the system. The connectivity.dat file is comprised of five separate sections: The first section dictates the number of molecular and atomic units within the system. The second section provides the number of atoms, bonds, bends, torsions and deformations per unit. The third section provides a list of all the bonds, which are mapped to molecules via the information in section two and the table of force centers provided through coords.inp. The fourth, fifth, and sixth sections provide a list of all the bends, torsions, and deformations which are mapped in the same manner as the bonds.

For ease of reference, we will collectively refer to either a molecule or an isolated single atom/ion as a *unit*, while the individual atoms/lone pairs which comprise the molecule or atom/ion will be referred to as a *force center*.

Section 1

**Unit declarations**

The first section provides the number of molecules (*NMol*) and single atoms/ions (*NSingle*). Each unit must be defined as one or the other.

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

\* **Number of Molecules Single atoms/ions**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*4 1*

Parameter Units Variable Type

*Number of Molecules (NMol)* None Integer

*Single Atoms/Ions (NSingle)* None Integer

Section 2

**Unit interaction list**

The second section provides *NMol+NSingle* lines (one for each unit). The *ith* line in the section lists the number of force centers, bonds, bends, torsions and deformations for the corresponding *ith* unit in the system.

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

\* Chain configuration

\*

\* **Atoms Bonds Bends Torsions Deformations**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*19 19 33 41 5*

*19 19 33 41 5*

*19 19 33 41 5*

*19 19 33 41 5*

*1 0 0 0 0*

Parameter Units Variable Type

*Atoms ()* None Integer

*Bonds ()* None Integer

*Bends ()*  None Integer

*Torsions ()* None Integer

*Deformations ()*  None Integer

The *Atoms* *()* value determines how many force centers are in the current (*ith*) molecule, with the force centers assigned in sequential order from the list generated from the coords.inp file. Thus, in the preceding example, the coords.inp file would be expected to contain 77 unique force center descriptions, with the information for force centers 1-19 ( ) representing the coordinates for the members of molecule 1, 20-38 for molecule 2 ( ), and so on, up to coordinate number 77 ( )representing the data for the sole single atom/ion in the system.

The *Bonds* () value determines how many bonds are in the current (*ith*) unit, with the bonds assigned from the list of bonds in *Section 3* of connectivity.dat. The bonds are assigned sequentially, with the first  bonds assigned to molecule 1, the next bonds assigned to molecule 2, and so on. For single force center units, even though , *Section 3* is required to have a single line which explicitly demonstrates the lack of bonds with the referenced force center. Thus, the total length of the bond list is .

The *Bends* *()* value determines how many bends are in the current (*ith*) unit, with the bends assigned from the list of bends in *Section 4* of connectivity.dat. The bends are assigned sequentially, with the first  bends assigned to molecule 1, the next bends assigned to molecule 2, and so on.

The *Torsions ()* value determines how many torsions are in the current (*ith*) unit, with the torsions assigned from the list of torsions in *Section 5* of connectivity.dat. The torsions are assigned sequentially, with the first  torsions assigned to molecule 1, the next torsions assigned to molecule 2, and so on.

The *Deformations ()* value determines how many deformations are in the current (*ith*) unit, with the deformations assigned from the list of deformations in *Section 6* of connectivity.dat. The deformations are assigned sequentially, with the first  deformations assigned to molecule 1, the next deformations assigned to molecule 2, and so on.

Section 3

**List of bonds**

The third section provides the bond list, which is used to determine the bonds present in the molecules as well as to explicitly denote non-bonded single force center units. The bond list should contain  lines. An example bond list follows:

\* Bonds

\*

**\* FC\_i FC\_j Bond Type**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*1 2 1*

*1 9 2*

*1 10 2*

*1 11 2*

*2 3 3*

*…*

*77*

Parameter Units Variable Type

*FCi* None Integer

*FCj* None Integer

*Bond Type*  None Integer

The values of *FCi* and *FCj* specify the force centers (*i* and *j*, respectively) which comprise the specified bond.

The *Bond Type* specifies the relevant parameters for the bond energy/force calculation by referencing the relative position in the bond type subsection of the ff.dat file. (*Section 3* of ff.dat)

The final example bond above provides an example of the required statement for a single force center unit. Here, only *FCi* is defined, with the rest of the line intentionally left blank to demonstrate the unbonded nature of the atom.

Section 4

**List of bends**

The fourth section provides the bend list, which is used along with the information from *Section 2* to determine which bend interactions are present within which units. The bend list should contain  lines. An example bend list follows:

\* Bends

\*

**\* FC\_i FC\_j FC\_k Bend Type**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*9 1 2 1*

*10 1 2 1*

*11 1 2 1*

*10 1 9 2*

*…*

Parameter Units Variable Type

*FCi* None Integer

*FCj* None Integer

*FCk* None Integer

*Bend Type*  None Integer

The values of *FCi*, *FCj*, and *FCk* specify the force centers (*i*, *j*, and *k* respectively) that comprise the specified bend. Here, *j* should represent the force center interior to the bend.

The *Bend Type* specifies the relevant parameters for the bend energy/force calculation constants by referencing the relative position in the bend type subsection of the ***ff.dat*** file. (*Section 4* of ***ff.dat***)

Section 5

**List of torsions**

The fifth section provides the torsion list, which is used along with the information from *Section 2* to determine which torsional interactions are present within which units. The torsion list should contain  lines. An example torsion list follows:

\* Torsions (Dihedrals)

\*

**\* FC\_i FC\_j FC\_k FC\_l Torsion Type**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

9 1 2 3 1

9 1 2 12 2

9 1 2 13 2

10 1 2 3 1

10 1 2 12 2

…

Parameter Units Variable Type

*FCi* None Integer

*FCj* None Integer

*FCk* None Integer

*FCl* None Integer

*Torsion Type*  None Integer

The values of FCi, FCj, FCk and FCl specify the force centers (*i, j, k* and *l* respectively) that comprise the specified torsional angle. Here, *j* should represent the force center interior to the bend *ijk* and *k* the force center interior to the bend *jkl*.

The *Torsion Type* specifies the relevant parameters for the torsional energy/force calculation constants by referencing the relative position in the torsion type subsection of the ***ff.dat*** file. (*Section 5* of ***ff.dat***)

Section 6

**List of deformations**

The sixth and final section provides the deformation list, which is used along with the information from *Section 2* to determine which force centers within which units have interactions which enforce planarity of a quartet of bonded atoms. The deformation list should contain  lines. An example deformation list follows:

\* Out-out-plane deformations (Improper Torsions)

**\* FC\_i FC\_j FC\_k FC\_l Deform Type Inside**

\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*5 3 4 2 1 1*

*6 4 3 14 2 1*

*7 5 3 15 3 1*

*7 6 4 16 2 1*

Parameter Units Variable Type

*FCi* None Integer

*FCj* None Integer

*FCk* None Integer

*FCl* None Integer

*Deform Type*  None Integer

*Inside* None Integer

The values of FCi, FCj, FCk and FCl specify the force centers (*i, j, k* and *l* respectively) that comprise the specified planar force center quartet. Here, *j* should represent the central force center and should be bonded to *i,k,* and *l*.

The *Deform Type* specifies the relevant parameters for the deformation energy/force calculation constants by referencing the relative position in the deform type subsection of the ***ff.dat*** file. (*Section 6* of ***ff.dat***)

The *Inside* parameter can take the value of either **-1** or **1** only, and is used as a flag to determine whetherthe force center represented by *FCl* is interior to angle *ijk*. If *Inside* is -1 *l* is interior to angle *ijk*, while a value of 1 means *l* is exterior to angle *ijk*.

**params.h**

The params.h file is part of the compile-time control file set for the Lucretius™ simulation package. Multiple values within the params.h file must be properly set in order for Lucretius™ to function properly. This section provides a list of the parameters within params.h and discusses appropriate values for these parameters. These parameters may be defined in any order (although some parameters depend on previously defined parameters and may thus be kept in the specified order for the definition to make sense.

C Parameters for unit and force center list

**integer maxnch** !Maximum number of molecules + atoms

*parameter (maxnch= 250)*

**integer maxat** !Maximum number of force centers

*parameter (maxat= 3500)*

C

C Parameters for connectivity lists

**integer maxbonds** !Maximum number of bonds in system

*parameter (maxbonds= 3375)*

**integer maxbends** !Maximum number of bends in system

*parameter (maxbends= 5750)*

**integer maxtorts** !Maximum number of torsions in system

*parameter (maxtorts= 5875)*

**integer maxdeforms** !Maximum number of deformations in system

*parameter (maxdeforms= 625)*

C

C Parameters for force field

**integer maxtypes** !Max number of repulsion-dispersion types

*parameter (maxtypes= 9)*

**integer maxcharges** !Max number of charge types

*parameter (maxcharges= 16)*

**integer maxdummy** !Max number of dummy (lone pair) atoms

*parameter (maxdummy= 0)*

**integer maxbtypes** !Max of types(bonds,bends,torts)

*parameter (maxbtypes= 23)*

**integer maxfolds** !Max number of terms in torsional energy

*parameter (maxfolds = 9 )*

C

C Parameters for neighbor list

**integer maxbox** !Maximum force centers per cell subbox

*parameter (maxbox = 25)*

**integer maxnay** !Maximum neighbors per force center

*parameter (maxnay = 999)*

**integer maxdim** !Maximum box subdivisions per axis

*parameter (maxdim = 25)*

**integer maxdim3** !Total number of box subdivisions

*parameter (maxdim3 = maxdim\*maxdim\*maxdim)*

C

C Polarization related parameters

**real\*8 a\_thole** !Dipole-dipole damping constant

*parameter (a\_thole=0.2d0)*

C

C Misc additional parameters as indicated (advanced)

**integer maxpoints** !Max points in interaction splines

*parameter (maxpoints = 501)*

**integer maxprop** !Max number of system averaged properties

*parameter (maxprop = 30)*

**integer maxnnb** !Max number of non-bonded interactions

*parameter (maxnnb = maxcharges\*(maxcharges+1)/2)*

**integer kmax** !Max k vectors in one direction

*parameter (kmax = 6)*

Section 1

The first two parameters control the length of the lists which hold the individual data about each force center as well as the overarching data about each unit (number of bonds, bends, torsions, etc):

*maxnch* The maximum number of units (molecules and atoms/ions) within the current simulation. (, see ***connectivity.dat***)

*maxat* The maximum number of force centers within the current simulation. (, see ***coords.inp***)

Section 2

The next four parameters hold the size data for the bond, bend, torsion, and out-of-plane deformation lists:

*maxbonds* The maximum number of individual bond definitions within the simulation. (, see ***connectivity.dat****, Section3*)

*maxbends* The maximum number of individual bend definitions within the simulation. (, see ***connectivity.dat****, Section 4*)

*maxtorts* The maximum number of individual torsion definitions within the simulation. (, see ***connectivity.dat****, Section 5*)

*maxdeforms* The maximum number of out-of-plane deformations within the simulation. (, see ***connectivity.dat****, Section 6*)

Section 3

The next three parameters hold necessary information for the size of data structures dictated by the employed forcefield:

*maxtypes* The maximum number of repulsion-dispersion interaction types. This value does not take into account differing charge/polarizability species. (, see ***ff.dat***, *Sections 1 and 2*)

*maxcharges* The maximum number of different non-bonded interaction types *including charge/polarizability* species. (see ***ff.dat***, *Sections 2 and 7*)

*maxdummy* The maximum number of “dummy” force centers utilized in the simulation to represent lone pairs for the system. There will be one dummy force center for every lone pair of type 1 (), and two dummy force centers for every lone pair of type 2 (). (, see ***ff.dat***, *Section 7* for further details about lone pair representation)

*maxbtypes* The maximum of the number of *types* of bonds, bends, torsions, or out-of-plane deformations found in the ***ff.dat*** file. (, see ***ff.dat****, Sections 3-6*)

*maxfolds* The maximum number of individual torsional terms allowed in the torsional energy definitions. (see ***ff.dat***, *Section 5* for details on torsional energy term expressions)

Section 4

The next four values contain information related to the neighbor list implementation utilized by Lucretius™. Lucretius™ calculates the neighbor list only when necessary (see ***mdrun.params****, Line 2*), utilizing a cell-based approach to organizing the force centers to further accelerate neighbor list formation. The simulation cell is subdivided into many equal volume sub-cells, and the individual force centers are indexed into the appropriate sub-cell. The neighbor list calculation then proceeds through all force centers, checking only those nearby sub-cells that might fall within the total interaction distance placing those which are inside the total interaction distance into the appropriate neighbor list. The following variables determine the size, number, and length of the sub-cells as well as the length of the neighbor list:

*maxbox* The maximum number of force centers which can be indexed per sub-cell. If Lucretius™ finds more atoms in a sub-cell than this number, the simulation will be terminated with an appropriate message instructing the user to increase this number. (We need to make sure this check is actually in the code, and insert it if it’s not.)

*maxnay* The maximum number of neighbors allowed in a neighbor list. If Lucretius™ attempts to add more atoms to the neighbor list than it can contain, the simulation will be terminated with an appropriate error message instructing the user to increase this number.

*maxdim* The maximum number of sub-cells in any given direction within the unit cell. For example, a *maxdim* value of 10 will subdivide the overall system cell into 10 subcells along each principle Cartesian vector.

*maxdim3 maxdim3 –* The total number of sub-cells within the system. A quick estimate for sufficient sub-cell numbers is:  , assuming homogeneous spatial distribution of the force centers. As the spatial distribution of the force center becomes less homogeneous, the prefactor on the right hand side of the preceding equation increases.

Section 5

This parameter is used to delocalize the dipole-dipole interaction in order to try to avoid a “polarization catastrophe”

*athole* The damping constant for Thole screening of the dipole-dipole interaction.

Section 6

The remaining parameters in the ***params.h*** file control miscellaneous variables for the internal workings of Lucretius™ and should generally be left as received. There are occasions, however, where they will need to be changed for specific applications, and as such they are summarized below:

*maxpoints* The maximum number of discrete points utilized in construction of the spline which is utilized to evaluate the real-space component of the non-bonded interactions.

*maxprop* The maximum number of time-averaged properties which can be tracked by Lucretius as the simulation progresses.

*maxnnb* An automatically generated quantity which sets the appropriate size for the interaction matrix used to track the various components of non-bonded interactions between atom types.

**Binary output file specifications**

There are four potential binary output files which can be generated via Lucretius™. Whether or not each file is created, as well as the interval of output if the file is created, is controlled via lines 5 and 6 of the ***mdrun.params*** file. Each file contains an arbitrary number of snapshots of the relevant data and any associated header information, and is output in a purely binary format from Lucretius™. These files are generally used as a compact method (compared to ASCII output) for saving data for which the desired analysis is too time consuming or otherwise unsuitable for being performed during the execution of the simulation. The four potential file types are:

*Total Box Dipole and Charge Flux* – filename: *fort.76*

This file is composed of an arbitrary number of two-line blocks. In each block, the first line contains a single value giving the time interval at which the dipole and flux are calculated (*tDipole*). The second line contains the total box dipole (in Åe ) in the X, Y, and Z direction (*DX*, *DY*, and *DZ* respectively) as well as the portion of the total box dipole due to the dipole-dipole interactions (in Åe ) in the X, Y, and Z direction (*PX*, *PY*, and *PZ* respectively). The variable template for the output block is:

Real\*4

6 × Real\*4

*Instantaneous Coordinates –* filename: *fort.77*

This file is composed of an arbitrary number of two-line blocks. In each block, the first line contains four values representing the total number of force centers (*NFC*), the time interval at which instantaneous coordinates are output (*tCoords*), the current snapshot index (*Si*), and the size of the simulation cell (*LBox*). The second line contains *NFC* triplets comprising the X, Y, and Z coordinates of each force center, respectively. The variable template for the output block is:

Integer, Real\*4, Integer, Real\*8

NFC × 3 × Real\*4

*Pressure Tensor* – filename: *fort.78*

This file is composed of an arbitrary number of two-line blocks. In each block, the first line contains a single value representing the time interval at which the stress tensor is calculated (*tStress*). The second line contains six values comprising the values of the symmetrized pressure tensor: σxx, σyy, σzz, σxy, σxy,σyz. The variable template for the output block is:

Real\*4

6 × Real\*4

*Instantaneous Velocities* – filename: *fort.79*

This file is composed of an arbitrary number of two-line blocks. In each block, the first line contains two values, the total number of force centers (*N*FC) and the time intervel at which instantaneous velocities are output (*tVeloc*). The second line contains *NFC* triplets comprising the X, Y, and Z components of the velocities (in m/s), respectively. The variable template for the output block is:

Integer, Real\*4

NFC × 3 × Real\*4

Trace file (fort.65)

The fort.65 file produces an ongoing trace of the average simulation quantities over the time window specified in *Line 4* of the ***mdrun.params*** file. The file format is straightforward, utilizing the following structure:

#-----------------------------------------------------------------------

# Running Averages

#

# **Time Temp Pressure Box Total E Hamiltonian**

# fs K Atm Ang Kcal/mol Kcal/mol

#-----------------------------------------------------------------------

*150.00 294.53 323.637 35.2207 -20839.15 -20367.43*

*300.00 293.17 208.734 35.2451 -20885.07 -20368.97*

*450.00 300.07 -82.525 35.2871 -20796.82 -20370.32*

*600.00 301.04 -275.374 35.3214 -20760.64 -20369.75*

*750.00 295.31 -212.401 35.3315 -20807.94 -20369.89*

*900.00 295.22 -2.149 35.3265 -20780.72 -20371.50*

*1050.00 305.55 -48.291 35.3183 -20699.03 -20370.61*

*1200.00 297.98 -43.759 35.3074 -20801.79 -20368.89*

Here, the values presented represent the average value over the preceding time period, with the units as indicated in the header. While a variety of different properties may be calculated and output to the trace file depending on interest, there is currently no way to change the output values without requiring manipulation of the underlying simulation code and recompilation of the program. The values output are those which are generally of most interest:

*Time* – Current simulation time in femtoseconds.

*Temp* – Average temperature over the preceding time interval, in Kelvin.

*Pressure* – Average pressure over the preceding time interval, in Atmospheres.

*Box* – Average *LB* over the preceding time interval, in Angstroms.

*Total E* – Average total energy over the preceding time interval, in kCal/mol.

*Hamiltonian* – Average Hamiltonian (including any extended contributions due to differing ensembles) over the preceding time interval, in kCal/mol.

1. Glenn J. Martyna, Michael L. Klein, and Mark Tuckerman, “Nosé-Hoover chains: The canonical ensemble via continuouse dynamics”, *J. Chem. Phys.* **97**(4), 2635 (1992). [↑](#endnote-ref--1)