

The GROMOS Software for (Bio)Molecular Simulation



Volume 4: Data Structures and Formats

January 9, 2021

Contents

Chapter 1. Introduction	4-1
Chapter 2. Block structure and title record of GROMOS files	4-3
Chapter 3. Topological information	4-5
3.1. Introduction	4-5
3.2. Molecular topology	4-6
3.3. Perturbation molecular topology	4-16
3.4. Atom-atom and distance-field distance restraints	4-23
3.5. Dihedral-angle restraints or constraints	4-27
3.6. 3J -coupling constant restraints	4-28
3.7. S^2 -order parameter restraining	4-29
3.8. Local-elevation coordinates	4-31
3.9. Local elevation umbrella sampling database file	4-32
3.10. Atomic friction coefficients	4-32
3.11. Position restraining or constraining atom specification list	4-33
3.12. B-factor restraining	4-33
3.13. Backwards compatibility with GROMOS96	4-34
Chapter 4. Configurational information	4-37
4.1. Introduction	4-37
4.2. Atomic coordinates	4-38
4.3. Atomic velocities	4-39
4.4. Atomic forces	4-40
4.5. Atomic stochastic integrals	4-40
4.6. Periodic box	4-41
4.7. Nose-Hoover chain thermostat variables	4-42
4.8. Roto-translational constraints reference variables	4-42
4.9. Perturbation data	4-42
4.10. Atom-atom distance restraints	4-43
4.11. 3J -coupling constant restraints	4-43
4.12. S^2 -order parameter restraints	4-44
4.13. Crystallographic restraints	4-45
4.14. Local-elevation data	4-45
4.15. Ball and stick local-elevation data	4-46
4.16. Time or step number data	4-49
4.17. Energies, pressure, volume and free-energy data	4-49
4.18. Atomic B-factors and positional fluctuations	4-54
4.19. Accelerated EDS parameter search data	4-55
4.20. Backwards compatibility with GROMOS96	4-56
Chapter 5. Molecular topology building blocks	4-57
5.1. Introduction	4-57
5.2. Separate molecules	4-57
5.3. Linking of building blocks	4-65
5.4. Other building blocks	4-66
5.5. End groups	4-67
5.6. Contents of the MTB file	4-67

Chapter 6. Interaction function parameters	4-69
6.1. Introduction	4-69
6.2. Mass atom types	4-69
6.3. Covalent bond-stretching interaction parameters	4-70
6.4. Covalent bond-angle bending interaction parameters	4-70
6.5. Improper dihedral-angle interaction parameters	4-70
6.6. Dihedral-angle torsional interaction parameters	4-71
6.7. Van der Waals interaction parameters and integer atom codes	4-71
6.8. Atomic charges and charge group codes	4-73
6.9. Excluded neighbours	4-73
6.10. Contents of the IFP file	4-73
Chapter 7. Library files for GROMOS++	4-75
7.1. Introduction	4-75
7.2. Interaction function parameter renumbering	4-75
7.3. Atomic naming conventions	4-76
7.4. Definition of file-names and joblists	4-77
7.5. Energy trajectory block definition	4-79
7.6. Hydrogen-bond donors and acceptors	4-79
7.7. Crystallographic transformations	4-80
7.8. NOE analysis	4-81
7.9. SASA implicit solvent model	4-83
7.10. DISICL angle, region and segment definitions	4-84
Chapter 8. Input file for MD++	4-87
Chapter 9. Output files for MD++	4-107
Chapter 10. Files accessed by MD++ for reading or writing	4-109
Chapter 11. Other non-GROMOS formats	4-115
Chapter 12. List of GROMOS blocknames	4-117
Chapter 13. Recommendations for standard input and output file names	4-121
Bibliography	4-i

Introduction

GROMOS knows different types of data and data files, which are described in this volume. Two types of information concerning a molecular system can be distinguished.

1. *Topological information*: data on the covalent structure, atomic masses, charges, van der Waals parameters, atom-atom distance restraints specification, 3J -value restraints specification, local-elevation coordinate specification, etc.
2. *Configurational information*: atomic coordinates and atomic coordinate dependent or related quantities, such as velocities and forces, atom-atom distances, dihedral angles, 3J -values, energies, size of the computational box, etc.

These two types of information are generally stored in separate files, since configurations change continuously during a simulation, whereas molecular topological data generally do not change. Both types of files, *topological files* and *configurational files*, for a specific molecular system are related through the requirement that in both the sequence of the quantities is the same, e.g.

1. sequence of atoms
2. sequence of atom-atom distance restraints
3. sequence of dihedral angle restraints
4. sequence of 3J -value restraints

This identity of sequence could be checked e.g. by comparing atom names occurring in topological files with those from the configurational files. However, in order to avoid dependence on naming conventions and to maintain maximum flexibility, this is not done in the GROMOS programs. When molecular information, such as residue numbers and names or atom sequence numbers or names, is present both in a topological file and in a configurational file of a molecular system, the program generally uses the data from the topological file and ignores the corresponding data on the configurational file.

The units of the quantities contained in the different files are all derived from the basic units: nm (length), ps (time), atomic mass units and electronic charge, leading in particular to $\text{kJ} \cdot \text{mol}^{-1}$ as unit of energies. The angles are always given in degrees in the files.

GROMOS data files have a block structure, which is defined in Chap. 2. Topological quantities, variables, blocks and files are described in Chap. 3. Configurational quantities, variables, blocks and files are described in Chap. 4. Two other types of data, molecular topology building blocks and interaction function parameters are described in Chap. 5 and Chap. 6. Library files to be used by the analysis programs of GROMOS++ are described in Chap. 7. Chap. 8 describes the MD++ input file.

Block structure and title record of GROMOS files

GROMOS files are composed of a sequence of blocks, which may be of different type. A block begins with a line (record) containing the *blockname* or *blockidentifier* beginning in the first position. The block ends with a line (record) containing the character string *END* beginning in the first position. A blockname or blockidentifier is a unique string of maximally 25 characters. It may not contain a # symbol in the first position and may not be an END string. Block names are given in upper case only. The currently defined blocknames and their functions are listed in Chap. 12.

Each input or output file of the program MD++, which executes a simulation, starts with a *Title block* (Blockname: TITLE), which may contain any character type of data and is meant to specify the contents of the file. When MD++ reads a file, the title record is always printed in order to check whether the wanted file has been assigned to a specific (reading) unit number. This convention is not followed by GROMOS++, which relies on file names rather than units, and ignores title blocks.

Generally, blocks may be listed in any order. However, when there are obvious dependences between data in different blocks, e.g. the definition of bond-angle types and sequence of bond-angles of a molecule, a specific order is required (the latter after the former).

Data files may contain *comment lines*, which may occur at any position and in any number. A comment line is recognized by the # symbol *in the first position* of the line. GROMOS++ also recognizes as comment any text following a # symbol anywhere on the line. In GROMOS, files are *written* using *fixed format* and are *read* using *free format*.

Topological information

3.1. Introduction

A *molecular topology* file contains information about the topology of a molecular system. In its simplest form it would contain lists of covalent bonds, angles, masses, charges, etc. for all the atoms in the molecular system. When the system contains topologically identical molecules, like water molecules in an aqueous solution or corresponding molecules in different asymmetric units in a unit cell in a crystal, these atom lists would contain redundant information. For MD++ and GROMOS++ the topology has to be specified explicitly for all identical solute units. Since a solvent generally consists of simple molecules like H₂O or CCl₄, it would generally be advantageous to avoid the overhead of handling the possibility of occurrence of internal dihedral angle degrees of freedom, non-bonded interactions, etc. within a solvent molecule. Therefore, a distinction is made between a general part (solutes) and a more restricted part (solvent) of a molecular topology file.

For historical reasons the *general part* of a molecular topology file is denoted by the notation “*solute*” *molecular topology*, although it may contain any collection of molecules including solvent molecules. The *restricted part* of a molecular topology file is denoted by the notation “*solvent*” *molecular topology*. In general, this part contains topological data on a single type of solvent molecule, unless a solvent molecule does not fit with the following *restrictions*:

- a solvent molecule must be rigid: no internal interactions like bond-stretching, bond-angle bending, (improper) dihedral torsion and non-bonded interactions are allowed,
- the internal structure of a solvent molecule is maintained by application of distance constraint forces between its atoms,
- a solvent molecule consists of one charge group, the position of the first atom of a solvent molecule is taken to represent the position of this charge group,
- a solvent molecule corresponds to a single “temperature group” and a single “pressure (virial) group”,
- the residue or molecule name cannot be specified, it is predefined as SOL,
- position restraining should only be applied to the first atom of a solvent molecule,
- fixed position constraints cannot be applied to solvent atoms,
- solvent parameters cannot be changed using a molecular topology perturbation file for obtaining free energy differences.

If a solvent molecule does not comply with these rules, its topological data must be included in the general or solute part of the molecular topology file.

A molecular topology file often contains fewer atoms than a corresponding configuration file. Let us assume that the former contains a set of molecules forming a “solute” of NRP atoms and a solvent molecule with NRAM atoms. In order to *match* this *molecular topology file*, a *configuration file* must contain the following sequence of atoms (for each block of atomic quantities):

1. the atomic coordinates or related quantities of the NRP “solute” atoms,
2. if the molecular system contains NSM solvent molecules, the atomic coordinates or related quantities of the NSM*NRAM solvent atoms.

Solvent coordinates always appear after solute coordinates in the various blocks of configuration files. All solute atoms should be included explicitly in the molecular topology file. The solvent parts of the molecular topology file are to be chosen as the smallest topologically identical units of each type.

In Sec. 3.2 the content of a molecular topology file is specified. This information is always kept in formatted form. In some applications of GROMOS, like calculating the free energy difference between two different states A and B of a system, it is required to change the molecular topology of the system from one corresponding to state A to another one corresponding to state B. In Sec. 3.3 the way a perturbation (change from A to B) of a molecular topology is to be specified will be discussed. The topological specification of atom-atom distance restraints and distance-field restraints is described in Sec. 3.4. The topological specification of dihedral angle restraints or constraints is described in Sec. 3.5. The topological definition of 3J -value restraints is described in Sec. 3.6. The topological definition of S^2 -order parameter restraints is described in Sec. 3.7. The topological specification of coordinates to be used in the local-elevation search technique is described in Sec. 3.8. The following three sections contain atomic property specifications: friction coefficients and position restraining or fixing indicators (Sec. 3.9-Sec. 3.11).

3.2. Molecular topology

A molecular topology is characterized by some or all of the following quantities, which are stored in a molecular topology file.

FPEPSI	$(4\pi\epsilon_0)^{-1}$, ϵ_0 = permittivity of vacuum
HBAR	$\hbar = h/(2\pi)$, h = Planck's constant
SPDL	c = speed of light
BOLTZ	k_B = Boltzmann's constant
TPVER	real number characterizing the version of the molecular topology
NRATT	number of (van der Waals) atom types
TYPE[1..NRATT]	names of the different atom types as a function of the integer atom code that defines an atom type (at most 5 characters)
NRAA2	number of residues in a solute
AANM[1..NRAA2]	residue names as a function of the residue sequence number (at most 5 characters)
NRP	number of solute atoms
MRES[1..NRP]	residue sequence number of solute atoms (\leq NRAA2)
PANM[1..NRP]	atom name of solute atoms
IAC[1..NRP]	integer atom code of solute atoms, determining the type of van der Waals interaction of an atom (\leq NRATT)
MASS[1..NRP]	mass of solute atoms
CG[1..NRP]	charge of solute atoms
CGC[1..NRP]	Atomic charge group codes. The last atom of a charge group is defined by CGC=1, the others must have CGC=0
INE[1..NRP]	number of neighbour atoms that are excluded from the non-bonded interaction with a solute atom
JNE[1..NRP] [1..INE[]]	excluded neighbours (solute, \leq NRP); sequence numbers J of atoms that are excluded from the non-bonded interaction with the atom with sequence number I; it is assumed that $I < J$ and that the J's appear in ascending order
INE14[1..NRP]	number of third-neighbour atoms of solute atoms, for which special 1-4 van der Waals interaction parameters are used when evaluating the non-bonded interaction

JNE14[1..NRP][1..INE14[]] third neighbours (solute, \leq NRP); sequence numbers J of atoms for which the 1-4 van der Waals parameters are used when calculating the non-bonded interaction with the atom with sequence number I; it is assumed that $I < J$ and that the J's appear in ascending order

NCGB[L] number of coarse-grained regions

NRCGF[1..NCGB] sequence number of the first coarse-grained solute particle in range

NRCGL[1..NCGB] sequence number of the last coarse-grained solute particle in range

MSCAL[1..NCGB] scaling factor for pressure correction of a coarse-grained region

NBTY number of covalent bond types

CB[1..NBTY] force constant of the bond-stretching term of the interaction as a function of the bond-type code, based on a quartic potential

CHB [1..NBTY] force constant of the bond-stretching term of the interaction as a function of the bond-type code, based on a harmonic potential

BO[1..NBTY] bond length at minimum energy of the bond-stretching term as a function of the bond-type code

NBONH number of bonds involving H-atoms in the solute

IBH, JBH[1..NBONH] atom sequence numbers of the atoms forming a bond i-j as a function of the bond sequence number (\leq NRP), i is always smaller than j

ICBH[1..NBONH] bond-type code as a function of the bond sequence number (\leq NBTY)

NBON number of bonds *NOT* involving H-atoms in the solute

IB, JB[1..NBON] atom sequence numbers of the atoms forming a bond i-j as a function of the bond sequence number (\leq NRP), i is always smaller than j

ICB[1..NBON] bond-type code as a function of the bond sequence number (\leq NBTY)

NBONDP number of bonds involving coarse grained particles in the solute

IBDP, JBDP[1..NBONDP] sequence numbers of the coarse grained particles forming a bond i-j as a function of the bond sequence number (\leq NRP)

ICBC[1..NBON] bond-type code as a function of the bond sequence number (\leq NBTY)

NTTY number of bond-angle types

CT[1..NTTY] force constant of the bond-angle bending term of the interaction as a function of the bond-angle type code, based on a potential harmonic in the angle cosine

CHT[1..NTTY] force constant of the bond-angle bending term of the interaction as a function of the bond-angle type code, based on a potential harmonic in the angle (in energy units per degree²)

TO[1..NTTY] bond angle (in degrees) at minimum energy of the bond-angle bending term as a function of the bond-angle type code; upon reading a molecular topology file by MD++, the bond angle is converted from degrees to radians; this conversion is not performed in GROMOS++

NTHEH number of bond-angles involving H-atoms in the solute

ITH, JTH, KTH [1.. NTHEH] atom sequence numbers of the atoms forming a bond-angle i-j-k as a function of the bond-angle sequence number (\leq NRP), i is always smaller than k

ICTH [1.. NTHEH] bond-angle type code as a function of the bond-angle sequence number (\leq NTTY)

NTHE number of bond-angles *NOT* involving H-atoms in the solute

IT, JT, KT [1.. NTHE] atom sequence numbers of the atoms forming a bond-angle i-j-k as a function of the bond-angle sequence number (\leq NRP), i is always smaller than k

ICT [1.. NTHE] bond-angle type code as a function of the bond-angle sequence number (\leq NTTY)

NQTY number of improper (harmonic) dihedral-angle types

CQ [1.. NQTY] force constant of the harmonic improper dihedral term of the interaction as a function of the improper dihedral-angle type code (in energy units per degree²); upon reading a molecular topology file by MD++, the force constant is converted to energy per rad²; this conversion is not performed by GROMOS++

QO [1.. NQTY] improper dihedral (in degrees) at minimum energy of the harmonic improper dihedral term as a function of the improper dihedral-angle type code; upon reading a molecular topology file by MD++, the improper dihedral angle is converted from degrees to radians; this conversion is not performed by GROMOS++

NQHIH number of improper dihedrals involving H-atoms in the solute

IQH, JQH, KQH, LQH [1.. NQHIH] atom sequence numbers of the atoms forming improper dihedral i-j-k-l as a function of the improper dihedral sequence number (\leq NRP), j is always smaller than k

ICQH [1.. NQHIH] improper dihedral type code as a function of the improper dihedral sequence number (\leq NQTY)

NQHI number of improper dihedrals *NOT* involving H-atoms in the solute

IQ, JQ, KQ, LQ [1.. NQHI] atom sequence numbers of the atoms forming improper dihedral i-j-k-l as a function of the improper dihedral sequence number (\leq NRP), j is always smaller than k

ICQ [1.. NQHI] improper dihedral type code as a function of the improper dihedral sequence number (\leq NQTY)

NPTY number of (trigonometric) dihedral-angle types

CP [1.. NPTY] force constant of the trigonometric dihedral term of the interaction as a function of the dihedral-angle type code

PD [1.. NPTY] phase-shift angle (in degrees) of the trigonometric dihedral term as a function of the dihedral-angle type code; upon reading a molecular topology file by MD++, the phase-shift angle is converted from degrees to radians; this conversion is not performed in GROMOS++

NP [1.. NPTY] multiplicity of the trigonometric dihedral term as a function of the dihedral-angle type code (1, 2, 3, 4, 5, or 6)

NPHIH number of dihedrals involving H-atoms in the solute

IPH, JPH, KPH, LPH [1.. NPHIH] atom sequence numbers of the atoms forming dihedral i-j-k-l as a function of the dihedral sequence number (\leq NRP), j is always smaller than k

ICPH[1..NPHIH]	dihedral type code as a function of the dihedral sequence number (\leq NPTY)
NPHI	number of dihedrals <i>NOT</i> involving H-atoms in the solute
IP,JP,KP,LP[1..NPHI]	atom sequence numbers of the atoms forming dihedral i-j-k-l as a function of the dihedral sequence number (\leq NRP), j is always smaller than k
ICP[1..NPHI]	dihedral type code as a function of the dihedral sequence number (\leq NPTY)
NPPCH	number of cross-dihedrals involving H-atoms in the solute
APH,BPH,CPH,DPH,EPH,FPH,GPH,HPH[1..NPPCH]	atom sequence numbers of the atoms forming cross-dihedrals a-b-c-d and e-f-g-h as a function of the dihedral sequence numbers of the separate dihedrals (\leq NRP), a,b,c,d are always smaller or equal to e,f,g,h respectively
ICCH[1..NPPCH]	dihedral type code as a function of the cross-dihedral sequence number (\leq NPTY)
NPPC	number of cross-dihedrals <i>NOT</i> involving H-atoms in the solute
AP,BP,CP,DP,EP,FP,GP,HP[1..NPPC]	atom sequence numbers of the atoms forming cross-dihedrals a-b-c-d and e-f-g-h as a function of the dihedral sequence numbers of the separate dihedrals (\leq NRP), a,b,c,d are always smaller or equal to e,f,g,h respectively
ICC[1..NPPC]	dihedral type code as a function of the cross-dihedral sequence number (\leq NPTY)
NRATT2	number of unique pairwise combinations of atom types (=NRATT*(NRATT +1)/2)
C12[1..NRATT2]	coefficient of the $1/r^{12}$ term in the non-bonded interaction as a function of the occurring pair codes; so, the sequence of atom pairs with integer atom codes ranging from 1 to NRATT is: 1-1, 1-2, ...,1-NRATT, 2-2, 2-NRATT, ..., NRATT-NRATT
C6[1..NRATT2]	coefficient of the $-1/r^6$ term in the non-bonded interaction as a function of the occurring pair codes
CS12[1..NRATT2]	coefficient of the $1/r^{12}$ term in the 1-4 non-bonded interaction between third-neighbour atoms as a function of the occurring pair codes
CS6[1..NRATT2]	coefficient of the $-1/r^6$ term in the 1-4 non-bonded interaction between third-neighbour atoms as a function of the occurring pair codes
NPPOL	number of polarisable solute atoms (\leq NRP)
IPOLP[1..NPPOL]	atom sequence numbers of the polarisable solute atoms (\leq NRP)
ALPP[1..NPPOL]	polarisabilities of solute atoms IPOLP[1..NPPOL]
QPOLP[1..NPPOL]	size of charge-on-spring connected to polarisable solute atoms IPOLP[1..NPPOL]
ENOTP[1..NPPOL]	damping level for polarisation of solute atoms IPOLP[1..NPPOL]
EPP[1..NPPOL]	damping parameter for polarisation of solute atoms IPOLP[1..NPPOL]
NSPM	number of all separate (covalently linked) molecules within the solute topology (e.g. separate protein chains, co-solute molecules, counterions, co-solvent molecules)
NSP[1..NSPM]	atom sequence number of the last atom of the successive submolecules (\leq NRP)

NSTM	number of temperature atom groups (used to separate translational from internal-plus-rotational velocity components for kinetic energy evaluation and thermostating) within the solute topology
NST[1..NSTM]	atom sequence number of the last atom of the successive temperature atom groups (\leq NRP)
NSVM	number of pressure (virial) atom groups (used to define a group-based pressure) within the solute topology
NSV[1..NSVM]	atom sequence number of the last atom of the successive pressure (virial) atom groups (\leq NRP)
NLJEX	number of LJ-exceptions
ILJEX, JLJEX[1..NLJEX]	atom sequence numbers of atoms i and j to interact with special LJ-interactions given by LJ-exceptions, i is always smaller than j
LJEXC12[1..NLJEX]	coefficient of the $1/r^{12}$ term in the non-bonded interaction for the corresponding atom pair
LJEXC6[1..NLJEX]	coefficient of the $1/r^6$ term in the non-bonded interaction for the corresponding atom pair
NRAM	number of atoms per solvent molecule
ANMS[1..NRAM]	atom name of solvent atoms
IACS[1..NRAM]	integer atom code of solvent atoms determining the type of van der Waals interaction of an atom (\leq NRATT)
WMASS[1..NRAM]	mass of solvent atoms
CGS[1..NRAM]	charge of solvent atoms
NCONS	number of distance constraints within a solvent molecule
ICONS, JCONS[1..NCONS]	atom sequence numbers of the atoms forming the constraint i-j as a function of the constraint sequence number (\leq NRAM), i is always smaller than j
CONS[1..NCONS]	constraint length as a function of the constraint sequence number
NVPOL	number of polarisable solvent atoms (\leq NRAM)
IPOLV[1..NVPOL]	atom sequence number of the polarisable solvent atoms (\leq NRAM)
ALPV[1..NVPOL]	polarisabilities of solvent atoms IPOLV[1..NVPOL]
QPOLV[1..NVPOL]	size of charge-on-spring connected to polarisable solvent atoms IPOLV[1..NVPOL]
ENOTV[1..NVPOL]	damping level for polarisation of solvent atoms IPOLV[1..NVPOL]
EPV[1..NVPOL]	damping parameter for polarisation of solvent atoms IPOLV[1..NVPOL]
NRSASAA	number of atoms to be considered for SASA implicit solvent interaction function
ISASA[1..NRSASAA]	atom sequence numbers of the atoms to be included in the SASA implicit solvent interaction function
RADI[1..NRSASAA]	atomic radii of the SASA atoms
PI[1..NRSASAA]	atom type-specific parameters P in SASA calculation

SIGMAI[1..NRSASAA] atom type-specific scaling parameters for SASA energy term

The blocks of a *molecular topology file* are (apart from the *Title block*) the following:

Physical constants block

Blockname: PHYSICALCONSTANTS

```
        WRITE (unit,12) FPEPSI
        WRITE (unit,12) HBAR
        WRITE (unit,12) SPDL
        WRITE (unit,12) BOLTZ
12  FORMAT (E15.7)
```

This block replaces the TOPPHYSCON block of GROMOS96.

Version block

Blockname: TOPVERSION

```
        WRITE (unit,13) TPVER
13  FORMAT (F3.1)
```

The version number expected by GROMOS is 2.0.

Van der Waals atom type sequence and name block

Blockname: ATOMTYPENAME

```
        WRITE (unit,14) NRATT
        DO 10 K=1, NRATT
10  WRITE (unit,15) TYPE[K]
14  FORMAT (5I5)
15  FORMAT (A5)
```

Solute residue sequence and name block

Blockname: RESNAME

```
        WRITE (unit,14) NRAA2
        DO 10 K=1, NRAA2
10  WRITE (unit,15) AANM[K]
```

Solute atom information block

Blockname: SOLUTEATOM

```
        WRITE (unit,14) NRP
        DO 10 I=1, NRP
        WRITE (unit,17) I, MRES[I], PANM[I], IAC[I], MASS[I], CG[I],
                        CGC[I], INE[I], JNE[I][K], K=1, INE[I])
10  WRITE (unit,18) INE14[I],(JNE14[I][K], K=1, INE14[I])
17  FORMAT (2I5,1X,A5,I4,2F11.5,2I4,6I5)
18  FORMAT (46X,I4,6I5)
```

If $INE[I] > 6$, then the remaining JNE values are written on the next line using 16I5 as format. Likewise for the JSNE14 values if $INE14[I] > 6$.

Coarse grained solute information block

Blockname: CGSOLUTE

```
        WRITE (unit,14) NCGB
        DO 10 K=1, NCGB
10     WRITE (unit,15) NRCGF[K], NRCGL[K], MSCAL[K]
```

Lennard-Jones interaction exception block

Blockname: LJEXCEPTIONS

```
        WRITE (unit,14) NLJEX
        DO 10 N=1, NLJEX
10     WRITE (unit,19) ILJEX[N], JLJEX[N], LJEXC12[N], LJEXC6[N]
19     FORMAT (2I5,2F15.7)
```

Bond interaction type block

Blockname: BONDSTRETCHTYPE

```
        WRITE (unit,14) NBTY
        DO 10 N=1, NBTY
10     WRITE (unit,19) CB[N], CHB[N], BO[N]
19     FORMAT (3F15.7)
```

The GROMOS96 BONDTYPE block is still accepted, it only contains force constants for the quartic interaction form, the HARBONDTYPE block contains only force constants for the harmonic form. If the BONDSTRETCHTYPE block is present, the other two are not allowed.

Solute bonds involving H-atoms block

Blockname: BONDH

```
        WRITE (unit,14) NBONH
        DO 10 N=1, NBONH
10     WRITE (unit,14) IBH[N], JBH[N], ICBH[N]
```

Solute bonds NOT involving H-atoms block

Blockname: BOND

```
        WRITE (unit,14) NBON
        DO 10 N=1, NBON
10     WRITE (unit,14) IB[N], JB[N], ICB[N]
```

Coarse grained solute bonds block

Blockname: BONDDP

```
        WRITE (unit,14) NBONCG
        DO 10 N=1, NBONCG
10     WRITE (unit,14) IBCG[N], JBCG[N], ICBCG[N]
```

Solute distance constraints

Blockname: CONSTRAINT

```
        WRITE (unit,14) NCON
```

```

      DO 10 N=1, NCON
10  WRITE (unit,14) IC[N], JC[N], ICC[N]

```

Bond angle interaction type block

Blockname: BONDANGLEBENDTYPE

```

      WRITE (unit,14) NTTY
      DO 10 N=1, NTTY
10  WRITE (unit,19) CT[N], CHT[N], TO[N]
19  FORMAT (3F15.7)

```

The GROMOS96 BONDANGLETYPE block is still accepted, it only contains force constants for the cosine harmonic interaction form, the HARMBONDANGLETYPE block contains only force constants for the harmonic form. If the BONDANGLEBENDTYPE block is present, the other two are not allowed.

Solute bond angles involving H-atoms block

Blockname: BONDANGLEH

```

      WRITE (unit,14) NTHEH
      DO 10 N=1, NTHEH
10  WRITE (unit,14) ITH[N], JTH[N], KTH[N], ICTH[N]

```

Solute bond angles NOT involving H-atoms block

Blockname: BONDANGLE

```

      WRITE (unit,14) NTHE
      DO 10 N=1, NTHE
10  WRITE (unit,14) IT[N], JT[N], KT[N], ICT[N]

```

Improper (harmonic) dihedral angle interaction type block

Blockname: IMPDIHEDRALTYPE

```

      WRITE (unit,14) NQTY
      DO 10 N=1, NQTY
10  WRITE (unit,19) CQ[N], QO[N]
19  FORMAT (3F15.7)

```

Solute improper dihedrals involving H-atoms block

Blockname: IMPDIHEDRALH

```

      WRITE (unit,14) NQHIH
      DO 10 N=1, NQHIH
10  WRITE (unit,14) IQH[N], JQH[N], KQH[N], LQH[N], ICQH[N]

```

Solute improper dihedrals NOT involving H-atoms block

Blockname: IMPDIHEDRAL

```

      WRITE (unit,14) NQHI
      DO 10 N=1, NQHI
10  WRITE (unit,14) IQ[N], JQ[N], KQ[N], LQ[N], ICQ[N]

```

Proper (trigonometric) dihedral angle interaction type block

Blockname: TORSDIHEDRALTYPE

```
        WRITE (unit,14) NPTY
        DO 10 N=1, NPTY
10     WRITE (unit,20) CP[N], PDL[N], NP[N]
20     FORMAT (2F10.5,I5)
```

The GROMOS96 DIHEDRALTYPE block is still accepted, it expects cosine values for the phase shifts allowing only values of -1 and 1 (0 or 180°). If both blocks are specified, only the TORSDIHEDRALTYPE block is read in.

Solute dihedrals involving H-atoms block

Blockname: DIHEDRALH

```
        WRITE (unit,14) NPHIH
        DO 10 N=1, NPHIH
10     WRITE (unit,14) IPH[N], JPH[N], KPH[N], LPH[N], ICPH[N]
```

Solute dihedrals NOT involving H-atoms block

Blockname: DIHEDRAL

```
        WRITE (unit,14) NPHI
        DO 10 N=1, NPHI
10     WRITE (unit,14) IP[N], JP[N], KP[N], LP[N], ICP[N]
```

Solute cross-dihedrals involving H-atoms block

Blockname: CROSSDIHEDRALH

```
        WRITE (unit,14) NPPCH
        DO 10 N=1, NPPCH
10     WRITE (unit,14) APH[N], BPH[N], CPH[N], DPH[N], EPH[N],
                FPH[N], GPH[N], HPH[N], ICCH[N]
```

Solute cross-dihedrals NOT involving H-atoms block

Blockname: CROSSDIHEDRAL

```
        WRITE (unit,14) NPPC
        DO 10 N=1, NPPC
10     WRITE (unit,14) AP[N], BP[N], CP[N], DP[N], EP[N],
                FP[N], GP[N], HP[N], ICC[N]
```

Van der Waals (Lennard-Jones) interaction block

Blockname: LJPARAMETERS

```
        NRATT2 = NRATT*(NRATT+1)/2
        WRITE (unit,14) NRATT2
        DO 10 J=1, NRATT
        DO 9 I=1, J
  9     WRITE (unit,16) I, J, C12[I,J], C6[I,J], CS12[I,J], CS6[I,J]
10     CONTINUE
16     FORMAT (2I5,4E15.7)
```

Coarse grain (Lennard-Jones) interaction block

Blockname: CGPARAMETERS

```
NRATT2 = NRATT*(NRATT+1)/2
WRITE (unit,14) NRATT2
DO 10 J=1, NRATT
DO 9 I=1, J
9 WRITE (unit,16) I, J, C12[I,J], C6[I,J]
10 CONTINUE
16 FORMAT (2I5,2E15.7)
```

Solute polarisation specification block (md++ only, optional)

Blockname: SOLUTEPOLARISATION

```
WRITE (unit,14) NPPOL
DO 10 N=1, NPPOL
10 WRITE (unit,99) IPOLP[N], ALPP[N], QPOLP[N], ENOTP[N], EPP[N]
99 FORMAT (I5,4F15.7)
```

Separate solute molecules specification block

Blockname: SOLUTEMOLECULES

```
WRITE (unit,14) NSPM
DO 10 N=1, NSPM
10 WRITE (unit,14) NSP[N]
```

Temperature atom groups specification block

Blockname: TEMPERATUREGROUPS

```
WRITE (unit,14) NSTM
DO 10 N=1, NSTM
10 WRITE (unit,14) NST[N]
```

Pressure groups specification block

Blockname: PRESSUREGROUPS

```
WRITE (unit,14) NSVM
DO 10 N=1, NSVM
10 WRITE (unit,14) NSV[N]
```

Solvent atom information block

Blockname: SOLVENTATOM

```
WRITE (unit,14) NRAM
DO 10 I=1, NRAM
10 WRITE (unit,21) I, ANMS[I], IACS[I], WMASS[I], CGS[I]
21 FORMAT (I5,1X,A5,I4,2F11.5)
```

Solvent distance constraint block

Blockname: SOLVENTCONSTR

```
WRITE (unit,14) NCONS
```

```

DO 10 K=1, NCONS
10 WRITE (unit,22) ICONS[K], JCONS[K], CONS[K]
22 FORMAT (2I5,F15.7)

```

Solvent polarisation specification block (optional)

Blockname: SOLVENTPOLARISATION

```

WRITE (unit,14) NVPOL
DO 10 N=1, NVPOL
10 WRITE (unit,99) IPOLV[N], ALPV[N], QPOLV[N], ENOTV[N], EPV[N]

```

SASA implicit solvent model parameter block

Blockname: SASAPARAMETERS

```

WRITE (unit,20) NRSASAA
DO 10 I=1, NRSASAA
10 WRITE (unit,21) ISASA[I], RADI[I], PI[I], SIGMAI[I]
20 FORMAT (I5)
21 FORMAT (I6,3X,F5.3,3X,F5.3,3X,F8.3)

```

Examples of molecular topology files are named:

*.top

3.3. Perturbation molecular topology

When simulating a molecular system or when analyzing a set of conformations of a molecule, the molecular topology file of the system remains unchanged. This is the rationale for separating topological and force field information resident in a molecular topology file from conformational information resident in configurational files. If a change of topological data or force field parameters is required, a new changed molecular topology file has to be generated by one of the molecular topology building or conversion programs.

However, when applying the thermodynamic integration formalism based on the coupling parameter (λ) approach in order to determine the difference in free energy between two states A and B of a molecular system, the molecular topology (Hamiltonian) of the system becomes a function of the coupling parameter λ such that it may change in a continuous way from the one corresponding to state A to the one corresponding to state B or vice versa. In general the difference between state B and state A is limited to a restricted part of the system, that is, a few tens of atoms. Therefore, this difference is represented by a *perturbation molecular topology*, which contains information on how to change or perturb the molecular topology of state A in order to obtain the one of state B.

The implementation of the parametrisation of the Hamiltonian of a molecular system in terms of a parameter λ has been described in Sec. 2-14.2. It has been implemented in MD++. Note that the GROMOS++ program *pt_top* can be used to merge a topology (A) and a perturbation topology (B-A) into a new topology (B). Similarly, GROMOS++ program *make_pt_top* can be used to create the perturbation topology (B-A) from the specified topologies (A) and (B). The contents of the file containing the perturbation Hamiltonian or molecular topology is described below. Here, a few comments are given:

1. The *molecular topology* that is read, the unperturbed one, corresponds to *state A*.
2. The value $\lambda = RLAM = 0$ corresponds to *state A* of the system (*unperturbed* molecular topology); the value $\lambda = RLAM = 1$ corresponds to *state B* of the system (*perturbed* molecular topology).
3. Since atoms cannot be created or destroyed, only their interaction with other atoms can be modified or perturbed. Thus, the unperturbed topology corresponding to *state A* *must contain all atoms* involved in the perturbation *as either real or dummy* (i.e. non-interacting) *atoms*. So, state B has the same number of atoms as state A.

4. The perturbation of non-bonded interaction is specified by giving the NJLA atom sequence number of the perturbed atoms (JLA) and the integer atom codes (IAC(A), IAC(B)), masses (MASS(A), MASS(B)) and charges (CHARGE(A), CHARGE(B)) in both states A and B. The *force field parameters for state A* given in the perturbation molecular topology must not necessarily match those given in the (unperturbed) molecular topology. In the perturbation calculation, interactions in state A will be described according to state A given in the perturbation topology and a warning will be printed if state A in the perturbed topology does not match state A in the unperturbed topology.
5. The change from state A to state B may involve the *breaking or formation of a covalent bond between two atoms*. In that case, the excluded neighbours and the third neighbours of these atoms will be different in state A and in state B. The type of interaction, i.e. normal interaction, 1-4 or third-neighbour interaction, must be changeable. The standard non-bonded interaction subroutines only allow for a continuous change from one integer atom code (IAC(A)) to another (IAC(B)), but not for a change of type in the sense of normal, third-neighbour or excluded-neighbour interaction. A change of type is implemented by specifying the NEB pairs of atoms (IEB, JEB) for which the type is to be changed when moving from state A to state B. The perturbation molecular topology file contains the variables IETA and IETB for each pair, denoting which of the three types of interaction is applicable in state A and in state B. The interaction for these specified atom pairs is evaluated in special subroutines in MD++. In order to avoid double counting, *all these specified pairs must be excluded atom pairs in the unperturbed molecular topology* (state A).
6. In the most common case, all interactions within the molecular system are made λ -dependent. However, in special cases, one may wish to restrict the λ -dependence to a specified subset of interactions. This can be done by defining individual λ values per interaction which can be different for interactions within or between every energy group. The individual λ values are defined as a polynomial function of order 4 of the overall λ value (see Sec. 2-14.4).
7. In a number of applications (e.g. creation or annihilation of atoms by conversion from or into a dummy, or free-energy extrapolation from an unphysical reference state) it is useful to make perturbed interactions soft. This is achieved by a modification of the λ -dependent Lennard-Jones and electrostatic interaction functions through the introduction of two corresponding soft-core parameters $\alpha_{LJ}(I, J)$ and $\alpha_{EL}(I, J)$. These parameters are calculated from atomic soft-core parameters $\alpha_{LJ}(I, I)$ and $\alpha_{EL}(I, I)$ using the combination rules described in Sec. 2-14.2.8.
8. The perturbation of the bond-stretching, bond-angle bending, improper dihedral or dihedral interaction terms is specified by giving the sequence numbers of the atoms involved and the type codes determining force field parameters in state A as well as in state B. The *force field parameters for state A* given in the perturbation molecular topology must not necessarily match those given in the (unperturbed) molecular topology. In the perturbation calculation, interactions in state A will be described according to state A given in perturbation topology and a warning will be printed if state A in the perturbed topology does not match the unperturbed topology. The occurrence of multiple force field terms involving the same atoms and the same type code in the unperturbed topology is not allowed in this case.
9. For bond stretching, bond-angle bending and improper dihedrals a soft potential energy function can be chosen to reduce numerical instabilities when force constants are being reduced to 0. Apart from the regular type codes, determining the force field parameters for states A and B, a type code of 0 can be given for either state A or state B, indicating an interaction with a force constant of 0. An additional softness parameter (α_b , α_θ or α_ξ , respectively) is added to the definition of the perturbed interaction (see Sec. 2-14.2.2).
10. Note that the *units* of the perturbation molecular topology file *must match* the units of the unperturbed molecular topology file.
11. For some GROMOS++ programs and for enveloping distribution sampling (EDS) in MD++, it is convenient to handle multiple perturbation topologies simultaneously. For the non-bonded interactions this is implemented in a special MPERTATOMS block in which only the interaction parameters IACB and charges CGB for state B are specified for different perturbations. The GROMOS++ program *pt_top* can convert a *multiple perturbation topology file* into a perturbation topology file.

A perturbation molecular topology file is characterized by the following quantities:

IBSG, JBSG[1..NBONSG] atom sequence numbers of the atoms forming the perturbed bonds i-j (\ll NRP)

ICBSA[1..NBONSG] bond-type code corresponding to bond-stretching interaction term in state A (\ll NBTY)

ICBSB[1..NBONSG] as ICBSA, but for state B (\ll NBTY)

ALB[1..NBONSG] softness parameter for soft harmonic bond

NTHEHG number of perturbed bond angles involving H-atoms (\ll NTHEH)

ITHG, JTHG, KTHG[1..NTHEHG] atom sequence numbers of the atoms forming the perturbed bond angle i-j-k (\ll NRP)

ICTHA[1..NTHEHG] bond-angle type code corresponding to bond-angle bending interaction term in state A (\ll NTTY)

ICTHB[1..NTHEHG] as ICTHA, but for state B (\ll NTTY)

NTHEG number of perturbed bond angles NOT involving H-atoms (\ll NTHE)

ITG, JTG, KTG[1..NTHEG] atom sequence numbers of the atoms forming the perturbed bond angle i-j-k (\ll NRP)

ICTA[1..NTHEG] bond-angle type code corresponding to bond-angle bending interaction term in state A (\ll NTTY)

ICTB[1..NTHEG] as ICTA, but for state B (\ll NTTY)

NTHESG number of perturbed bond angles with a soft potential energy function (\ll NTHE)

ITSG, JTSG, KTSG[1..NTHESG] atom sequence numbers of the atoms forming the perturbed bond angle i-j-k (\ll NRP)

ICTSA[1..NTHESG] bond-angle type code corresponding to bond-angle bending interaction term in state A (\ll NTTY)

ICTSB[1..NTHESG] as ICTSA, but for state B (\ll NTTY)

ALA[1..NTHESG] softness parameter for soft bond angle

NQHIG number of perturbed improper (harmonic) dihedrals involving H-atoms (\ll NQHIIH)

IQHG, JQHG, KQHG, LQHG[1..NQHIG] atom sequence numbers of the atoms forming the perturbed improper (harmonic) dihedral i-j-k-l (\ll NRP)

ICQHA[1..NQHIG] improper dihedral type code corresponding to improper-dihedral interaction term in state A (\ll NQTY)

ICQHB[1..NQHIG] as ICQHA, but for state B (\ll NQTY)

NQHIG number of perturbed improper (harmonic) dihedrals NOT involving H-atoms (\ll NQHI)

IQG, JQG, KQG, LQG[1..NQHIG] atom sequence numbers of the atoms forming the perturbed improper (harmonic) dihedral i-j-k-l (\ll NRP)

ICQA[1..NQHIG] improper-dihedral type code corresponding to improper-dihedral interaction term in state A (\ll NQTY)

ICQB[1..NQHIG] as ICQA, but for state B (\leq NQTY)
 NQHISG number of perturbed improper (harmonic) dihedrals with a soft potential energy function (\leq NQHI)
 IQSG, JQSG, KQSG, LQSG[1..NQHISG] atom sequence numbers of the atoms forming the perturbed improper (harmonic) dihedral i-j-k-l (\leq NRP)
 ICQSA[1..NQHISG] improper-dihedral type code corresponding to improper-dihedral interaction term in state A (\leq NQTY)
 ICQSB[1..NQHISG] as ICQSA, but for state B (\leq NQTY)
 ALI[1..NQHISG] softness parameter for improper dihedral
 NPHIHG number of perturbed (trigonometric) dihedrals involving H-atoms (\leq NPHIH)
 IPHG, JPHG, KPHG, LPHG[1..NPHIHG] atom sequence numbers of the atoms forming the perturbed (trigonometric) dihedral i-j-k-l (\leq NRP)
 ICPHA[1..NPHIHG] dihedral-type code corresponding to trigonometric dihedral interaction term in state A (\leq NPTY)
 ICPHB[1..NPHIHG] as ICPHA, but for state B (\leq NPTY)
 NPHIG number of perturbed (trigonometric) dihedrals NOT involving H-atoms (\leq NPHI)
 IPG, JPG, KPG, LPG[1..NPHIG] atom sequence numbers of the atoms forming the perturbed (trigonometric) dihedral i-j-k-l (\leq NRP)
 ICPA[1..NPHIG] dihedral-type code corresponding to trigonometric dihedral interaction term in state A (\leq NPTY)
 ICPB[1..NPHIG] as ICPA, but for state B (\leq NPTY)
 NPOLG number of perturbed polarisabilities of perturbed atoms with atom sequence number JLA (NPOLG \leq NJLA)
 ALPA[1..NPOLG] polarisability of perturbed atoms in state A
 ENOTA[1..NPOLG] damping level for polarisation of perturbed atoms in state B
 ALPB[1..NPOLG] polarisability of perturbed atoms in state A
 ENOTB[1..NPOLG] damping level for polarisation of perturbed atoms in state B

The blocks of a *perturbation molecular topology file* are (apart from the *Title block*) the following:

Perturbed atom information block
 Blockname: PERTATOMPARAM

```

WRITE (unit,14) NJLA
DO 10 N=1, NJLA
10 WRITE (unit,23) NR[N],RES[N],NAME[N],IAC(A)[N],MASS(A)[N],CHARGE(A)[N],

```

```
                IAC(B) [N], MASS(B) [N], CHARGE(B) [N], ALJ [N], ACRF [N]
23  FORMAT (2I5,1X,A5,2F11.5,I4,4F11.5)
```

Multiple perturbed atom information block (for use in GROMOS++ only)

Blockname: MPERTATOM

```
        WRITE (unit,14) NJLA, NPTB
        WRITE (unit 21) (PTNAME[I],I=1,NPTB)
        DO 10 N=1, NJLA
10  WRITE (unit,22) NR[N], NAME[N], ((IAC(B) [I,N], CHARGE(B) [I,N]), I=1, NPTB), ALJ [N], ACRF [N]
21  FORMAT (16A5)
22  FORMAT (I5,1X,A5,16(I4,F11.5))
```

Perturbed NONBPL atom pair block

Blockname: PERTATOMPAIR

```
        WRITE (unit,14) NEB
        DO 10 N=1, NEB
10  WRITE (unit,14) IEB [N], JEB [N], IETA [N], IETB [N]
```

Perturbed bonds involving H-atoms block

Blockname: PERTBONDSTRETCHH

```
        WRITE (unit,14) NBONHG
        DO 10 N=1, NBONHG
10  WRITE (unit,14) IBHG [N], JBHG [N], ICBHA [N], ICBHB [N]
```

Perturbed bonds NOT involving H-atoms block

Blockname: PERTBONDSTRETCH

```
        WRITE (unit,14) NBONG
        DO 10 N=1, NBONG
10  WRITE (unit,14) IBG [N], JBG [N], ICBA [N], ICBB [N]
```

Perturbed bonds with a soft potential

Blockname: PERTBONDSOFT

```
        WRITE (unit,14) NBONSG
        DO 10 N=1, NBONSG
10  WRITE (unit,14) IBSG [N], JBSG [N], ICBSA [N], ICBSB [N], ALB [N]
```

Perturbed bond angles involving H-atoms block

Blockname: PERTBONDANGLEH

```
        WRITE (unit,14) NTHEHG
        DO 10 N=1, NTHEHG
10  WRITE (unit,14) ITHG [N], JTHG [N], KTHG [N], ICTHA [N], ICTHB [N]
```

Perturbed bond angles NOT involving H-atoms block

Blockname: PERTBONDANGLE

```
WRITE (unit,14) NTHEG
DO 10 N=1, NTHEG
10 WRITE (unit,14) ITG[N], JTG[N], KTG[N], ICTA[N], ICTB[N]
```

Perturbed bond angles with a soft potential

Blockname: PERTANGLESOFT

```
WRITE (unit,14) NTHESG
DO 10 N=1, NTHESG
10 WRITE (unit,14) ITSG[N], JTSG[N], KTSG[N], ICTSA[N], ICTSB[N], ALA[N]
```

Perturbed improper (harmonic) dihedrals involving H-atoms block

Blockname: PERTIMPROPERDIHH

```
WRITE (unit,14) NQHIG
DO 10 N=1, NQHIG
10 WRITE (unit,26) IQHG[N], JQHG[N], KQHG[N], LQHG[N], ICQHA[N],
              ICQHB[N]
26 FORMAT (6I5)
```

Perturbed improper (harmonic) dihedrals NOT involving H-atoms block

Blockname: PERTIMPROPERDIH

```
WRITE (unit,14) NQHIG
DO 10 N=1, NQHIG
10 WRITE (unit,26) IQG[N], JQG[N], KQG[N], LQG[N], ICQA[N], ICQB[N]
```

Perturbed improper (harmonic) dihedrals with a soft potential

Blockname: PERTIMPROPERDIHSOFT

```
WRITE (unit,14) NQHISG
DO 10 N=1, NQHISG
10 WRITE (unit,26) IQG[N], JQG[N], KQG[N], LQG[N], ICQA[N], ICQB[N], ALI[N]
```

Perturbed (trigonometric) dihedrals involving H-atoms block

Blockname: PERTPROPERDIHH

```
WRITE (unit,14) NPHIIG
DO 10 N=1, NPHIIG
10 WRITE (unit,26) IPHG[N], JPHG[N], KPHG[N], LPHG[N], ICPHA[N],
              ICPHB[N]
```

Perturbed (trigonometric) dihedrals NOT involving H-atoms block

Blockname: PERTPROPERDIH

```
WRITE (unit,14) NPHIG
DO 10 N=1, NPHIG
10 WRITE (unit,26) IPG[N], JPG[N], KPG[N], LPG[N], ICPA[N], ICPB[N]
```

Perturbed atomic polarisabilities block

Blockname: PERTPOLPARAM

```

WRITE (unit,14) NPOLG
DO 10 N=1, NPOLG
10 WRITE (unit,99) JLA[N], RESNR[N], ATNAME[N], ALPA[N], ENOTA[N],
      ALPB[N], ENOTB[N]

```

Examples of perturbation molecular topology files are named:

*.ptp

3.4. Atom-atom and distance-field distance restraints

When performing a simulation or energy minimization, a special interaction function term that restrains atom-atom distances can be added to the interaction function, see Sec. 2-9.3 and Sec. 2-9.12. Such a term may be used to make a molecule satisfy a given set of atom-atom distance upper or lower bounds, or to direct a molecule into the active site of a protein. A slight complication is that an atom involved in an atom-atom distance restraint pair may be a virtual or a pseudo atom (Sec. 2-9.4). In terms of a molecular topology or a molecular configuration such an atom is non-existing. As discussed in Sec. 2-9.4, its geometric position is defined in terms of the positions of its non-virtual neighbour atoms. For a virtual or pseudo atom the atom-atom distance restraint specification will contain the atom sequence numbers of the real atoms that define the virtual or pseudo atom position together with a geometry code denoting the specific geometric definition.

A set of atom-atom distance restraints in an atom-atom distance restraints file is characterized by the following quantities:

NDR	number of distance restraint atom pairs per “solute” molecule
I1, J1, K1, L1[1..NDR]	atom sequence numbers of the real atoms defining the geometric position of the first atom of a distance restraint pair (\leq NRP)
TYPE1[1..NDR]	geometric code defining the position of the first atom of a distance restraint pair [-2, -1, ..., 7]
I2, J2, K2, L2[1..NDR]	atom sequence numbers of the real atoms defining the geometric position of the second atom of a distance restraint pair (\leq NRP)
TYPE2[1..NDR]	geometric code defining the position of the second atom of a distance restraint pair [-2, -1, ..., 7]
R0[1..NDR]	in case of a full-harmonic distance restraint ($RAH = 0$), R0 is the minimum-energy distance; in case of an attractive or repulsive half-harmonic restraint ($RAH = \pm 1$), R0 is the upper or lower bound, respectively, beyond which the restraining forces become non-zero. When using distance restraints for NMR-NOE distance restraining, pseudo-atom corrections should already be included in R0 (see Sec. 2-9.4)
WO[1..NDR]	individual distance restraint weight factor, by which the distance restraint interaction term may be multiplied.
DIM	dimensionality-code for distance restraints. See below for allowed options. The value of DIM is determined from the value of RAH and is not stored separately.
RAH[1..NDR]	type of distance restraint; this parameter sets both the dimensions in which the restraint is applied as well as the shape of the functional form. if $RAH = DIM - 1$, a half-harmonic repulsive distance restraint is applied; if $RAH = DIM$, a full harmonic distance restraint is applied; if $RAH = DIM + 1$, a half-harmonic attractive distance restraint is applied.

DISH carbon-hydrogen distance, used for geometries TYPE = 1-6

DISC carbon-carbon distance, used for geometry TYPE = 6

In MD++ a distance restraint can also be modified in the course of a free energy perturbation. The perturbed distance restraints make use of the additional parameters

NDRP number of perturbed distance restraint atom pairs

M[1..NDRP] hidden restraint parameter: exponent of λ in state superposition prefactor

N[1..NDRP] hidden restraint parameter: exponent of $(1-\lambda)$ in state superposition prefactor

A_R0[1..NDRP] upper or lower bound beyond which the restraining forces become non-zero for state A

A_W0[1..NDRP] individual distance restraint weight factor by which the distance restraint interaction term may be multiplied for state A

B_R0[1..NDRP] as A_R0, but for state B

B_W0[1..NDRP] as A_W0, but for state B

As discussed in Sec. 2-9.4, the allowed geometries are the following ones. The notation is given in terms of hydrogen atoms.

TYPE = 0 real atom; its atom sequence number is given by IDR

TYPE = 1 virtual H-atom, aliphatic CH; it is bound to real atom I (carbon, atom sequence number IDR) and the three covalently-bound real neighbours of atom I are the real atoms J, K and L (atom sequence numbers JDR, KDR and LDR)

TYPE = 2 virtual H-atom, aromatic CH; it is bound to real atom I (carbon) and the two covalently-bound real neighbours of atom I are the real atoms J and K (LDR is not used)

TYPE = 3 pseudo H-atom, geometric mean of the two H-atoms of an aliphatic CH₂; it is (pseudo) bound to real atom I (carbon) and the two covalently-bound real neighbours of atom I are the real atoms J and K (LDR is not used)

TYPE = 4 virtual H-atom, one of the two H-atoms of an aliphatic CH₂; it is bound to real atom I (carbon) and the two covalently-bound real neighbours of atom I are the real atoms J and K (LDR is not used); the definition is the following: looking along covalent bond vector J-I from atom J to the central (carbon) atom I, the direction of the virtual bond I-H is obtained from the direction of the bond I-K by a counter-clockwise rotation over 120° around bond J-I; the other virtual H-atom can be obtained by interchanging the sequence numbers JDR and KDR

TYPE = 5 pseudo H-atom, geometric mean of the three H-atoms of a CH₃ group; it is (pseudo) bound to real atom I (carbon) and the one covalently-bound real neighbour of atom I is the real atom J (KDR and LDR are not used)

TYPE = 6 pseudo H-atom, geometric mean of the six H-atoms of two CH₃ groups that are both bound to a common third carbon atom; it is (pseudo) bound to this real third carbon atom I and the carbon atoms of the two CH₃ groups are the real atoms J and K (LDR is not used)

TYPE = 7 pseudo H-atom, geometric mean of the nine H-atoms of three CH₃ groups that are all three bound to a common fourth carbon atom I; it is (pseudo) bound to I and the fifth atom J is the real atom that is bound to I as well (KDR and LDR not used)

TYPE = -1 virtual atom, centre of geometry of the atoms I,J,K and L if their specifications are non-zero. (Example: the two (δ or ϵ) H-atoms I and J of an aromatic ring, or the two H-atoms I and J of a planar NH₂-group.)

TYPE = -2 virtual atom, centre of mass of the atoms I,J,K and L if their specifications are non-zero.

Atom-atom distance restraints may be applied in selected dimensions only. This is specified by the parameter RAH, from which the nearest integer code DIM is deduced. The following values of DIM are implemented in MD++:

DIM = 0 dimensions to apply distance restraint: X, Y, Z.

DIM = 10 dimensions to apply distance restraint: X, Y.

DIM = 20 dimensions to apply distance restraint: X, Z.

DIM = 30 dimensions to apply distance restraint: Y, Z.

DIM = 40 dimension to apply distance restraint: X.

DIM = 50 dimension to apply distance restraint: Y.

DIM = 60 dimension to apply distance restraint: Z.

Atom-atom distance restraint specification block

Blockname: DISTANCERESSPEC

```

WRITE (unit,11) DISH, DISC
DO 10 N=1, NDR
10 WRITE (unit,12) I1[N], J1[N], K1[N], L1[N], TYPE1[N],
                 I2[N], J2[N], K2[N], L2[N], TYPE2[N],
                 RO[N], WO[N], RAH[N]
11 FORMAT (2F10.5)
12 FORMAT (5I5,5X,5I5,3F10.5)

```

Perturbed atom-atom distance restraint specification block

Blockname:PERTDISRESSPEC

```

WRITE (unit,11) DISH, DISC
DO 10 N=1, NDRP
10 WRITE (unit,13) I1[N], J1[N], K1[N], L1[N], TYPE1[N],
                 I2[N], J2[N], K2[N], L2[N], TYPE2[N], M[N], N[N]
                 A_RO[N], A_WO[N], B_RO[N], B_WO[N], RAH[N]
13 FORMAT (5I5,5X,7I5,5F10.5)

```

For enveloping distribution sampling (EDS) it is convenient to define multiple perturbed distances. For this the MDISRESSPEC block may be used.

Multiple atom-atom distance restraint specification block

Blockname:MDISRESSPEC

```

WRITE (unit,11) DISH, DISC
DO 10 N=1, NDRP
10 WRITE (unit,13) I1[N], J1[N], K1[N], L1[N], TYPE1[N],
                 I2[N], J2[N], K2[N], L2[N], TYPE2[N],

```

```

                (RO [N,M],M=1,NEDS), (WO [N,M],M=1,NEDS), RAH [N]
13  FORMAT (5I5,5X,5I5,NEDS(F10.5),NEDS(F10.5),I5)

```

A distance-field distance restraint in an atom-atom distance restraints file is characterized by the following additional quantities:

PROTEINATOMS	last atom of the set of atoms to be defined as being part of the protein (\leq NRP)
K	force constant for the harmonic distance-field distance restraint
RO	minimum-energy distance on the distance-field grid
TYPE_I	geometric code defining the position of the first atom of the distance-field distance restraint, typically the protein [-2, -1, ..., 7]
NUM_I	number of atoms used to define the virtual atom I
ATOM_I [1..NUM_I]	atom sequence numbers of the atoms used to define the virtual atom I
TYPE_J	geometric code defining the position of the first atom of the distance-field distance restraint, typically the ligand [-2, -1, ..., 7]
NUM_J	number of atoms used to define the virtual atom I
ATOM_J [1..NUM_J]	atom sequence numbers of the atoms used to define the virtual atom I

In MD++ a distance-field distance restraint can also be modified in the course of a free energy perturbation. The perturbed distance-field distance restraint makes use of the additional parameters

K_A	force constant for the harmonic distance-field distance restraint in state A
K_B	force constant for the harmonic distance-field distance restraint in state B
A_RO	minimum-energy distance on the distance-field grid in state A
B_RO	minimum-energy distance on the distance-field grid in state B

Distance-field distance restraint specification block

Blockname: DFRESSPEC

```

        WRITE (unit,11) DISH, DISC
        WRITE (unit,12) PROTEINATOMS, K, RO
        WRITE (unit,13) TYPE_I, NUM_I, (ATOM_I [K],K=1,NUM_I)
        WRITE (unit,13) TYPE_J, NUM_J, (ATOM_J [K],K=1,NUM_J)
11  FORMAT (2F10.5)
12  FORMAT (I5,2F10.5)
13  FORMAT (16I5)

```

Perturbed distance-field distance restraint specification block

Blockname: PERTDFRESSPEC

```

        WRITE (unit,11) DISH, DISC
        WRITE (unit,12) PROTEINATOMS, A_RO, K_A, B_RO, K_B, M, N
        WRITE (unit,13) TYPE_I, NUM_I, (ATOM_I [K],K=1,NUM_I)
        WRITE (unit,13) TYPE_J, NUM_J, (ATOM_J [K],K=1,NUM_I)
11  FORMAT (2F10.5)
12  FORMAT (I5,4F10.5, 2I5)
13  FORMAT (16I5)

```

Examples of atom-atom distance restrained files are named:

*.dsr

Program *prep_noe* can produce an atom-atom distance restrained file for virtual and pseudo atoms from a list of proton-proton distances and a library file. See also Sec. 7.8.

3.5. Dihedral-angle restraints or constraints

When performing a simulation or energy minimization, a special interaction function term that restrains dihedral angles can be added to the interaction function. Dihedral angles can also be constraint, see Chap. 2. This approach may be used to make a molecule satisfy a given set of dihedral angle values.

A set of dihedral-angle restraints or constraints in a dihedral-angle restraints file is characterized by the following quantities:

NDLR	number of dihedral-angle restraints
IPLR, JPLR, KPLR, LPLR [1..NDLR]	atom sequence numbers of the atoms defining the restrained dihedral i-j-k-l (\leq NPM*NRP), j is always smaller than k
WDLR[1..NDLR]	individual dihedral restraint weight factor by which the harmonic dihedral restraining term may be multiplied.
PDLR[1..NDLR]	dihedral angle value (in degrees) at minimum energy of the harmonic dihedral restraining term; upon reading a dihedral angle restraints file, the dihedral angle is converted from degrees to radians and stored in PDLR
DELTA[1..NDLR]	dihedral angle value (in degrees) defining the periodic dihedral angle interval. The current dihedral angle value is shifted to the interval [PDLR[I] + DELTA[I] - 360.0 ,PDLR[I] + DELTA[I]] before force calculation

In MD++ a dihedral angle restraint can also be modified in the course of a free energy perturbation. The perturbed dihedral angle restraints make use of the additional parameters

NDLRP	number of perturbed dihedral-angle restraints
MLR[1..NDLRP]	hidden restraint parameter: exponent of λ in state superposition prefactor
NLR[1..NDLRP]	hidden restraint parameter: exponent of $(1-\lambda)$ in state superposition prefactor
APDLR[1..NDLRP]	dihedral angle value (in degrees) at minimum energy of the harmonic dihedral restraining term in state A
AWDLR[1..NDLRP]	Individual dihedral restraint weight factor by which the harmonic dihedral restraining term may be multiplied in state A
BPDLR[1..NDLRP]	as APDLR, but for state B
BWDLR[1..NDLRP]	as AWDLR, but for state B

Dihedral angle restraint specification block

Blockname: DIHEDRALRESSPEC

```
DO 10 N=1, NDLR
10 WRITE (unit,11) IPLR[N], JPLR[N], KPLR[N], LPLR[N], WDLR[N],
      PDLR[N], DELTA[N]
11 FORMAT (4I5,2F15.7)
```

Perturbed dihedral angle restraint specification block

Blockname: PERTDIHRESSPEC

```
DO 10 N=1, NDLRP
10 WRITE (unit,11) IPLR[N], JPLR[N], KPLR[N], LPLR[N], MLR[N], NLR[N], DELTA[N],
      APDLR[N], AWDLR[N], BPDLR[N], BWDLR[N]
11 FORMAT (4I5,5F15.7)
```

Examples of dihedral angle restraint files are named:

*.dhr

3.6. 3J -coupling constant restraints

When performing a simulation or energy minimization, a special interaction function term that restrains NMR 3J -coupling constants can be added to the interaction function, see Sec. 2-9.7. Such a term may be used to make a molecule satisfy a given set of 3J -values.

A set of 3J -coupling constant restraints in a 3J -coupling constant restraints file is characterized by the following quantities:

NDJV number of 3J -coupling constant restraints.

IPJV, JPJV, KPJV, LPJV [1..NDJV] atom sequence numbers of the real atoms present in the simulation that define the dihedral angle related to the restrained 3J -value (\leq NRP).

WJVR[1..NDJV] individual 3J -value restraint weight factor by which the restraining term for each 3J -value may be multiplied.

PJR0[1..NDJV] experimental or reference 3J -value, J_0 (≥ 0). In the case of a full-harmonic 3J -value restraint (NHJV = 0), PJR0 is the minimum-energy 3J -value; in the case of an attractive or repulsive half-harmonic 3J -value restraint (NHJV = ± 1), it is the upper or lower bound, respectively, beyond which the restraining force becomes non-zero.

PSJR[1..NDJV] phase shift or difference $\delta = \theta - \phi$ between the dihedral angle θ formed by the possibly non-existent atoms defining the experimental 3J -coupling and the dihedral angle $\phi(i-j-k-l)$ formed by the real atoms present in the simulation (in degrees); upon reading a 3J -coupling constant restraints file, the phase shift is converted from degrees to radians and stored in PSJR.

AJV, BJV, CJV[1..NDJV] Karplus parameters a , b and c for the 3J -coupling constant expressed as function of the dihedral angle θ

NHJV[1..NDJV] the type of 3J -value restraint; if H = -1, a half-harmonic repulsive 3J -value restraint is applied; if H = 0, a full harmonic 3J -value restraint is applied; if H = 1, a half-harmonic attractive 3J -value is applied. Note that the half-harmonic forms of the potential are only implemented in analogy to distance restraining and make little sense for restraining 3J -values, which depend on a periodic structural parameter.

3J -coupling constant restraint specification block

Blockname: JVALRESSPEC

```

      DO 10 N=1, NDJV
10  WRITE (unit,11) IPJV[N], JPJV[N], KPJV[N], LPJV[N], WJVR[N],
      PJRO[N], PSJR [N], AJV[N], BJV[N], CJV[N], NHJV[N]
11  FORMAT (4I5,7F10.5)

```

Examples of 3J -coupling constant restraint files are named:

*.jvr

3.7. S^2 -order parameter restraining

When performing a simulation or energy minimization, a special interaction function term that restrains NMR S^2 -order parameters can be added to the interaction function, see Sec. 2-9.8. Such a term may be used to make a molecule satisfy a given set of S^2 -values.

A set of S^2 -order parameter restraints in a S^2 -order parameter restraints file is characterized by the following quantities:

NOPR	number of S^2 -order parameter restraints.
I1, J1, K1, L1 [1..NOPR]	atom sequence numbers of the real atoms defining the geometric position of the first atom of the order parameter restraint pair (\leq NRP).
TYPE1 [1..NOPR]	geometric code defining the position of the first atom of a order parameter restraint pair [-2, -1, ..., 7]
I2, J2, K2, L2 [1..NOPR]	atom sequence numbers of the real atoms defining the geometric position of the second atom of the order parameter restraint pair (\leq NRP).
TYPE2 [1..NOPR]	geometric code defining the position of the second atom of a order parameter restraint pair [-2, -1, ..., 7]
RN [1..NOPR]	effective distance used to make the order parameter dimensionless.
S0 [1..NOPR]	experimental or reference S^2 -value
DSO [1..NOPR]	size of flat bottom region in one direction
WOPR [1..NOPR]	individual order parameter restraint weight factor, by which the order parameter restraint term may be multiplied
DISH	carbon-hydrogen distance, used for geometries ICOPR = 1-6
DISC	carbon-carbon distance, used for geometry ICOPR = 6

S^2 -order parameter restraint specification block

Blockname: ORDERPARAMRESSPEC

```

      WRITE (unit,11) DISH, DISC
      DO 10 N=1, NOPR
10  WRITE (unit,12) IOPR1[N], JOPR1[N], KOPR1[N], LOPR1[N], ICOPR1[N],
      IOPR2[N], JOPR2[N], KOPR2[N], LOPR2[N], ICOPR2[N],
      RN[N], SO[N], DSO[N], WOPR[N]
11  FORMAT (2F10.5)
12  FORMAT (5I5,5X,5I5,3F10.5)

```

Examples of S^2 -order parameter restraint files are named:

*.opr

Symmetry restraining block If the symmetry within a unit cell is to be restrained additional parameters, a description of the asymmetric unit and the atoms to be restrained have to be given.

NTSYM A switch to determine the method of symmetry restraining. 0: no symmetry restraining, 1: harmonic symmetry restraining, 2: symmetry constraining

CSYM The force constant for the symmetry restraints

SYMSPGR The space group in Hermann-Mauguin format

SYMNUMSYM The number of asymmetric units combined to form the unit cell

ASUDEF [1 .. SYMNUMSYM]
 The index of the first atom of each asymmetric unit

NSYMATOMS The number of atoms to be symmetry restrained

SYMATOMS [1 .. NSYMATOMS]
 The index of the atom to be symmetry restrained. Only the atoms in the first asymmetric unit are to be specified. The indices of the atoms of the other asymmetric units are determined automatically.

Blockname: XRAYSYMRESSPEC

```
8  WRITE (unit,9) NTSYM, CSYM
9  FORMAT (I5,F10.5)
10 WRITE (unit,11) SYMSPGR
11 FORMAT (A20)
12 WRITE (unit,13) ASUDEF[N], N = 1, SYMNUMSYM
13 FORMAT (100I5)
   DO 10 N=1, NSYMATOMS
14 WRITE (unit,15) SYMATOMS[N]
15 FORMAT (17X, I5)
```

B-factor optimisation blocks

For the optimisation of atomic B-factors additional parameters can be specified. In addition, groups of equal atoms can be specified.

BFOPTS Optimise B-factors every BFOPTSth step

BFOPTTI Terminate after BFOPTTI minimisation iterations

BFOPTTG Terminate if the gradient is smaller than BFOPTTG

BFOPTMN The minimal B-factor

BFOPTMX The maximal B-factor

BFOPTNG The number of B-factor groups

BFOPTGS [1 .. BFOPTNG]
 The size of a B-factor group

BFOPTGM [1 .. BFOPTNG] [1 .. BFOPTGS]
 The index of the atom being member in this group

Blockname: XRAYBFACTOROPTIMISATION

```
1  WRITE (unit, 2) BFOPTS, BFOPTTI, BFOPTTG, BFOPTMN, BFOPTMX
2  FORMAT (2I5, 3F10.5)
```

```

3  WRITE (unit, 4) BFOPTNG
4  FORMAT (I5)
   DO 5  N=1, BFOPTNG
5  WRITE (unit, 6) BFOPTGS[N], BFOPTGM[N][K], K=1, BFOPTGS[N]
6  FORMAT (1001I5)

```

Structure factor computation As the computation of structure factors is computationally demanding, it can be either carried out every selected step or whenever an atom has moved by some distance.

SFCTOL The distance an atom is allowed to move before the structure factors are recalculated

SFCST recalculate the structure factors every SFCST steps

Blockname: XRAYSFALC

```

1  WRITE (unit, 2) SFCTOL, SFCST
2  FORMAT (2I5)

```

Replica exchange properties block This block is used to couple the crystallographic restraints with lambda for Hamiltonian replica-exchange simulations.

NTXRRE Determines the coupling method to be used. 0: do not couple the X-ray restraints with lambda, 1: couple the force constant, 2: couple the resolution

CXREEMN The RLAM= 0 value of CXR or RESO

CXREEMX The RLAM= 1 value of CXR or RESO

Blockname: XRAYREPLICAEXCHANGE

```

1  WRITE (unit, 2) NTXRRE, CXREEMN, CXREEMX
2  FORMAT (I5, 2F10.5)

```

Examples of crystallographic restraint files are named:

*.xrs

3.8. Local-elevation coordinates

When performing a simulation, a special (changing) interaction function term that memorizes the values adopted during the simulation by a specified set of so-called local-elevation (LE) coordinates and increasingly penalizes readopting of these values, can be added to the interaction function, see Chap. 2. Such a local-elevation term may be used to drive a molecule out of a low energy conformation (LE searching).

The biasing potential energy term may also be frozen at some point and used to perform umbrella sampling (LEUS).

A set of local-elevation coordinates in a local-elevation coordinates file is characterized by the following quantities:

NPHILE number of local-elevation (LE) coordinates

NLEPID ID of LE potential-energy functions which will be associated to this dihedral. For n -dimensional potentials, the n dihedrals will have to be listed sequentially, using the same ID. Multiple sets of dihedrals may be associated with the same potential-energy function (thus multiple sets of dihedrals may build on the same potential-energy function).

VARTYPE Integer defining the variable type.

NVARAT Number of atoms needed to define the local elevation coordinate

AI[1..NVARAT,1..NPHILE]
atom sequence numbers of the atoms defining the local-elevation coordinate (\leq NRP)

The variable VARTYPE can take the following values

VARTYPE = 1 Dihedral angle, NVARAT = 4
VARTYPE = 2 Distance, NVARAT = 2
VARTYPE = 6 Distance-field distance, according to specification in distance restraint file, NVARAT = 0

local-elevation coordinate specification block

Blockname: LOCALELEVSPEC

```
DO 10 N=1, NPHILE
10 WRITE (unit,11) NLEPID[N], IPLE[N], JPLE[N], KPLE[N], LPLE[N]
11 FORMAT (6I5)
```

Note that also if NVARAT \neq 4, four values are read from the file.

Examples of local-elevation coordinate files are named:

*.led

3.9. Local elevation umbrella sampling database file

The LEUS database file contains a set of biasing potential energy functions that can be applied to specific subsets of collective variables so as to improve sampling (LEUS sampling). This file contains a title block and a LEUSBIAS and/or LEUSBIASBAS block (see page 4-46 and/or 4-48)

Samples of LEUS database files are named:

*.lud

3.10. Atomic friction coefficients

When performing stochastic dynamics simulations, atomic friction coefficients γ_i must be specified or calculated in some way, see Chap. 2. They may either be calculated in subroutine FRIC or specified in an atomic friction coefficient block or file.

A set of atomic friction coefficients is specified as follows:

NR number of atoms (=NATTOT)
GAM[1..NR] atomic friction coefficients

Atomic friction coefficient block

Blockname: FRICTIONSPEC

```
DO 10 J=1, NR
10 WRITE (unit,12) GAM[J]
12 FORMAT (24X,F15.9)
```

The first 24 positions are reserved for atom identification information.

Examples of atomic friction coefficient files are named:

*.frc

3.11. Position restraining or constraining atom specification list

When performing a simulation or energy minimization, a special interaction function term that restrains atomic positions can be added to the interaction function, see Chap. 2. Such a term may be used to keep parts of a molecular system relatively rigid. Another possibility is to keep atom positions fixed (constrained positions).

A set of atoms that are to be positionally restrained or constrained is specified as follows:

NRC number of position restrained or constrained atoms (\leq NRP +NSM*NRAM)
JRC[1..NRC] atom sequence numbers of the position-restrained or constrained atoms(\leq NRP+NSM*NRAM
for restraining)

Position restraining or constrained atom specification block

Blockname: POSRESSPEC

```
DO 10 N=1, NRC
10 WRITE (unit,12) JRC[N]
12 FORMAT (17X,I7)
```

The first 17 positions are reserved for residue number, residue name and atom name.

Examples of position restraining (or constraining) atom specification files are named:

*.por

3.12. B-factor restraining

Atomic mobilities or positional fluctuations can be stored in the form of isotropic B-factors

$$B_i = (8\pi^2/3) \langle (r_i - \langle r_i \rangle)^2 \rangle$$

The quantities characterising fluctuations or coordinate distributions are the following:

NR number of atoms (= NATTOT)
X[1..3, 1..NR] atomic Cartesian coordinates
BFAC[1..NR] atomic isotropic B-factors
DXY[1..3, 1..3, 1..NR] atomic positional fluctuation tensors (6 components)

The B-factor of fluctuation blocks are the following:

Isotropic B-factor block

Blockname: BFACTOR

Formatted form

```
DO 10 J=1, NR
WRITE (unit,25) MRES[J], AANMA[J], PANM[J], J, BFAC[J]
25 FORMAT (I5,2(1X,A5),I7,6F9.5)
```

Anisotropic B-factor block

Blockname: BFACTORANISO

Formatted form

```
DO 10 J=1, NR
WRITE (unit,25) MRES[J], AANMA[J], PANM[J], J,
((8*pye**2*DXY[K1,K2,J], K1=1,K2), K2=1,3)
```

3.13. Backwards compatibility with GROMOS96

The file formats used by MD++ for topological information differ in a number of respects from those of GROMOS96:

- Molecular Topology (section Sec. 3.2)
 - the TOPVERSION block is expected to contain the version number 2.0
 - the GROMOS 96 TOPPHYSICON block must be replaced by a GROMOS05 PHYSICALCONSTANTS block (including the value of Boltzmann’s constant in addition)
 - the GROMOS96 BONDTYPE block may be replaced by a BONDSTRETCHTYPE block (including force constants for a harmonic interaction form in addition to the quartic one). The two types of blocks cannot be present simultaneously. A HARMBONDTYPE block, containing only force constants for the harmonic interaction form can be used with the BONDTYPE block.
 - the GROMOS96 BONDANGLETYPE block may be replaced by a BONDANGLEBENDTYPE block (including force constants for an interaction form harmonic in the bond angle, in addition to those for the form harmonic in the angle cosine. The two types of blocks cannot be present simultaneously. A HARMBONDANGLETYPE block, containing only force constants for the harmonic interaction form can be used with the BONDANGLETYPE block.
 - the GROMOS96 DIHEDRALTYPE block may be replaced by a TORSDIHEDRALTYPE block (including arbitrary phase-shift angles in degrees within the range 0 to 360°, rather than phase-shift cosines restricted to the values -1 or +1). The two types of blocks cannot be present simultaneously.
 - a SOLUTEMOLECULES block must be included that defines all separate (covalently-linked) solute molecules (per solute unit)
 - a TEMPERATUREGROUP block must be included that defines groups of solute atoms (per solute unit) used to separate translational from internal-plus-rotational velocity components for the application of thermostatting and/or roto-translational constraints
 - a PRESSUREGROUPS block must be included that defines groups of solute atoms (per solute unit) used for the definition of a group-based virial
- Perturbation Molecular Topology (section Sec. 3.3)

- the GROMOS96 PERTATOM block must be replaced by a PERTATOMPARAM block containing in addition parameters for state A (for consistency checking).
- the GROMOS96 blocks
PERTBONDH, PERTBOND, PERTANGLEH, PERTANGLE, PERTIMPDIHEDRALH, PERTIMPDIHEDRAL, PERTDIHEDRALH and PERTDIHEDRAL
must be replaced by
PERTBONDSTRETCHH, PERTBONDSTRETCH, PERTBONDANGLEH, PERTBONDANGLE, PERTIMPROPERDIHH, PERTIMPROPERDIH, PERTPROPERDIHH and PERTPROPERDIH
respectively, containing the corresponding information in the form of type codes rather than interaction parameters.

The other types of topological information are essentially unaltered with respect to GROMOS96.

Configurational information

4.1. Introduction

Here, it is described in which form configurational quantities, atomic coordinates and atomic coordinate dependent (e.g. energies, internal coordinates) or related (e.g. velocities, forces, atomic stochastic integrals) quantities are stored on file.

Generally, the blocks on a file are written in formatted form. A number of quantities can be written in two ways (different blocktypes):

1. *standard formatted form*
2. *reduced-information formatted form*

The former form is used when writing a file containing data concerning a *single configuration* or *time frame* of a molecular system. When performing MD, a whole time series of configurations or a *trajectory* of a molecular system is produced. Since trajectories require much more storage capacity, they are stored using the reduced-information formatted form, in which redundant information has been omitted. The extra information in the *standard formatted form block* is helpful for interpretation, but is redundant, since it is also present in the topologicaql files for the molecular system.

There is no structural difference between single configuration and trajectory files. On the latter, a specific block will occur more than once.

We note that the sequence of data (atoms, distance restraints, etc.) within a block on a configurational file must match the sequence of the same data in the corresponding topological file (molecular topology, distance restraint specification, etc.) for the molecular system, see Chap. 1.

Quantities are generally stored with one atom or quantity per line, thereby repeating the atom or quantity identification information in different blocks in order to allow for easy identification of atoms or quantities. When a program has read a topological file, it takes the topological information, such as MRES[J], AANMA[J], PANM[J], J from there and it ignores these quantities on the configurational file.

A molecular configuration may not only be characterized by atom coordinates, and atom sequence numbers, but also by other quantities such as crystallographic B-factors.

In MD or SD simulations quantities such as velocities, stochastic integrals and random number generator seed need to be stored with a final configuration in order to be able to later continue the simulation. When the volume of the system varies with time, i.e. when pressure coupling is applied, the dimensions of the periodic box need to be stored, as well as the (possibly changing) reference positions if position restraining is also applied.

For simulations under periodic boundary conditions (where particles diffuse in an infinite periodic system) lattice-shift vectors are stored along with the atomic coordinates translated into the reference box (these are used for the calculation of a group-based pressure). Application of Nosé-Hoover chains for thermostatting and barostatting require the storage of the thermostat variables. Note that MD++ only supports Nosé-Hoover chains for thermostatting. Application of roto-translational constraints require the storage of parameters defining the position and orientation of specific atom groups.

Free energy calculations using the slow-growth or continuous λ -change technique require the storage of the actual λ -value and the cumulative derivatives of the Hamiltonian terms with respect to λ . When using time-averaging in atom-atom distance restraining or in 3J -value restraining, the current averaged distances or 3J -values need to be stored. When applying local-elevation search, the information with respect to parts of the energy hypersurface that have been elevated so far need to be stored in order to use it when continuing the simulation.

In the next sections the various quantities and their mode of storage is described.

4.2. Atomic coordinates

The atomic Cartesian coordinates of a molecular configuration can be stored as follows:

NR	number of atoms
NDIM	dimensionality of the Cartesian space
X[1..3, 1..NR]	atomic Cartesian coordinates (MD: trajectory at time t, final configuration at time t+ Δ t)
MRES[1..NR]	residue number
AANMA[1..NR]	residue name
PANM[1..NR]	atom name
J	atom sequence number
XC[1..3, 1..NR]	atomic Cartesian reference positions for position restraining (MD: final configuration at time t+ Δ t)
NLSHFT[1..3,1..NR]	lattice-shift vectors defining the position of particles in the infinite periodic system relative to their position in the reference box (in units of the box edge vectors for rectangular and triclinic boxes; in units of half the box edge for a truncated-octahedron box)
RPOL[1..3, 1..NR]	displacement of charge-on-spring from atomic centres

The atomic coordinate blocks are the following:

3-dimensional coordinate block

Blockname: POSITION

```

DO 10 J=1, NR
10 WRITE (unit,12) MRES[J], AANMA[J], PANM[J], J, (X[M,J], M=1,3)
12 FORMAT (I5,2(1X,A5),I7,3F15.9)

```

Blockname: POSITIONRED

Reduced-information form

```

DO 10 J=1, NR
10 WRITE (unit,13) (X[M,J], M=1,3)
13 FORMAT (3F15.9)

```

Reference coordinate block

Blockname: REFPOSITION

```

      DO 10 J=1, NR
10  WRITE (unit,12) MRES[J], AANMA[J], PANM[J], J, (XC[M,J], M=1,3)
12  FORMAT (I5,2(1X,A5),I7,3F15.9)

```

Coordinates after SHAKE failure

Blockname: SHAKEFAILPOSITION

same format as POSITION (other quantities are written to the final coordinate file in the usual blocks)

Coordinates just before SHAKE failure (i.e. before the coordinate resetting was attempted)

Blockname: SHAKEFAILPREVPOSITION

same format as POSITION (other quantities are written to the final coordinate file in the usual blocks)

Blockname: LATTICESHIFTS

```

      DO 10 J=1, NR
10  WRITE (unit,30) (NLSHFT[M,J], M=1,3)
30  FORMAT (3I10)

```

Displacement of charge-on-spring from polarizable centres

Blockname: COSDISPLACEMENTS

```

      DO 10 J=1, NR
10  WRITE (unit,13) RPOL[M,J], M=1,3)

```

4.3. Atomic velocities

The atomic velocities (at time $t-\Delta t/2$) belonging to a molecular configuration (at time t) can be stored as follows:

V[1..NDIM, 1..NR] atomic velocities (trajectory at time $t-\Delta t/2$, final velocities at $t+\Delta t/2$)

3-dimensional velocity block

Blockname: VELOCITY

```

      DO 10 J=1, NR
10  WRITE (unit,12) MRES[J], AANMA[J], PANM[J], J, (V[M,J], M=1,3)
12  FORMAT (I5,2(1X,A5),I7,3F15.9)

```

Blockname: VELOCITYRED

Reduced-information form

```

      DO 10 J=1, NR
10  WRITE (unit,13) (V[M,J], M=1,3)
13  FORMAT (3F15.9)

```

4.4. Atomic forces

The atomic forces (at time t) belonging to a molecular configuration (at time t) can be stored as follows:

FF[1..NDIM, 1..NR] atomic free-flight forces (trajectory at time t); these are the raw atomic forces (physical forces and possibly including special forces) prior to the application of any constraint

FC[1..NDIM,1..NR] atomic constraint forces (trajectory at time t); these are the atomic forces induced by the application of all constraints on the system (typically SHAKE, but also possibly including special constraints)

The sum of the two contributions represents the actual total force used to propagate the system via the integration scheme.

3-dimensional free-flight force block

Blockname: FREEFORCE

```
      DO 10  J=1, NR
10    WRITE  (unit,12) MRES[J], AANMA[J], PANM[J], J, (FF[M,J], M=1,3)
12    FORMAT (I5,2(1X,A5),I7,3F15.9)
```

Blockname: FREEFORCERED

Reduced-information form

```
      DO 10  J=1, NR
10    WRITE  (unit,14) (FF[M,J], M=1,3)
14    FORMAT (3F 20.9)
```

3-dimensional constraint force block

Blockname: CONSFORCE

```
      DO 10  J=1, NR
10    WRITE  (unit,12) MRES[J], AANMA[J], PANM[J], J, (FC[M,J], M=1,3)
12    FORMAT (I5,2(1X,A5),I7,3F15.9)
```

Blockname: CONSFORCERED

Reduced-information form

```
      DO 10  J=1, NR
10    WRITE  (unit,13) (FC[M,J], M=1,3)
13    FORMAT (3F15.9)
```

4.5. Atomic stochastic integrals

When performing stochastic dynamics (SD) using the leap frog algorithm, the integrals of the stochastic forces over time are correlated between successive time steps. Therefore, the stochastic integrals SX[1..NDIM, 1..NR] over the time interval $(t+\Delta t/2, t+\Delta t)$ are stored to allow for continuation runs. For the same purpose the random number generator seed needs to be stored.

SX[1..NDIM, 1..NR] atomic stochastic integrals (interval $t+\Delta t/2$ to $t+\Delta t$)

IG random number generator seed (at time $t+\Delta t$)

3-dimensional stochastic integral block

Blockname: STOCHINT

```
DO 10 J=1, NR
10 WRITE (unit,15) MRES[J], AANMA[J], PANM[J], J, (SX[M,J], M=1,3)
   WRITE (unit,16) IG
15 FORMAT (I5,2(1X,A5),I7,3E15.7)
16 FORMAT (I10)
```

4.6. Periodic box

When using pressure coupling in a MD simulation, the parameters characterizing the size and shape of the periodic box that contains the molecular system as well as its orientation in space are a function of time, so these parameters need to be stored.

NTB type of boundary conditions; truncated-octahedron (-1), vacuum (0), rectangular (1), or triclinic (2)

BOX[1..3] lengths of the a-, b- and c-edges of the periodic box (trajectory at time t, final configuration at time $t+\Delta t$)

ALPHA angle between the b and c axes in degrees

BETA angle between the a and c axes in degrees

GAMMA angle between the a and b axes in degrees

PHI Euler yaw angle of the box (z-axis rotation) in degrees

THETA Euler pitch angle of the box (y-axis rotation) in degrees

PSI Euler roll angle of the box (x-axis rotation) in degrees

OX (Cartesian) x-coordinate of origin of triclinic box

OY (Cartesian) y-coordinate of origin of triclinic box

OZ (Cartesian) z-coordinate of origin of triclinic box

Boxsize block

Blockname: GENBOX

```
WRITE (unit,17) NTB
WRITE (unit,18) (BOX[M], M=1,3)
WRITE (unit,18) ALPHA, BETA, GAMMA
WRITE (unit,18) PHI, THETA, PSI
WRITE (unit,18) OX, OY, OZ
17 FORMAT (I5)
18 FORMAT (3F15.9)
```

4.7. Nose-Hoover chain thermostat variables

When using temperature coupling in an MD simulation based on the Nosé-Hoover chain thermostat, the values of the thermostat variables need to be stored.

NUM_NHC_TMP_BTH number of heat baths employing Nosé-Hoover chain thermostat coupling

NUM_VAR_NHC_TMP_BTH[1..NUM_NHC_TMP_BTH]
 number of variables used for each bath

VAL_VAR_NHC_TMP_BTH[I=1..NUM_NHC_TMP_BTH, 1..NUM_VAR_NHC_TMP_BTH[I]]
 values of the corresponding thermostat variables

Nosé-Hoover chain thermostat variables block

Blockname: NHCVARIABLES

```
DO 10 I=1, NUM_NHC_TMP_BTH
10 WRITE (unit,41) (VAL_VAR_NHC_TMP_BTH[I,J], J=1..NUM_VAR_NHC_TMP_BTH[I])
41 FORMAT (2I5,5F15.9)
```

4.8. Roto-translational constraints reference variables

When using translational or/and rotational constraining in an MD simulation, the values of the variables defining the reference position or/and orientation of all constrained atoms need to be stored. In MD++ roto-translational constraints are always applied on the first specified number of atoms. Therefore, only the translation and rotation matrices and reference positions of the first atoms are written.

RTCLAST The first RTCLAST atoms are roto-translationally constrained.

RTCTM[1..3,1..3] The translation matrix.

RTCRM[1..3,1..3] The rotation matrix.

RTCREF[1..3,1..NUMRTC]
 The reference atomic coordinates.

Rototranslational reference matrices and positions block

Blockname: ROTTRANSREFPOS

```
DO 10 J=1, 3
10 WRITE (unit,13) (RTCTM[M,J], M=1,3)
DO 11 J=1, 3
11 WRITE (unit,13) (RTCRM[M,J], M=1,3)
13 FORMAT (9F15.9)
DO 20 I=1, RTCLAST
20 WRITE (unit,40) (RTCREF[M,I], M=1,3)
40 FORMAT (3F15.9)
```

4.9. Perturbation data

When applying a perturbation to the Hamiltonian in a simulation in order to determine a free energy difference between two states of a molecular system using the so-called slow-growth or continuous λ -change technique, the value for the thermodynamic integration coupling parameter λ at time $t+\Delta t$ needs to be stored to allow for a continuation run. This value is stored in the PERTDATA block of the molecular configuration file.

RLAM thermodynamic integration coupling parameter λ (at time $t+\Delta t$)

Perturbation data block
Blockname: PERTDATA

```
WRITE (unit,19) RLAM
19 FORMAT (E15.7)
```

4.10. Atom-atom distance restraints

When applying time-averaging in atom-atom distance restraining in a simulation, the exponentially weighted average of r_{ij}^{-3} for the restrained atom-atom distance r_{ij} needs to be stored in order to allow for continuation runs.

NDR number of distance restraint atom pairs

RIIAVE[1..NDR] minus $\frac{1}{3}$ power of the exponentially weighted average of r_{ij}^{-3} for the restrained atom-atom distances r_{ij} (at time t)

Exponentially averaged distance restraint block
Blockname: DISRESEXPAVE

```
DO 10 N=1, NDR
10 WRITE (unit,19) RIIAVE[N]
19 FORMAT (E15.7)
```

4.11. 3J -coupling constant restraints

When applying time-averaging in 3J -coupling constant restraining in a simulation, the exponentially weighted time-average of the 3J -coupling value(s) need to be stored in order to allow for continuation runs.

NJR number of 3J -coupling constant restraints.

JVALAV[1..NJR] exponentially weighted average of each 3J -coupling value (at time t).

Exponentially averaged 3J -value restraint block
Blockname: JVALUERESXPAVE

```
DO 10 N=1, NJR
10 WRITE (unit,19) JVALAV[N]
19 FORMAT (E15.7)
```

In MD++, a local-elevation interaction term can be applied (along with time-averaging) to perform 3J -coupling constant restraining. In this case, the time-averaged, weighted heights at time t of the Gaussians describing the potential-energy penalty function at grid points [1..NJLEGR] need to be stored in order to allow for continuation runs.

NJLEGR number of grid points.

HJLEG[1..NJRJ, 1..NJLEGR]

time-averaged, weighted height of the local-elevation Gaussian at grid point M=[1..NJLEGR]
for the 3J -coupling-related dihedral angle θ (at time t).

Time-average weighted height of the local elevation penalty functions used in combination with 3J -coupling constant restraining

Blockname: JVALUERESEPS

```
      DO 10  N=1, NJR
      DO 11  M=1, NJLEGR
11    WRITE (unit,19) HJLEG[N] [M]
10    CONTINUE
19    FORMAT (E15.7)
```

In MD++, the force constant can be periodically scaled. In this case the scaling constant and time need to be stored in order to allow for continuation runs.

JVALSC[1..NJR] integer (0 or 1) which indicates whether the 3J -value's force constant is scaled.

JVALT[1..NJR] current time in scaling period.

Periodic scaling 3J -value restraint block

Blockname: JVALUEPERSCALE

```
      DO 10  N=1, NJR
10    WRITE (unit,19) JVALSC[N], JVALT[N]
19    FORMAT (I10, F15.9)
```

4.12. S^2 -order parameter restraints

When applying S^2 -order parameter restraining in a simulation, the exponentially weighted time averages of $Q_{\alpha\beta}$ and D need to be stored in order to allow for continuation runs.

NOPR number of order parameter restraints

QABAVE[1..NOPR,A,B]
exponentially weighted average of matrix elements $Q_{\alpha\beta}$ (9 elements) (at time t)

DAVE[1..NOPR] exponentially weighted average D (at time t)

Exponentially averaged S^2 -order parameter restraint block

Blockname: ORDERPARAMRESEXPAVE

```
      DO 9  N=1, NOPR
      WRITE (unit,19) QABAVE[N,1,1], QABAVE[N,1,2], QABAVE[N,1,3], QABAVE[N,2,1], QABAVE[N,2,2]
      WRITE (unit,19) QABAVE[N,2,3], QABAVE[N,3,1], QABAVE[N,3,2], QABAVE[N,3,3], DAVE[N]
9    CONTINUE
19   FORMAT (E15.7)
```

4.13. Crystallographic restraints

When applying time-averaging in structure-factor amplitude or electron density restraining in a simulation, the exponentially weighted average of the complex structure factor needs to be stored in order to allow for continuation runs.

NFXR number of structure-factor amplitudes
FXRAVE[1..NFXR] time-averaged structure-factor amplitude
PHXRAVE[1..NFXR] time-averaged structure-factor phase

Exponentially averaged structure factor block

Blockname: XRAYRESEXPAVE

```
      DO 10  N=1, NFXR
10  WRITE  (unit,19) FXRAVE[N], PHXRAVE[1..NFXR]
19  FORMAT (E15.7, E15.7)
```

When refining the atomic B-factors they are written to the configuration as well (XRAYBFOCCSPEC block).

4.14. Local-elevation data

When performing a simulation in which the local-elevation interaction term is switched on and the memory progressively builds up (LE searching), data on the local-elevation conformations that have been visited so far during the simulation needs to be stored in order to allow for continuation runs.

NUMB number of umbrella potential energy functions contained in the block

NLEPID[1..NUMB] potential energy function ID

NLEDIM[1..NUMB] dimensionality of the potential energy function

CLES[1..NUMB] force constant associated with the local functions

VARTYPE[1..NUMB,1..NLEDIM]
 type of variable (1: Dihedral angle, 2: Distance, 6: Distance-field distance)

NTLEFU[1..NUMB,1..NLEDIM]
 functional form of the local functions (0: truncated polynomials; 1: Gaussian)

WLES[1..NUMB,1..NLEDIM]
 width of the local functions in units of the grid spacing

RLES[1..NUMB,1..NLEDIM]
 cutoff applied to the range of action of the local functions in units of grid spacing

NGRID[1..NUMB,1..NLEDIM]
 number of grid points used along each dimension

GRIDMIN[1..NUMB,1..NLEDIM]
 minimum grid point used along each dimension

GRIDMAX[1..NUMB,1..NLEDIM]
 maximum grid point used along each dimension. If GRIDMAX[N]==GRIDMIN[N]
 a cyclic variable is assumed, applying NLEGRD[N] undistant grid points over the
 whole variable range, with first grid point at GRIDMIN[N]

NCONLE[1..NUMB] number of LE conformations visited and stored so far in memory (at time t)

ICONF[1..NUMB,1..NLEDIM, 1..NCONLE]
 integer coded representation of LE conformations. Position of grid point in di-
 mension N for potential K is given as POS = GRIDMIN[N] + (ICONF[N,K] -
 1)GRIDSPACING[N]

NVISLE[1..NUMB,1..NCONLE]
 number of times LE conformations have been visited so far (at time t)

local-elevation memory block

Blockname: LEUSBIAS

```

      WRITE (unit,20) NUMB
DO 9 I=1, NUMB
  WRITE (unit,21) NLEPID[I], NDIM[I], CLES[I]
  DO 10 N=1, NDIM[I]
10  WRITE (unit,22) VARTYPE[I,N],NTLEFU[I,N],WLES[I,N],RLES[I,N], &
      NGRID[I,N],GRIDMIN[I,N],GRIDMAX[I,N]
  WRITE (unit,20) NCONLE[I]
  DO 11 N=1, NCONLE[I]
11  WRITE (unit,23) (NVISLE[I,N], (ICONF[I,M,N], M=1, NDIM[I])
9  CONTINUE
20  FORMAT (1I8)
21  FORMAT (2I8, 1E18.10)
22  FORMAT (2I8, 2E18.10,I8,2E18.10)
23  FORMAT (8I8)

```

4.15. Ball and stick local-elevation data

When performing a simulation in which the Ball and Stick local-elevation interaction term is switched on and the memory progressively builds up (LE searching), data on the local-elevation conformations that have been visited so far during the simulation needs to be stored in order to allow for continuation runs.

NUMB number of umbrella potential energy functions contained in the block

NLEPID[1..NUMB] potential energy function ID

NLEDIM[1..NUMB] dimensionality of the potential energy function

ACTPOT[1..NUMB] the ID of the active potential energy function (not used when EDS combination is applied)

BETA[1..NUMB] the factor $s\beta$ used for EDS combination of bias potential energy functions

VARTYPE[1..NUMB,1..NLEDIM]
 type of variable (1: Dihedral angle, 2: Distance, 6: Distance-field distance)

DIMSCALE[1..NUMB,1..NLEDIM]
 by which factor should the respective variable be divided

SHIFTTYPE[1..NUMB,1..NLEDIM]
 0: Do not shift; 1: Shift to nearest image; 2: Apply shift vectors

REFSHIFT[1..NUMB,1..NLEDIM]
 Value of coordinate at last time step, used for updating shift vectors (only used for
 SHIFTTYPE=2)

NSPHERES[1..NUMB] Number of defined spherical potentials

SID[1..NUMB,1..NSPHERES]
 ID of defined sphere

NPRAD[1..NUMB,1..NSPHERES]
 Number of radial grid points of defined sphere

DR[1..NUMB,1..NSPHERES]
 Radial distance between grid points

IBUILD[1..NUMB,1..NSPHERES]
 0: Do not build; 1: Build proportional to EDS weight w ; 3: Build proportional to
 EDS weight and grid index to the power of SCALEVAL

SCALEVAL[1..NUMB,1..NSPHERES]
 Value for scaling the build-up as function of grid index

CLES[1..NUMB,1..NSPHERES]
 (Current) build-up force constant [kJ/mol]

REDFAC[1..NUMB,1..NSPHERES]
 Factor for reduction of build-up force constant

CRES[1..NUMB,1..NSPHERES]
 Force constant [kJ/mol] for restraint

VADD[1..NUMB,1..NSPHERES]
 Potential energy [kJ/mol] added to the energy of sphere (grid-point independent)

CENTRE[1..NUMB,1..NSPHERES,1..NDIM]
 Value of LE coordinates defining centre of the sphere

VSPHERE[1..NUMB,1..NSPHERES,1..NPRAD]
 Value of LE potential energy function at radial grid point [kJ/mol]

VISSPHERE[1..NUMB,1..NSPHERES,1..NPRAD]
 Number of visits at grid point

NLINES[1..NUMB] Number of defined lines

LID[1..NUMB,1..NLINES]
 ID of defined line

NPLINE[1..NUMB,1..NLINES]
 Number of grid points on line

IBUILD[1..NUMB,1..NLINES]
 0: Do not build 1: Build

CLES[1..NUMB,1..NLINES]
 LE force constant [kJ/mol]

REDFAC[1..NUMB,1..NLINES]
 Factor to reduce the LE force constant

VADD[1..NUMB,1..NLINES]
 Potential energy [kJ/mol] added to the energy of line (grid-point independent)

PSTART[1..NUMB,1..NLINES,1..NDIM]
 LE coordinates defining starting point of line

PEND[1..NUMB,1..NLINES,1..NDIM]
 LE coordinates defining end point of line

NDIS[1..NUMB,1..NLINES]
 Number of displacement vectors

DISVEC[1..NUMB,1..NLINES,1..NDIS,1..NDIM]
 Vector components of displacement vectors

VLINE[1..NUMB,1..NLINES,1..NPLINE]
 Potential energy [kJ/mol] at grid point

VISLINE[1..NUMB,1..NLINES,1..NPLINE]
 Number of visits at grid point

WIDTH[1..NUMB,1..NLINES,1..NPLINE]
 Width orthogonal to the line before start of restraining potential energy function

CRES[1..NUMB,1..NLINES,1..NPLINE]
 Restraining force constant

LAM[1..NUMB,1..NLINES,1..NPLINE]
 Value of λ for fixed λ simulations (currently unused)

DIS[1..NUMB,1..NLINES,1..NPLINE,1..NDIS]
 Displacement of current grid point along respective (orthonormalised) displacement vectors

local-elevation ball and stick memory block

Blockname: LEUSBIASBAS

```

      WRITE (unit,21) NUMB
DO 9 I=1, NUMB
  WRITE (unit,22) NLEPID[I], NDIM[I], ACTPOT[I], BETA[I]
  DO 11 N=1, NDIM[I]
11  WRITE (unit,23) VARTYPE[I,N], DIMSCALE[I,N], SHIFTTYPE[I,N], REFSHIFT[I,N]
  WRITE (unit,21) NSPHERES[I]
  DO 12 N=1, NSPHERES[I]
  WRITE (unit,24) ID[I,N], NPRAD[I,N], DR[I,N], IBUILD[I,N], RBUILD[I,N], CLES[I,N], &
    REDFAC[I,N], CRES[I,N], VADD[I,N], (CENTRE[I,N,M], M=1, NDIM[I])
  WRITE (unit,25) (VSPHERE[I,M], M=1, NPRAD[I,N])
  WRITE (unit,25) (VISSPHERE[I,M], M=1, NPRAD[I,N])
12  CONTINUE

```

```

WRITE (unit,21) NLines[I]
DO 13 N=1, NLines[I]
WRITE (unit,26) ID[I,N], NPLINE[I,N], IBUILD[I,N], CLES[I,N], REDFAC[I,N], VADD[I,N]
WRITE (unit,25) (PSTART[I,N,M], M=1,NDIM[I])
WRITE (unit,25) (PEND[I,N,M], M=1,NDIM[I])
WRITE (unit,21) NDIS[I,N]
DO 14 K=1, NDIS[I,N]
WRITE (unit,25) (DISVEC[I,N,K,M], M=1,NDIM[I])
14 CONTINUE
DO 15 M=1, NPLINE[I,N]
WRITE (unit,25) VLINE[I,N,M], VISLINE[I,N,M], WIDTH[I,N,M], CRES[I,N,M], LAM[I,N,M], &
(DIS[I,N,M,K], K=1,NDIS[I,N])
15 CONTINUE
13 CONTINUE
WRITE (unit,27) NSTATES[I], NCHECK[I], NCHECKCUR[I]
DO 16 N=1, NSTATES[I]
WRITE (unit,28/29) TYPE[I,N], NVISITS[I,N], (PARAMS[I,N,M], NPAR)
16 CONTINUE
9 CONTINUE
21 FORMAT (1I8)
22 FORMAT (3I8,1E18.10)
23 FORMAT (1I8,1E18.10,I8,5E18.10)
24 FORMAT (2I8,1E18.10,I8,50E18.10)
25 FORMAT (50E18.10)
26 FORMAT (3I8,5E18.10)
27 FORMAT (3I8)
28 FORMAT (3I8,50E18.10)
29 FORMAT (3I8,E18.10,I8)

```

4.16. Time or step number data

Generally, trajectory files are written such that the time frames are equidistant in time, i.e. correspond to a multiple of a specified number of simulation steps. So, time or step number of a block are known in that case. When selecting configurations to be stored using a criterion such as low potential energy, the configurations will not be equidistant in time. In that case time or step number information should be added to a configuration.

T time in the simulation ($t = t_n$)

NSTEP step number since the beginning of the current runs (n)

Time and step number data block

Blockname: TIMESTEP

```

WRITE (unit,21) NSTEP, T
21 FORMAT (I15,F20.9)

```

4.17. Energies, pressure, volume and free-energy data

Program MD++ writes out the following arrays:

ENER[1] total energy of the molecular system (at time t)

ENER[2] total kinetic energy of the molecular system (at time t)

ENER[3] total potential energy of the molecular system (at time t)

ENER[4]	total energy of covalent terms (solutes, at time t)
ENER[5]	total energy of bond-stretching terms (solutes, at time t)
ENER[6]	total energy of bond-angle bending terms (solutes, at time t)
ENER[7]	total energy of improper (harmonic) dihedral angle terms (solutes, at time t)
ENER[8]	total energy of (trigonometric) dihedral angle terms (solutes, at time t)
ENER[9]	total energy of crossdihedral angle terms (solutes, at time t)
ENER[10]	total energy of nonbonded terms (solutes, at time t)
ENER[11]	total energy of van der Waals interaction terms (at time t)
ENER[12]	total energy of electrostatic interaction terms (at time t)
ENER[13]	total energy of lattice sum terms (at time t)
ENER[14]	total energy of lattice sum pair term (at time t)
ENER[15]	total energy of lattice sum real space term (at time t)
ENER[16]	total energy of lattice sum reciprocal space term (at time t)
ENER[17]	total energy of lattice sum A term (at time t)
ENER[18]	total energy of lattice sum self term (at time t)
ENER[19]	total energy of lattice sum surface term (at time t)
ENER[20]	total energy of polarisation self term (at time t)
ENER[21]	total energy of special terms (at time t)
ENER[22]	total energy of SASA term (at time t)
ENER[23]	total energy of SASA volume term (at time t)
ENER[24]	total energy due to constraints in the molecular system (at time t)
ENER[25]	total energy of atom-atom distance restraint terms (at time t)
ENER[26]	total energy of distance-field restraining terms (at time t)
ENER[27]	total energy of dihedral angle restraining terms (at time t)
ENER[28]	total energy of atom position restraining terms (at time t)
ENER[29]	total energy of 3J -value restraining terms (at time t)
ENER[30]	total energy of X-ray restraining terms (at time t)
ENER[31]	total energy of local-elevation terms (at time t)
ENER[32]	total energy of S^2 order parameter restraining terms (at time t)
ENER[33]	total energy of symmetry restraining terms (at time t)
ENER[34]	total energy of non-accelerated EDS reference state in accelerated EDS (at time t)
ENER[35]	total energy of EDS reference state (at time t)
ENER[36]	accelerated EDS parameter E_{max} (at time t)

ENER[37] accelerated EDS parameter E_{min} (at time t)

ENER[38] average energy of the end-state with the lowest energy in accelerated EDS parameter search (at time t)

ENER[39] standard deviation of the energy of the end-state with the lowest energy in accelerated EDS parameter search (at time t)

ENER[40] total entropy term (at time t)

ENER[41] total energy QM

ENER[42] total energy of ball and stick local elevation (at time t)

ENER[43] total energy of RDC restraining terms (at time t)

NBATHS number of temperature baths defined for constant temperature simulations

KINENER[1,1..NBATHS] total kinetic energy of individual temperature baths (at time t)

KINENER[2,1..NBATHS] total translational kinetic energy of the centres of mass of the molecules coupled to the individual baths (at time t)

KINENER[3,1..NBATHS] total internal-rotational kinetic energy of the individual temperature baths (at time t)

NEGR number of groups G_i of atoms for which the energy terms are separately stored

BONDED[1,1..NEGR] total energy of bond-stretching terms of which the first atom belongs to the energy group G

BONDED[2,1..NEGR] total energy of bond-angle bending terms of which the first atom belongs to the energy group G

BONDED[3,1..NEGR] total energy of improper (harmonic) dihedral angle terms of which the first atom belongs to the energy group G

BONDED[4,1..NEGR] total energy of dihedral (trigonometric) angle terms of which the first atom belongs to the energy group G

BONDED[5,1..NEGR] total energy of crossdihedral angle terms of which the first atom belongs to the energy group G

NONBONDED[1,1..NEGR*(NEGR+1)/2] total Van der Waals interaction energies between atoms belonging to the different groups G_i (at time t); the order of the group-group energies is 1-1, 1-2, 2-2, ..., 1-NEGR, 2-NEGR, ..., NEGR-NEGR

NONBONDED[2,1..NEGR*(NEGR+1)/2] idem, but for the total electrostatic interaction

NONBONDED[3,1..NEGR*(NEGR+1)/2] idem, but for the total lattice sum real space energy

NONBONDED[4,1..NEGR*(NEGR+1)/2] idem, but for the total lattice sum reciprocal space energy

SPECIAL[1,1..NEGR] total constraint energy per energy group G

SPECIAL[2,1..NEGR] total energy of position restraining terms per energy group G

SPECIAL [3,1..NEGR] total energy of distance restraints per energy group G

SPECIAL [4,1..NEGR] total energy of distance-field restraints per energy group G

SPECIAL [5,1..NEGR] total energy of dihedral restraints per energy group G

SPECIAL [6,1..NEGR] total energy of SASA term per energy group G

SPECIAL [7,1..NEGR] total energy of SASA volume term per energy group G

SPECIAL [8,1..NEGR] total energy of 3J -value restraints per energy group G ($= 0$; the 3J -value restraints are not split up per energy group)

SPECIAL [9,1..NEGR] total energy of RDC restraints per energy group G

SPECIAL [10,1..NEGR]
total energy of local-elevation terms per energy group G ($= 0$; the local-elevation terms are not split up per energy group)

SPECIAL [11,1..NEGR]
total energy of X-ray restraining terms per energy group G ($= 0$; the X-ray restraining terms are not split up per energy group)

NEDS number of EDS states

EDSENER [1,1..NEDS] total potential energy per EDS state

EDSENER [2,1..NEDS] total nonbonded energy per EDS state

EDSENER [3,1..NEDS] total special energy functions per EDS state

EDSENER [4,1..NEDS] energy offset parameter per EDS state in accelerated EDS

MASS total mass of all particles in the system

TEMPERATURE [1,1..NBATHS]
temperature of the part of the system that is coupled to every temperature bath

TEMPERATURE [2,1..NBATHS]
temperature associated with the centre of mass translational degrees of freedom of the submolecules that are coupled to every temperature bath

TEMPERATURE [3,1..NBATHS]
temperature associated with the internal and rotational degrees of freedom of the part of the system that that is coupled to every temperature bath

TEMPERATURE [4,1..NBATHS]
scaling factor for scaling the corresponding degrees of freedom for every temperature bath (used at time $t + \Delta t/2$)

VOLUME total volume of the computational box.

BOX [1..3,1..3] triclinic unit vectors K, L, M

PRESSURE [1] total pressure of the system

PRESSURE [2] total virial of the system

PRESSURE [3] total translational kinetic energy matrix for centre of mass for all submolecules

PRES [1..3,1..3] pressure tensor

VIRIAL [1..3,1..3] virial matrix

KINETIC[1..3,1..3] translational kinetic energy matrix for centre of mass for all submolecules

RLAM perturbation parameter λ (at time t)

FREEENER[1..38] derivatives of the various terms of the Hamiltonian with respect to λ ; the energy terms are the same as in ENER[1..38]

FREEKINENER[1..3,1..NBATHS] derivatives of the kinetic energy terms with respect to λ ; the energy terms are the same as in KINENER[1..3,1..NBATHS]

FREEBONDED[1..5,1..NEGR] derivatives of the bonded energy terms with respect to λ ; the energy terms are the same as in BONDED[1..5,1..NBATHS]

FREENONBONDED[1..4,1..NEGR*(NEGR+1)2] derivatives of the various terms of the Hamiltonian with respect to λ ; the energy terms are the same as in NONBONDED[1..4,1..NEGR*(NEGR+1)2]

FREESPECIAL[1..11,1..NEGR] derivatives of the special interaction energy terms with respect to λ ; the energy terms are the same as in SPECIAL[1..11, 1..NEGR]

FREEEDSENER[1..3,1..NEDS] derivatives of the EDS energies with respect to λ ; the energy terms are the same as in EDSENER[1..3,1..NEDS]

Energy block

Blockname: ENERGY03

```

DO 10 N=1,43
10 WRITE (unit,23) ENER[N]
   WRITE (unit,22) NBATHS
DO 11 N=1,NBATHS
11 WRITE (unit,23) KINENER[1,N],KINENER[2,N],KINENER[3,N]
   WRITE (unit,22) NEGR
DO 12 N=1, NEGR
12 WRITE (unit,23) BONDED[1,N],BONDED[2,N],BONDED[3,N],
   BONDED[4,N],BONDED[5,N]
DO 13 N=1, NEGR*(NEGR+1)/2
13 WRITE (unit,23) NONBONDED[1,N],NONBONDED[2,N],
   NONBONDED[3,N],NONBONDED[4,N]
DO 14 N=1, NEGR
14 WRITE (unit,23) SPECIAL[1,N],SPECIAL[2,N],SPECIAL[3,N],SPECIAL[4,N],
   SPECIAL[5,N],SPECIAL[6,N],SPECIAL[7,N],SPECIAL[8,N],
   SPECIAL[9,N],SPECIAL[10,N],SPECIAL[11,N]
   WRITE (unit,22) NEDS
DO 15 N=1,NEDS
15 WRITE (unit,23) EDSENER[1,N],EDSENER[2,N],EDSENER[3,N],EDSENER[4,N]
22 FORMAT (I5)
23 FORMAT (11E17.9)

```

Volume, pressure block

Blockname: VOLUMEPRESSURE03

```

WRITE (unit,23) MASS

```

```

WRITE (unit,22) NBATHS
DO 10 N=1, NBATHS
10 WRITE (unit,23) TEMPERATURE[1,N], TEMPERATURE[2,N], TEMPERATURE
    [3,N], TEMPERATURE[4,N]
WRITE (unit,23) VOLUME
DO 11 N=1, 3
11 WRITE (unit,23) BOX[1,N], BOX[2,N], BOX[3,N]]
DO 12 N=1, 3
12 WRITE (unit,23) PRESSURE[N]
DO 13 N=1, 3
13 WRITE (unit,23) PRESS[1,N], PRESS[2,N], PRESS[3,N]
DO 14 N=1, 3
14 WRITE (unit,23) VIRIAL[1,N], VIRIAL[2,N], VIRIAL[3,N]
DO 15 N=1, 3
15 WRITE (unit,23) KINETIC[1,N], KINETIC[2,N], KINETIC[3,N]

```

Free energy derivative lambda block

Blockname: FREEENERGYDERIVS03

```

WRITE (unit, 23)RLAM
DO 10 N=1,38
10 WRITE (unit,23) FREEENER[N]
WRITE (unit,22) NBATHS
DO 11 N=1,NBATHS
11 WRITE (unit,23) FREEKINENER[1,N], FREEKINENER[2,N], FREEKINENER
    [3,N]
WRITE (unit,22) NEGR
DO 12 N=1, NEGR
12 WRITE (unit,23) FREEBONDED[1,N],FREEBONDED[2,N],FREEBONDED[3,N],
    FREEBONDED[4,N],FREEBONDED[5,N]
DO 13 N=1, NEGR*(NEGR+1)/2
13 WRITE (unit23) FREENONBONDED[1,N],FREENONBONDED[2,N],
    FREENONBONDED[3,N],FREENONBONDED[4,N]
DO 14 N=1, NEGR
14 WRITE (unit, 23)FREESPECIAL[1,N], FREESPECIAL[2,N], FREESPECIAL[3,N]
    FREESPECIAL[4,N],FREESPECIAL[5,N],FREESPECIAL[6,N]
    FREESPECIAL[7,N],FREESPECIAL[8,N],FREESPECIAL[9,N]
    FREESPECIAL[10,N],FREESPECIAL[11,N]
DO 15 N=1,NEDS
15 WRITE (unit,23) FREEEDSENER[1,N],EDSENER[2,N],EDSENER[3,N]
22 FORMAT (I5)
23 FORMAT (11E17.9)

```

4.18. Atomic B-factors and positional fluctuations

Atomic mobilities or positional fluctuations can be stored in the form of isotropic B-factors

$$B_i = (8\pi^2/3) \langle (r_i - \langle r_i \rangle)^2 \rangle .$$

The quantities characterizing fluctuations of coordinate distributions are the following:

NR	number of atoms
X[1..3, 1..NR]	atomic Cartesian coordinates

BFAC[1..NR] atomic isotropic B-factors

The B-factor or fluctuation blocks are the following:

Isotropic B-factor block

Blockname: BFACTOR

Formatted form

```
DO 10 J=1, NR
10 WRITE (unit,25) MRES[J], AANMA[J], PANM[J], J, BFAC[J]
25 FORMAT (I5,2(1X,A5),I7,6F9.5)
```

4.19. Accelerated EDS parameter search data

For accelerated EDS parameter search simulations, several data are stored to allow for continuation of the search in a new run. The quantities are:

AEDSS[1]	accelerated EDS parameter E_{max}
AEDSS[2]	accelerated EDS parameter E_{min}
AEDSS[3]	current maximum transition energy within this state-visit period
AEDSS[4]	number of found maximum transitions energies (i.e. number of completed state-visit periods)
AEDSS[5]	number of the end-state sampled in the last simulation step
AEDSS[6]	should E_{min} be allowed to be smaller than the average energy of the end-state with the lowest energy?
NEDS	number of EDS end-states
AEDSS[7,1..NEDS]	energy offset parameter per EDS state
AEDSS[8,1..NEDS]	natural logarithm of the exponentially averaged energy difference between the accelerated EDS reference state and this accelerated end-state
AEDSS[9,1..NEDS]	free-energy difference between the accelerated EDS reference state and this accelerated end-state
AEDSS[10,1..NEDS]	has this state already been visited within the current state-visit period?
AEDSS[11,1..NEDS]	number of visits of this end-state
AEDSS[12,1..NEDS]	average energy of this end-state
AEDSS[13,1..NEDS]	average of the energy of this end-state minus the energy offset parameter of this end-state
AEDSS[14,1..NEDS]	helper variable for the calculation of the standard deviation of the energy of this end-state
AEDSS[15,1..NEDS]	standard deviation of the energy of this end-state

The accelerated EDS parameter search block is the following:

Accelerated EDS parameter search block

Blockname: AEDSS

Formatted form

```
      DO 10  N=1,6
10    WRITE  (unit,24) AEDSS[N]
      DO 11  N=1,NEDS
11    WRITE  (unit,42) AEDSS[7,N],AEDSS[8,N],AEDSS[9,N],AEDSS[10,N],AEDSS[11,N],
                AEDSS[12,N],AEDSS[13,N],AEDSS[14,N],AEDSS[15,N]
24    FORMAT (3F15.9,3I15)
42    FORMAT (3F15.9,2I15,4F15.9)
```

4.20. Backwards compatibility with GROMOS96

The changes with respect to GROMOS96 have been detailed above. They consist of alterations in a number of blocks, introduction of a number of new blocks, and the deletion of a number of blocks.

For the alterations:

the GROMOS96 blocks

have been replaced by

BOX,
ENERGY,
VOLUMEPRESSURE,
FREEENERGYLAMBDA
and
FREEENERGY3D4

GENBOX,TRICLINICBOX
ENERGIES,ENERY03
RUNDATA,VOLUMEPRESSURE03
FREELAMBDA,DATA,FREEENERGY-DERIVS03
and
FREE3D4DDATA respectively.

Additional blocks include:

SHAKEFAILPOSITION,
SHAKEFAILPREVPOSITION,
LATTICESHIFTS,
FREEFORCE,
FREEFORCERED,

CONSFORCE,
CONSFORCERED,
NHCVARIABLES.

Deleted blocks include:

BFACTORANISO,
POSITIONSEDONDM,
POSITIONTHIRDM,
POSITIONFOURTHM,
POSITIONSECONDMT,
QUANTITYAVER,
QUANENEAVEVER,

QUANSUMENEAVEVER,
QUANTIMESERIES,
QUANDISTRIB,
QUANTIMECORR and
QUANTIMECORRSPE.

Molecular topology building blocks

5.1. Introduction

Most programs of GROMOS do require a molecular topology file containing the topological and force field data concerning the molecular system that is considered. Specifying a complete molecular topology for a large molecule like a protein is a tedious task. Long lists of atomic properties have to be typed. Therefore, GROMOS contains a program *make_top* that is able to generate a complete molecular topology from *molecular topology building blocks*, that is, molecules or parts of molecules like amino acid residues, nucleotides, etc., which are constituting the molecular system that is considered. The building blocks are linked in order to form the wanted molecular topology.

Linking of building blocks consisting of *separate*, non-covalently connected, *molecules* is straightforward. This will be discussed in Sec. 5.2 together with the content and format of a *molecular topology building block file*. The linking of *covalently connected building blocks* by *make_top* demands a set of rules to be satisfied by the molecular topology building blocks. These rules will be discussed in Sec. 5.3 to Sec. 5.5.

Reading a molecular topology building block file occurs in programs:

`make_top`, `check_top`.

Examples of molecular topology building block files are named:

*.mtb

5.2. Separate molecules

A molecular topology building block file contains two types of information.

1. information regarding all building blocks:

FPEPSI	$(4\pi\epsilon_0)^{-1}$, ϵ_0 = permittivity of vacuum
HBAR	$\hbar = h/(2\pi)$, h = Planck's constant
SPDL	c = speed of light
BOLTZ	k_B = Boltzmann's constant

2. information specifying a building block:

L	sequence number of the solute building block in the molecular topology building block file; below, the <i>primary sequence number</i> of the residue or nucleotide in the protein or polynucleotide is denoted by M
RNME [L]	name of residue or nucleotide or molecule (at most 5 characters); these names are to be used in the input of <i>make_top</i> to select building blocks
NMAT [L]	number of atoms

NLIN[L] number of atoms for which the exclusions are given before the exclusions of the building block atoms themselves:
 = 0 for a separate molecule
 \neq 0 for a residue, nucleotide or monosaccharide unit

ANM[1..NMAT[L], L] atom names (at most 5 characters)

IMCM[1..NMAT[L], L] integer mass type codes for selection of atomic masses

ATOM[1..NMAT[L], L] atom sequence number

NREP[L] number of atoms that are replacing existing atoms at the beginning (NREP[L]>0) or at the end (NREP[L]<0) of a chain of building blocks

IACM[1..NMAT[L], L] integer atom type codes for selection of van der Waals parameters

CGM[1..NMAT[L], L] atomic charges

ICGM[1..NMAT[L], L] atomic charge group codes; the atoms forming a charge group must have sequential sequence numbers; the last atom of a charge group is denoted by ICGM=1, the others must have ICGM=0

MAE[1..NMAT[L], L] number of neighbours of atom i that are excluded from non-bonded interaction with atom i

MSAE[1..MAE[ATOM], 1..NMAT[L], L] atom sequence numbers j of the excluded neighbours of atom i; it is required that $i < j$ and that the j's occur in ascending order. The exclusions of the last NLIN[L] atoms of the building block are not specified here, but handled in the next building block of the molecular chain.
 Exclusions between different building blocks that do not go over the systems main chain (e.g. disulfide bridges, covalently bound Heme groups) are described in section Sec. 5.3.

NCGB[L] number of coarse-grained regions

NRCGF[1..NCGB], NRCGL[1..NCGB] first and last atom sequence number of a coarse-grained region

MSCAL[1..NCGB] scaling factor for pressure correction of a coarse-grained region

NPPOL[L] number of polarisable solute atoms

IPOLP[1..NPPOL], ALPP[1..NPPOL] atom sequence number and polarisability of the polarisable solute atom

QPOLP[1..NPPOL] size of charge-on-spring connected to polarisable solute atoms

ENOTP[1..NPPOL], EPP[1..NPPOL] damping level and power for polarisation

GAMP[1..NPPOL], IP[1..NPPOL], JP[1..NPPOL] gamma and the first and second atom for off-site polarisable centre construction

NEX[L] number of LJ-exceptions

AEX[1..2, 1..NEX[L], L]
 atom sequence numbers of the atoms to have special LJ-interactions defined
 by LJ-exceptions

AEXTYPE[1..NEX[L], L]
 LJ-exception type codes for selection of interaction parameters

NMB[L] number of bonds

MB[1..2, 1..NMB[L], L]
 atom sequence numbers of the atoms forming the bonds i-j, i is always smaller
 than j

MCBL[1..NMB[L], L] bond-type codes for selection of interaction parameters

NMBDP[L] number of dipole bonds

MBDP[1..2, 1..NMBDP[L], L]
 atom sequence numbers of the atoms forming the dipole bonds i-j, i is always
 smaller than j

NMBA[L] number of bond angles

MBA[1..3, 1..NMBA[L], L]
 atom sequence numbers of the atoms forming the bond angles i-j-k, i is always
 smaller than k

MCBA[1..NMBA[L], L]
 bond-angle type codes for selection of interaction parameters

NMIDA[L] number of improper dihedral angles

MIDA[1..4, 1..NMIDA[L], L]
 atom sequence numbers of the atoms forming the improper dihedrals i-j-k-l,
 j is always smaller than k

MCIA[1..NMIDA[L], L]
 improper dihedral angle type codes for selection of interaction parameters

NMDA[L] number of dihedral angles

MDA[1..4, 1..NMDA[L], L]
 atom sequence numbers of the atoms forming the dihedrals i-j-k-l, j is always
 smaller than k

MCDA[1..NMDA[L], L]
 dihedral angle type codes for selection of interaction parameters

LL sequence number of the solvent building block in the molecular topology
 building block file

RNMES[LL] name of solvent molecule (at most 5 characters): one of these names is used
 in the input of make_top to select a solvent building block

NMATS[LL] number of atoms

ANMMS[1..NMATS[LL], LL]
 atoms names (at most 5 characters)

IMCMS[1..NMATS[LL], LL]
 integer mass type codes for selection of atomic masses

IACMS[1..NMATS[LL], LL]
 integer atom type codes for selection of van der Waals parameters
 CGMS[1..NMATS[LL], LL]
 atomic charges
 NVPOL[L]
 number of polarisable solvent atoms
 IPOLV[1..NVPOL], ALPV[1..NVPOL]
 atom sequence number and polarisability of the polarisable solvent atom
 QPOLV[1..NVPOL] size of charge-on-spring connected to polarisable solvent atoms
 ENOTV[1..NPVOL], EPV[1..NVPOL]
 damping level and power for polarisation
 GAMV[1..NVPOL], IV[1..NVPOL], JV[1..NVPOL]
 gamma and the first and second atom for off-site polarisable centre construction
 NCONM[LL]
 number of distance constraints
 ICONM, JCONM[1..NCONM[LL], LL]
 atom sequence numbers of the atoms forming the constraint i-j, i is always smaller than j
 CONM[1..NCONM[LL], LL]
 constraint length of the constraint i-j

The blocks of a *molecular topology building block file* are (apart from the *Title block*) the following:

Physical constants block

Blockname: PHYSICALCONSTANTS

```

WRITE (unit,12) FPEPSI
WRITE (unit,12) HBAR
WRITE (unit,12) SPDL
WRITE (unit,12) BOLTZ
12  FORMAT (E15.7)

```

Molecular topology solute building block

Blockname: MTBUILDBLSOLUTE

```

WRITE (unit,30) RNME[L]
WRITE (unit,33) NMAT[L], NLIN[L]

```

for every preceding exclusion I:

```

DO 19 I=1, NLIN[L]
19  WRITE (unit,31) I-NLIN[L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

```

for atom $I < \text{NMATL}[L] - \text{NLIN}[L]$:

```

DO 20 I=1, NMAT[L]-NLIN[L]
20  WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

```

for every atom I with $\text{NMAT}[L] - \text{NLIN}[L] < I \leq \text{NMAT}[L]$:

```

DO 21 I=1, NMAT[L]-NLIN[L] + 1, NMAT[L]
21  WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L]

```

```

WRITE (unit,33) NLJEX[L]
for every LJ-exception N:
DO 22 N=1, NLJEX[L]
22 WRITE (unit,33) (AEX[M,N,L], M=1,2), AEXTYPE[N,L]

```

```

WRITE (unit,33) NMB[L]
for every bond N:
DO 23 N=1, NMB[L]
23 WRITE (unit,33) (MB[M,N,L], M=1,2), MCBL[N,L]

```

```

WRITE (unit,33) NMBA[L]
for every bond angle N:
DO 24 N=1, NMBA[L]
24 WRITE (unit,33) (MBA[M,N,L], M=1,3), MCBA[N,L]

```

```

WRITE (unit,33) NMIDA[L]
for every improper dihedral angle N:
DO 25 N=1, NMIDA[L]
25 WRITE (unit,33) (MIDA[M,N,L], M=1,4), MCIA[N,L]

```

```

WRITE (unit,33) NMDA[L]
for every proper torsional dihedral N:
DO 26 N=1, NMDA[L]
26 WRITE (unit,33) (MDA[M,N,L], M=1,4), MCDA[N,L]
30 FORMAT (A5)
31 FORMAT (I5,30X,I4,8I5)
32 FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33 FORMAT (16I5)

```

If $MAE[I,L] > 8$, then the remaining MSAE values are written on the next line using 16I5 as format.

Molecular topology coarse-grained (CG) solute building block
Blockname: MTBUILDBLCSOLUTE

```

WRITE (unit,30) RNME[L]
WRITE (unit,33) NMAT[L], NLIN[L]

```

```

for every preceding exclusion I:
DO 19 I=1, NLIN[L]
19 WRITE (unit,31) I-NLIN[L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

```

```

for atom I < NMATL[L]-NLIN[L]:
DO 20 I=1, NMAT[L]-NLIN[L]
20 WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
ICGM[I,L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

```

```

for every atom I with NMAT[L]-NLIN[L] < I ≤ NMAT[L]:
DO 21 I=1, NMAT[L]-NLIN[L] + 1, NMAT[L]
21 WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
ICGM[I,L]

```

```

WRITE (unit,33) NCGB[L]
for every coarse-grained region N:
DO 22 N=1, NCGB[L]

```

```

22  WRITE (unit,34) NRCGF[N], NRCGL[N], MSCAL

      WRITE (unit,33) NLJEX[L]
for every LJ-exception N:
      DO 23  N=1, NLJEX[L]
23  WRITE (unit,33) (AEX[M,N,L], M=1,2), AEXTYPE[N,L]

      WRITE (unit,33) NMB[L]
for every dipole bond N:
      DO 24  N=1, NMBDP[L]
24  WRITE (unit,33) (MBDP[M,N,L], M=1,2), MCBL[N,L]

      WRITE (unit,33) NMBA[L]
for every bond N:
      DO 24  N=1, NMB[L]
24  WRITE (unit,33) (MB[M,N,L], M=1,2), MCBL[N,L]

      WRITE (unit,33) NMBA[L]
for every bond angle N:
      DO 25  N=1, NMBA[L]
25  WRITE (unit,33) (MBA[M,N,L], M=1,3), MCBA[N,L]

      WRITE (unit,33) NMIDA[L]
for every improper dihedral angle N:
      DO 26  N=1, NMIDA[L]
26  WRITE (unit,33) (MIDA[M,N,L], M=1,4), MCIA[N,L]

      WRITE (unit,33) NMDA[L]
for every proper torsional dihedral N:
      DO 27  N=1, NMDA[L]
27  WRITE (unit,33) (MDA[M,N,L], M=1,4), MCDA[N,L]
30  FORMAT (A5)
31  FORMAT (I5,30X,I4,8I5)
32  FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33  FORMAT (16I5)
34  FORMAT (2I5,F15.7)

```

If $MAE[I,L] > 8$, then the remaining MSAE values are written on the next line using 16I5 as format.

Molecular topology polarisable solute building block

Blockname: MTBUILDBLPOLSOLUTE

```

      WRITE (unit,30) RNME[L]
      WRITE (unit,33) NMAT[L], NLIN[L]

for every preceding exclusion I:
      DO 19  I=1, NLIN[L]
19  WRITE (unit,31) I-NLIN[L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

for atom I < NMATL[L]-NLIN[L]:
      DO 20  I=1, NMAT[L]-NLIN[L]
20  WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

for every atom I with NMAT[L]-NLIN[L] < I ≤ NMAT[L]:

```

```

DO 21 I=1, NMAT[L]-NLIN[L] + 1, NMAT[L]
21 WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L]

WRITE (unit,33) NCGB[L]
for every polarisable atom I:
DO 22 I=1, NPPOL[L]
22 WRITE (unit,35) IPOLP[I], ALPP[I], QPOLP[I], ENOTP[I], EPP[I], GAMP[I], IP[I], JP[I]

WRITE (unit,33) NLJEX[L]
for every LJ-exception N:
DO 23 N=1, NLJEX[L]
23 WRITE (unit,33) (AEX[M,N,L], M=1,2), AEXTYPE[N,L]

WRITE (unit,33) NMB[L]
for every bond N:
DO 24 N=1, NMB[L]
24 WRITE (unit,33) (MB[M,N,L], M=1,2), MCBL[N,L]

WRITE (unit,33) NMBA[L]
for every bond angle N:
DO 25 N=1, NMBA[L]
25 WRITE (unit,33) (MBA[M,N,L], M=1,3), MCBA[N,L]

WRITE (unit,33) NMIDA[L]
for every improper dihedral angle N:
DO 26 N=1, NMIDA[L]
26 WRITE (unit,33) (MIDA[M,N,L], M=1,4), MCIA[N,L]

WRITE (unit,33) NMDA[L]
for every proper torsional dihedral N:
DO 27 N=1, NMDA[L]
27 WRITE (unit,33) (MDA[M,N,L], M=1,4), MCDA[N,L]
30 FORMAT (A5)
31 FORMAT (I5,30X,I4,8I5)
32 FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33 FORMAT (16I5)
35 FORMAT (I5,5F11.5,2I5)

```

If MAE[I,L] > 8, then the remaining MSAE values are written on the next line using 16I5 as format.

Molecular topology solute end group building block
Blockname: MTBUILDBLEND

```

WRITE (unit,30) RNME[L]
WRITE (unit,33) NMAT[L], NREP[L]

if NREP[L] > 0 then
for atom I ≤ NMATL[L]-NREP[L]:
DO 20 I=1, NMAT[L]-NREP[L]
20 WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])
DO 21 I=NMAT[L]-NREP[L]+1, NMAT[L]
21 WRITE (unit,32) I, ANM[I,L], IACM[I,L], ICMCM[I,L], CGM[I,L],
      ICGM[I,L]

```

```

if NREP[L] < 0 then
for every atom I
  DO 22 I= NMAT[L]-NREP[L]+1, NMAT[L]
22  WRITE (unit,32) I, ANM[I,L], IACM[I,L], IMCM[I,L], CGM[I,L],
      ICGM[I,L], MAE[I,L], (MSAE[N,I,L],N=1,MAE[I,L])

      WRITE (unit,33) NMB[L]
for every bond N:
  DO 23 N=1, NMB[L]
23  WRITE (unit,33) (MB[M,N,L], M=1,2), MCBL[N,L]

      WRITE (unit,33) NMBA[L]
for every bond angle N:
  DO 24 N=1, NMBA[L]
24  WRITE (unit,33) (MBA[M,N,L], M=1,3), MCBA[N,L]

      WRITE (unit,33) NMIDA[L]
for every improper dihedral angle N:
  DO 25 N=1, NMIDA[L]
25  WRITE (unit,33) (MIDA[M,N,L], M=1,4), MCIA[N,L]

      WRITE (unit,33) NMDA[L]
for every proper torsional dihedral N:
  DO 26 N=1, NMDA[L]
26  WRITE (unit,33) (MDA[M,N,L], M=1,4), MCDA[N,L]
30  FORMAT (A5)
31  FORMAT (I5,30X,I4,8I5)
32  FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33  FORMAT (16I5)

```

If MAE[I,L] > 8, then the remaining MSAE values are written on the next line using 16I5 as format.

Molecular topology solvent building block

Blockname: MTBUILDBLSOLVENT

```

      WRITE (unit,30) RNMES[LL]
      WRITE (unit,33) NMATS[LL]
      DO 20 J=1, NMATS[LL]
20  WRITE (unit,32) J, ANMMS[J,LL], IACMS[J,LL], IMCMS[J,LL],
      CGMS[J,LL]

      WRITE (unit,33) NCONM[LL]
      DO 21 N=1, NCONM[LL]
21  WRITE (unit,34) ICONM[N,LL], JCONM[N,LL], CONM[N,LL]
30  FORMAT (A5)
32  FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33  FORMAT (16I5)

34  FORMAT (2I5,F15.7)

```

Molecular topology polarisable solvent building block

Blockname: MTBUILDBLPOLSOLVENT

```

WRITE (unit,30) RNMES[LL]
WRITE (unit,33) NMATS[LL]
DO 20 J=1, NMATS[LL]
20 WRITE (unit,32) J, ANMMS[J,LL], IACMS[J,LL], IMCMS[J,LL],
      CGMS[J,LL]
WRITE (unit,33) NVPOL[LL]
DO 21 I=1, NVPOL[LL]
21 WRITE (unit,35) IPOLV[J], ALPV[J], QPOLV[J], ENOTV[J], EPV[J], GAMV[J], IV[J], JV[J]

WRITE (unit,33) NCONM[LL]
DO 22 N=1, NCONM[LL]
22 WRITE (unit,34) ICONM[N,LL], JCONM[N,LL], CONM[N,LL]
30 FORMAT (A5)
32 FORMAT (I5,1X,A5,I4,I5,F11.5,2I4,8I5)
33 FORMAT (16I5)

34 FORMAT (2I5,F15.7)
35 FORMAT (I5,5F11.5,2I5)

```

5.3. Linking of building blocks

When several building blocks need to be covalently linked to obtain the required molecular topology, a few rules must be satisfied compared to the case of separate molecules. These rules are due to the fact that in a chain of building blocks the bonds, bond-angles and (improper) dihedral angles involve atoms from *different* building blocks. Also excluded neighbours may reside in different building blocks. These rules are the following:

- When listing a bond (i-j), bond-angle (i-j-k), improper dihedral (i-j-k-l) or dihedral (i-j-k-l) connecting atoms with sequence numbers i, j, k or l in two residues with residue sequence numbers M-1 and M or M and M+1, through a *peptide C-N link*, the following rules apply:
 - for the bond i-j, neither i nor j may lie in residue M-1 and only j may lie in residue M+1;
 - for the bond-angle i-j-k, only i may lie in residue M-1, and only k may lie in residue M+1;
 - for the improper dihedral i-j-k-l, only j or k may lie in residue M-1, and only i or j or k or l may lie in residue M+1;
 - for the dihedral i-j-k-l, only i or i and j may lie in residue M-1. When i and j lie in residue M-1, atom i always refers to the first atom bound to j with $i < j$, it can be specified by -2. Only l may lie in residue M+1.
- Cross links such as disulfide bridges or the covalent link between a histidine and the heme group can be made between different building blocks. For a cross-link between building blocks M and N ($M < N$) the rules for listing the bond, bond-angles and dihedrals are the following:
 - In M, the atoms of the building block N are identified by a negative sign of atom sequence numbers
 - for the bond i-j, only in M -j may denote atom j in N
 - for the bond-angle i-j-k, only in M -k or -j and -k may denote atoms j and k in N
 - for the dihedral i-j-k-l, only in M -l, or -h and -l or -j, -k and -l may denote atoms j,k and l in N
 - for the improper dihedral i-j-k-l, only in M -i or -l may denote atoms i and l in N
 - In M, the excluded neighbours residing in N are denoted by a negative sign of their atom sequence numbers
- More general cross links between molecules can be made by post-processing an unlinked topology using GROMOS++ program *link_top*. The link is defined according to the MTBUILDBLLINK block outlined below. The rules for creating cross links with *link_top* are the following:
 - the atoms are identified in the original topology by the residue sequence number indicated in the input and the name of the atom specified in the MTBUILDBLLINK block

- all atoms for which the IAC is set to 0 will be removed from the original topology. For all remaining atoms specified in the MTBUILDBLLINK block, the values for the IAC, the MASS, the CHARGE and the charge group code are updated. Note that the actual MASS is given in the MTBUILDBLLINK file and not the integer mass type code
- the exclusions of the original topology (without the removed atom) remain, and exclusions that are specified in the MTBUILDBLLINK block are *added*
- all covalent interactions that are specified in the MTBUILDBLLINK block are first removed from the original topology (if present) and subsequently added according to the current definitions.
- for dihedral angles, *link_top* allows the user to refer to the first and/or last atom with an atom sequence number 0. For these atoms, the program will search in the topology that is bound to the second or third atom in the dihedral angle definition, respectively, and assign the dihedral angle to this atom. Any dihedral angles that were already defined for this group is replaced.

Molecular topology solute building block

Blockname: MTBUILDBLLINK

```

WRITE (unit,30) RNME[L]
WRITE (unit,33) NMAT[L]

DO 20 I=1, NMAT[L]
20 WRITE (unit,32) I, RES[I,L], ANM[I,L], IACM[I,L], MASS[I,L],
           CGM[I,L], ICGM[I,L], MAE[I,L], (MSAE[N,I,L], N=1, MAE[I,L])

DO 23 N=1, NMB[L]
23 WRITE (unit,33) (MB[M,N,L], M=1,2), MCBL[N,L]

WRITE (unit,33) NMBA[L]
DO 24 N=1, NMBA[L]
24 WRITE (unit,33) (MBA[M,N,L], M=1,3), MCBA[N,L]

WRITE (unit,33) NMIDA[L]
DO 25 N=1, NMIDA[L]
25 WRITE (unit,33) (MIDA[M,N,L], M=1,4), MCIA[N,L]

WRITE (unit,33) NMDA[L]
DO 26 N=1, NMDA[L]
26 WRITE (unit,33) (MDA[M,N,L], M=1,4), MCDA[N,L]
30 FORMAT (A5)
31 FORMAT (I5,30X,I4,8I5)
32 FORMAT (I5,1X,I5,A5,I4,F10.5,F11.5,2I4,8I5)
33 FORMAT (16I5)

```

An example of a molecular topology building block containing an amino acid residue is the Alanine building block to be found under the name ALA in the *.mtb files.

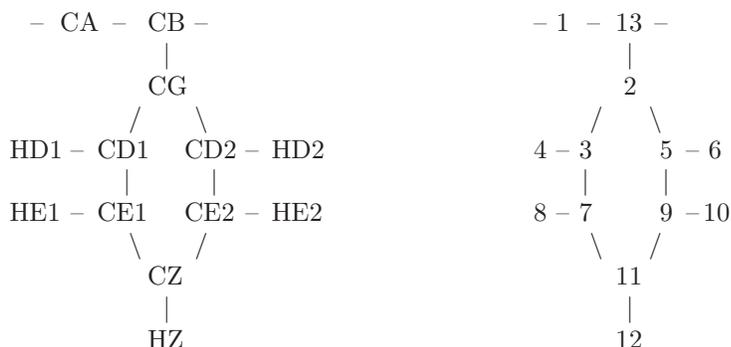
An example of a molecular topology building block containing a nucleotide is the Adenosine building block to be found under the name DADE in the *.mtb files.

An example of a molecular topology building block containing a glucose unit is the sugar building block to be found under the name GLCA in the *.mtb files.

5.4. Other building blocks

From the previous paragraphs it has become clear that program *make_top* may link any type of molecular topology building blocks into a linear co-valently connected chain, as long as the characteristics of the link

satisfy the given rules. A Styrene residue topology building block may serve as an example.



By choosing the displayed atom sequence numbers, all the rules for connecting these building blocks into a polystyrene chain using *make_top* are satisfied.

It should always be checked in the complete molecular topology generated from the building blocks by program *make_top* whether the linking has been carried out correctly.

5.5. End groups

The linking of molecular topology building blocks has been described in the previous sections. This leaves open the question of how to treat the head and tail of the molecular chain one is interested in. Often the bonded and nonbonded parameters at the head and tail of e.g. a protein will be different from the parameters that are usually describing amino acid interactions. For this reason GROMOS knows the end-group building blocks, which describe which atoms to change and add or remove at the head and tail of the sequence. The following rules apply:

- if $NREP > 0$, the end-group building block describes the head of the chain. The last $NREP$ atoms will replace the first $NREP$ atoms in the next building block. Only the exclusions that are specified remain from the next building block, all other parameters are overwritten.
- if $NREP < 0$, the end-group building block describes the tail of the chain. The first $-NREP$ atoms will replace the last $-NREP$ atoms in the previous building block.
- in order to remove atoms from the next or previous building block they need to be specified in the end-group building block as a replacing atom with a negative IAC
- for all covalent interactions that cross between an end-group and a regular building block the same rules apply as for linking two building blocks.

Examples of protein end-group building blocks can be found under the names NH_3^+ or $COOH$ in the *.mtb files.

Examples of nucleotide end-group building blocks can be found under the names $D3OH$ and $D5OH$ in the *.mtb files.

Examples of saccharide end-group building blocks can be found under the names $C1OH$ and $C6OH$ in the *.mtb files.

It should always be checked in the complete molecular topology generated from the building blocks by program *make_top* whether the end-groups have been implemented correctly.

5.6. Contents of the MTB file

MTB file:

TITLE
LINKEXCLUSIONS
PHYSICALCONSTANTS
MTBUILDBLSOLUTE
MTBUILDBLSOLVENT
MTBUILDBLEND

Interaction function parameters

6.1. Introduction

The molecular topology file of a molecular system does not only contain topological information about the system, but also force field parameters. These parameters have been listed in Chap. 3 (Force Field and Topology Data Set). They are part of a molecular topology as described in Sec. 3.2. The molecular topology can be generated using the program *make_top*. All the force field parameters that belong to a specific force field are kept in two different files. The force field parameters that are related to the molecular topology, like atomic charges and third or excluded nearest neighbour information, are included in the *molecular topology building block file*, which has been described in Sec. 5.2. The remaining force field parameters, which are independent of the molecular topology, are kept in another file, the *interaction function parameter file*. Both files are combined by program *make_top* to generate a complete molecular topology file (Sec. 3.2) corresponding to the molecular system that is considered.

The various blocks of an interaction function parameter file are (apart from the *Title block*) described in CSec. 6.2 to Sec. 6.7.

Reading an interaction parameter file occurs in programs:

`make_top`, `check_top`, `con_top`

Examples of interaction function parameter files are named:

`*.ifp`

6.2. Mass atom types

The mass atom type codes, mass values and names are stored as follows:

<code>NRMATY</code>	number of defined mass atom types
<code>NMATY</code>	largest (integer) mass atom type code
<code>ATMAS [1..NMATY]</code>	atomic mass as a function of the (integer) mass atom type code
<code>ATMASN [1..NMATY]</code>	(mass) atom names as a function of the (integer) mass atom type code (at most 5 characters)

Mass atom type code block

Blockname: MASSATOMTYPECODE

```

WRITE (unit,11) NRMATY, NMATY
DO 10 M=1, NRMATY
10 WRITE (unit,12) M, ATMAS[M], ATMASN[M]
11 FORMAT (2I5)
12 FORMAT (I5,F10.5,1X,A5)

```

6.3. Covalent bond-stretching interaction parameters

The parameters concerning the bond-stretching interaction are stored as follows:

NRBTY	number of defined covalent bond types
NBTY	largest (integer) bond-type code
CB[1..NBTY]	force constant of the bond-stretching term of the interaction as a function of the bond-type code, based on a quartic potential energy function
CHB [1..NBTY]	force constant of the bond-stretching term of the interaction as a function of the bond-type code, based on a harmonic potential energy function
BO[1..NBTY]	bond length at minimum energy of the bond-stretching term as a function of the bond-type code

Bond-type code and parameters block

Blockname: BONDSTRETCHTYPECODE

```
      WRITE (unit,11) NRBTY, NBTY
      DO 10 M=1, NRBTY
10    WRITE (unit,13) N, CB[N], CHB [N], BO[N]
13    FORMAT (I5,3F15.7)
```

6.4. Covalent bond-angle bending interaction parameters

The parameters concerning the bond-angle bending interaction are stored as follows:

NRTTY	number of defined bond-angle types
NTTY	largest (integer) bond-angle type code
CT[1..NRTTY]	force constant of the bond-angle bending term of the interaction as a function of the bond-angle type code, based on a potential energy function harmonic in the angle cosine
CHT[1..NRTTY]	force constant of the bond-angle bending term of the interaction as a function of the bond-angle type code, based on a potential energy function harmonic in the angle (in energy units per degree ²)
TO[1..NRTTY]	bond angle (in degrees) at minimum energy of the bond-angle bending term as a function of the bond-angle type code

Bond-angle type code and parameters block

Blockname: BONDANGLEBENDTYPECODE

```
      WRITE (unit,11) NRTTY, NTTY
      DO 10 M=1, NRTTY
10    WRITE (unit,13) N, CT[N], CHT [N], TO[N]
13    FORMAT (I5,3F15.7)
```

6.5. Improper dihedral-angle interaction parameters

The parameters concerning the harmonic improper dihedral-angle interaction are stored as follows:

NRQTY	number of defined (harmonic) improper dihedral-angle types
NQTY	largest (integer) improper dihedral-angle type code
CQ[1..NQTY]	force constant of the harmonic improper dihedral term of the interaction as a function of the improper dihedral-angle type code (in energy units per degree ²)
QO[1..NQTY]	improper dihedral-angle (in degrees) at minimum energy of the harmonic improper dihedral term as a function of the improper dihedral-angle type code

Improper (harmonic) dihedral-angle type code and parameters block

Blockname: IMPDIHEDRALTYPECODE

```

WRITE (unit,11) NRQTY, NQTY
DO 10 M=1, NRQTY
10 WRITE (unit,13) N, CQ[N], QO[N]
13 FORMAT (I5,3F15.7)

```

6.6. Dihedral-angle torsional interaction parameters

The parameters concerning the trigonometric dihedral-angle interaction are stored as follows:

NRPTY	number of defined (trigonometric) dihedral-angle types
NPTY	largest (integer) dihedral-angle type code
CP[1..NPTY]	force constant of the trigonometric dihedral term of the interaction as a function of the dihedral-angle type code
PDL [1..NPTY]	phase-shift angle (in degrees) of the trigonometric dihedral term as a function of the dihedral-angle type code
NP[1..NPTY]	multiplicity of the trigonometric dihedral term as a function of the dihedral-angle type code (1,2,3,4,5,6)

Proper (trigonometric) dihedral-angle type code and parameters block

Blockname: TORSDIHEDRALTYPECODE

```

WRITE (unit,11) NRPTY, NPTY
DO 10 M=1, NRPTY
10 WRITE (unit,14) N, CP[N], PDL[N], NP[N]
14 FORMAT (I5,2F10.5,I5)

```

6.7. Van der Waals interaction parameters and integer atom codes

The interaction function parameter file contains information on the van der Waals interaction parameters. These are $C_{12}(i,j)$, the coefficient of the $1/r^{12}$ term, and $C_6(i,j)$, the coefficient of the $-1/r^6$ term in the non-bonded interaction. These coefficients depend on the integer atom codes I and J (1..NRATT) of atoms i and j. In a molecular topology file these parameters are stored in the arrays C12, C6[1..NRATT*(NRATT+1)/2]. The corresponding parameters for the 1-4 or third neighbour non-bonded interaction are stored in arrays CS12, CS6 [1..NRATT*(NRATT+1)/2].

If NRATT is large, direct specification of all these parameters becomes tedious. Therefore, the information on the van der Waals parameters is stored in a different manner.

1. A first block contains single atom type normal van der Waals parameters $C_6^{1/2}$ (I,I) and (maximally 3 values) $C_{12}^{1/2}$ (I,I) and third-neighbour parameters $C_6^{1/2}$ (I,I) and $C_{12}^{1/2}$ (I,I), from which the van der Waals parameters for all atom pairs are calculated (in *make_top*) using geometric combination rules.
2. A second block contains van der Waals parameters for a given set of atom pairs, which will replace the combination rule values (upon reading in *make_top*).

The following variables are used to define the van der Waals interaction parameters:

NRATT	number of (van der Waals) atom types
TYPE[1..NRATT]	names of the different atom types as a function of the integer atom code that defines an atom type (at most 5 characters)
C612[1..NRATT]	square root of the single atom coefficient of the $-1/r^6$ term in the normal van der Waals interaction as a function of the integer atom code
C1212[1..NRATT, 1..3]	three values for the square root of the single atom coefficient of the $1/r^{12}$ term in the normal van der Waals interaction as a function of the integer atom code
LPAIR[1..NRATT, 1..NRATT]	pointer matrix for selection of one of the three C1212 values when applying the combination rules: $C12 [I,J] = C1212[I, LPAIR[I,J]] * C1212[J, LPAIR[J,I]]$; $LPAIR[I,J] = 1, 2$ or 3
CS612[1..NRATT]	square root of the single atom coefficient of the $-1/r^6$ term in the third-neighbour van der Waals interaction as a function of the integer atom code
CS1212[1..NRATT]	square root of the single atom coefficient of the $1/r^{12}$ term in the third-neighbour van der Waals interaction as a function of the integer atom code
NRPAIR	number of atom type pairs for which the van der Waals parameters are explicitly given
MPAC[1..NRATT, 1..NRATT]	pair codes for atom pairs as a function of their integer atom codes I and J (\leq NRATT * (NRATT+1)/2 = NRATT2), the pair code is defined as $I+J*(J-1)/2$ when $I \leq J$ and as $J+I*(I-1)/2$ when $J \leq I$; these values are not stored in the interaction function parameter file, since they can be and are calculated upon reading the file
C12[1..NRATT*(NRATT+1)/2]	coefficient of the $1/r^{12}$ term in the non-bonded interaction as a function of the occurring pair codes; so, the sequence of atom pairs with integer atom codes ranging from 1 to NRATT is: 1-1, 1-2, 2-2, ...1-NRATT, ...2-NRATT, ... NRATT-NRATT
C6[1..NRATT*(NRATT+1)/2]	coefficient of the $-1/r^6$ term in the non-bonded interaction as a function of the occurring pair codes
CS12[1..NRATT*(NRATT+1)/2]	coefficient of the $1/r^{12}$ term in the 1-4 non-bonded interaction between third-neighbour atoms as a function of the occurring pair codes
CS6[1..NRATT*(NRATT+1)/2]	coefficient of the $-1/r^6$ term in the 1-4 non-bonded interaction between third-neighbour atoms as a function of the occurring pair codes

Single atom type van der Waals (Lennard-Jones) parameters block
Blockname: SINGLEATOMLJPAIR

```

WRITE (unit,11) NRATT
DO 10 M=1, NRATT
WRITE (unit,15) I, TYPE[I], C612[I], (C1212[I,K], K=1,3),
WRITE (unit,16) CS612[I], CS1212[I]
10 WRITE (unit,17) (LPAIR[I,K], K=1, NRATT)
15 FORMAT (I5,1X,A5,4E15.7)
16 FORMAT (11X,2E15.7)
17 FORMAT (20I2)

```

If NRATT > 20, the LPAIR values are written with 20 entries on each line using 20I2 as format.

Mixed atom type van der Waals (Lennard-Jones) parameters block
Blockname: MIXEDATOMLJPAIR

```

WRITE (unit,11) NRPAIR
DO 10 M=1, NRPAIR
WRITE (unit,18) I, J, C6[I,J], C12[I,J], CS6[I,J], CS12[I,J]
18 FORMAT (2I5, 4E15.7)

```

The MIXEDATOMLJPAIR block must occur after the SINGLEATOMLJPAIR block on the interaction function parameter file.

GROMOS integer atom codes, single atom type van der Waals parameters for normal and third-neighbour interactions, and mixed atom type van der Waals parameters are given in Vol. 3.

Special atom pair based van der Waals interactions (LJ-exceptions) parameters block
Blockname: LJEXCEPTIONTYPE

```

WRITE (unit,11) NLJEXTYPE
DO 10 M=1, NLJEXTYPE
WRITE (unit,18) M, LJEXC12[M], LJEXC6[M]
18 FORMAT (1I5, 2E15.7)

```

6.8. Atomic charges and charge group codes

The atomic charges and the charge group codes are to be specified with the atoms in the molecular topology building blocks, in the molecular topology building block file. This is discussed in Sec. 5.2.

6.9. Excluded neighbours

The information about which atoms j will be excluded from non-bonded interaction with atom i based on the proximity of atom i and j measured along the covalently bound chain (nearest neighbours), is to be specified with the atomic information in the molecular topology building blocks in the molecular topology building block file. This is discussed in Sec. 5.2.

6.10. Contents of the IFP file

IFP file:

```

TITLE
MASSATOMTYPECODE
BONDSTRETCHTYPECODE
BONDANGLEBENDTYPECODE
IMPDIHEDRALTYPEPEC

```

DIHEDRALTYPECODE
SINGLEATOMLJPAIR
MIXEDATOMLJPAIR

Library files for GROMOS++

7.1. Introduction

The pre- and post-processing programs of GROMOS++ that are described in Chap. 5 make use of different additional library files that are described in the following sections.

7.2. Interaction function parameter renumbering

Several parameter sets of the GROMOS force field are available. Program `con_top` (see Sec. 5-2.6) is able to convert existing topologies to a different parameter set. From parameter set 45A4 to 53A5 all interaction parameter types have been renumbered. To convert topologies that were generated with a parameter set older than 53A5 to the new numbering a renumber file needs to be specified.

The renumbering information is stored as follows:

BTFROM	bond-stretch parameter type in original topology
BTTO	bond-stretch parameter type in resulting topology
ATFROM	bond-angle bend parameter type in original topology
ATTO	bond-angle bend parameter type in resulting topology
IDTFROM	improper (harmonic) dihedral angle parameter type in original topology
IDTTO	improper (harmonic) dihedral angle parameter type in resulting topology
DTFROM	dihedral (trigonometric) angle parameter type in original topology
DTTO	dihedral (trigonometric) angle parameter type in resulting topology
ATOMFROM	Lennard-Jones interaction type code (IAC) in original topology
ATOMTO	Lennard-Jones interaction type code (IAC) in resulting topology

The renumber-file contains the following blocks (apart from the title):

Bondtype conversion block

Blockname: BONDTYPECONV

```
DO 10 N=1, NBT
10 WRITE (unit,11) BTFROM, BTTO
11 FORMAT (2I5)
```

Bond-angle bend conversion block

Blockname: ANGLETYPECONV

```
DO 10 N=1, NAT
10 WRITE (unit,11) ATFROM, ATTO
```

Improper dihedral conversion block

Blockname IMPROPERTYPECONV

```
DO 10 N=1, NIMP
10 WRITE (unit,11) IDTFROM, IDTTO
```

Dihedral angle conversion block

Blockname: DIHEDRALTYPECONV

```
DO 10 N=1, NDIH
10 WRITE (unit,11) DTFROM, DTTO
```

Atomtype conversion block

Blockname: ATOMTYPECONV

```
DO 10 N=1, NATOM
10 WRITE (unit,11) ATOMFROM, ATOMTO
```

An example of a force field renumber file is ren45a4_to_53a5.dat.

7.3. Atomic naming conventions

Program `pdb2g96` can be used to convert molecular coordinate files in `pdb`-format to `GROMOS`-format (see Sec. 5-2.19). This program matches residue and atom names in the `pdb`-file with the names of residues and atoms specified in the molecular topology of the system. For proteins and nucleotides, the names by which residues or nucleotides and atoms are denoted will correspond exactly in the two files. However, some often occurring differences are known. These can be defined in the `pdb2g96`-library file which is to be specified when using the `pdb2g96` program.

The library file is defined by the following variables:

RESPDB	The name of a residue that is encountered in a <code>pdb</code> -file
RESTOPO	The name by which the corresponding residue is denoted in the topology RESAT The name of a residue for which an atom name difference is listed, according to the topology.
ATMPDB	The name of an atom in residue RESAT as it may be encountered in a <code>pdb</code> -file
ATMTOPO	The name of the corresponding atom in residue RESAT will be denoted in the topology

Apart from the title block, the `pdb2g96` library file contains the following blocks:

Residue name block

Blockname: RESIDUENAMELIB

```
DO 10 N=1, NRES
10 WRITE (unit,11) RESPDB, RESTOPO
11 FORMAT (3A6)
```

Atomic name block

Blockname: ATOMNAMELIB

```

DO 10 N=1, NATOM
10 WRITE (unit,11) RESAT, ATPDB, ATMTOPO

```

An example of this library file is `pdb2g96.lib`.

7.4. Definition of file-names and joblists

Program `mk_script` can generate jobscripts and input files to run MD++ (see Sec. 5-2.18). Although there are recommended file-names for the different GROMOS files that are used in a molecular simulation, there is no requirement to use these names. Program `mk_script` can generate names for files according to user-defined rules, that use the simulation time or a simulation sequence number. The rules to define these files are given in a `mk_script` template file. Some additional string constants to be used in the scripts can also be defined in the template file. The following types are recognized:

<code>script</code>	A rule to define the scriptname
<code>qinput</code>	A rule to define the name of a MD++ input file
<code>output</code>	A rule to define the name of a MD++ output file
<code>coord</code>	A rule to define the name of a single structure coordinate file
<code>pttopo</code>	A rule to define the name of a perturbation topology
<code>refpos</code>	A rule to define the name of a reference position coordinate file
<code>posresspec</code>	A rule to define the name of a position restraints specification file
<code>disres</code>	A rule to define the name of an atom-atom distance restraint file
<code>dihres</code>	A rule to define the name of a dihedral angle restraining file
<code>jvalue</code>	A rule to define the name of a 3J -value restraints specification file
<code>ledih</code>	A rule to define the name of a local-elevation specification file
<code>outtrx</code>	A rule to define the name of a molecular coordinate trajectory file
<code>outtrv</code>	A rule to define the name of a molecular velocity trajectory file
<code>outtre</code>	A rule to define the name of an energy trajectory file
<code>outbae</code>	A rule to define the name of a block-averaged energy trajectory file
<code>outtrg</code>	A rule to define the name of a free energy trajectory file
<code>outbag</code>	A rule to define the name of a block-averaged free energy trajectory file
<code>workdir</code>	A rule to define a directory where the simulation can be run locally
<code>mpicommand</code>	A rule to define the command used to run in an MPI parallel environment
<code>firstcommand</code>	A rule to define an initial command that needs to be performed before the call to MD++
<code>lastcommand</code>	A rule to define a final command that is to be called when the simulation script finishes

The variables to store the rules are as follows:

`FILETYPE` The type of a file for which the rule will be specified.

FILENAME The rule to form the corresponding file name

MISCTYPE The type of another stringconstant that needs to be formed based on the simulation time or sequence number

MISCNAME The rule to form the corresponding stringconstant

The template file is built up with the following blocks (apart from the title block):

Filename specification block

Blockname: FILENAMES

```

DO 10  N=1, NSPEC
10  WRITE  (unit,11) FILETYPE, FILENAME
11  FORMAT (A20, A60)

```

Miscellaneous specification block

Blockname: MISCELLANEOUS

```

DO 10  N=1, NSPEC
10  WRITE  (unit,11) MISCTYPE, MISCNAME

```

An example of a template file is `mkscript.lib`.

Program `mk_script` can not only write a single script with the appropriate naming conventions for the files that are involved, but it can also generate a consistent set of simulations that perform a specific task. This is done by specifying a joblist in which specific variable of the input file can be given different values from simulation script to simulation script.

A joblist is specified by the following variables:

NVAR The number of variables that are to be modified between scripts

NSCRIPTS The number of scripts and input files that will be written

VARID[1..NVAR+3] An identifying string for every variable. The value of VARID[1] is required to be “job_id”, the value of VARID[NVAR+2] to be “subdir” and the value of VARID[NVAR+3] to be “run_after”.

VARVAL[1..NVAR+3,1..NSCRIPTS] The value of the specified variables in the input files that are to be generated. VARVAL[1,1..NSCRIPTS] contains the job-sequence number by which the scripts can be identified. VARVAL[NVAR+2,1..NSCRIPTS] contains a string constant that refers to the subdirectory where the simulations will be run and VARVAL[NVAR+3,1..NSCRIPTS] specifies which script-id should end with a call (or submission) of this script (defined through variable `lastcommand` in the template file).

Joblist specification file

Blockname: JOBSSCRIPTS

```

WRITE  (unit, 11)(VARID[N], N=1, NVAR)
DO 10  M=1, NSCRIPTS
10  WRITE  (unit, 11)VARVAL[M,N], N=1, NVAR)
11  FORMAT (20A10)

```

Examples for joblist specification files are `joblist.startup` and `joblist.perturbation`.

7.5. Energy trajectory block definition

Energy and free energy data is written at a user specified interval to (free) energy trajectory files. MD++ can also write block averaged trajectory files. Program `ene_ana` can be used to extract time series of properties derived from variables stored in these trajectory files (see Sec. 5-4.21). `ene_ana` has been written such that it can read any block-based trajectory file. The block format is specified in a library file that can be modified by the user. In addition, this library file contains definitions to calculate properties from the values that are stored in the trajectory files.

The format of the library file depends on the format of the free energy trajectory it is defining. It contains three blocks, ENERTRJ, FRENERTJ and VARIABLES. The blocks ENERTRJ and FRENERTJ define the format of the energy trajectory and the free energy trajectory respectively. Every line in this block contains one entry, with a first keyword specifying what kind of entry this is. The following keywords are recognized:

block	followed by a block name. For every configuration that is written to the trajectory file, program <code>ene_ana</code> will try to read these blocks. The entries in the library file that follow on subsequent lines specify the expected format of the block.
size	followed by a variable name. This entry tells <code>ene_ana</code> that it should read an integer number and store this in the specified name. This number can subsequently be used in the definition of arraysizes
subblock	followed by a variable name and two dimensions. This entry tells <code>ene_ana</code> to read a block of data of the specified dimensions and store the data under the specified name. The dimensions can be either specified by an integer number, by a previously defined size-variable or by such a variable preceded by the word "matrix_". For example, if the size-variable NEGR was defined previously, the dimension specification "matrix_NEGR" will expand to the value $\text{NEGR} * (\text{NEGR} + 1) / 2$.

In the VARIABLES block properties can be specified based on the data that was read in from the trajectory files. A new property is defined if the second word on a line consists of the character '='. All string constants read in until the next definition of a property will be considered to be part of the same definition. The raw data read in from the energy trajectory files are referred to by the name of the subblocks followed by rectangular brackets to indicate the individual elements of the arrays. If the second dimension of a subblock is one, the second set of brackets may be omitted. Properties can be defined using subblock names, the characters +, -, *, /, (,) and any properties that were previously defined.

An examples of an energy trajectory specification file is `ene_ana.md++.lib`

7.6. Hydrogen-bond donors and acceptors

Program `hbond` monitors the presence of hydrogen bonds throughout a simulation (see Sec. 5-4.32). The explicit hydrogen atoms and H-bond acceptor atoms to be monitored can be specified individually, or can be obtained by applying a mass-filter on a larger set of specified atoms. The definition of a mass representing a hydrogen atom and possibly H-bond acceptors is done through a massfile, which is defined by the following variables

NHMASS	Number of masses that represent hydrogen atoms
HMASS[1..NHMASS]	Mass representative for hydrogen atoms

NACCMASS Number of masses that represent H-bond acceptor atoms

ACCMASS[1..NACCMASS]
 Mass representative for H-bond acceptor atoms

This information is stored in the following blocks

Hydrogenmass block

Blockname: HYDROGENMASS

```
      DO 10  N=1, NHMASS
10  WRITE  (unit,11) HMASS[N]
11  FORMAT (F15.7)
```

Acceptormass block

Blockname: ACCEPTORMASS

```
      DO 10  N=1, NACCMASS
10  WRITE  (unit,11) ACCMASS[N]
```

An example of a massfile is hbond.massfile.

7.7. Crystallographic transformations

Program cry can construct crystallographic unit cells by applying the appropriate symmetry transformations on a given molecular structure (see Sec. 5-2.8). The symmetry transformations are specified by a rotation matrix, M, and a translation vector, V in a specification file. The following variables are required

NSOP Number of symmetry transformations that are defined

M[1..3,1..3,1..NSOP]
 Rotation matrix M for every transformation

V[1..3, 1..NSOP] Translation vector V for every transformation

These variables are stored in the following block:

Symmetry transformation block

Blockname: TRANSFORM

```
      WRITE  (unit,20) NSOP
      DO 10  I=1, NSOP
10  DO 11  J=1,3
11  WRITE  (unit,21) M[1,J,I], M[2,J,I], M[3,J,I], V[J,I]
20  FORMAT (I5)
21  FORMAT (3F11.5,4X,F11.5)
```

An example of a transformation file is cry.spec.

7.8. NOE analysis

The programs `prep_noe` (Sec. 5-4.45), `noe` (Sec. 5-4.41) and `post_noe` (Sec. 5-4.42) analyse a trajectory for atom-atom distances and compare to experimentally determined upper-bounds to such distances. The NOE's are specified using virtual and pseudo-atoms as described in section Sec. 3.4. Program `prep_noe` can generate this NOE specification file from a list of proton-proton distances and a library file. Corrections to the experimentally determined upper-bounds for pseudo-atoms and multiplicities are defined in a correction file.

The proton-proton distances can be specified using a XPLOR like NOE specification file, which can be easily generated from e.g. an XPLOR-format. This format usually uses three distances, from which the upper- and lower-bounds for the atom-atom distances can be calculated. It uses the following variables:

NNOE	Number of NOE distances specified
SEQN	Sequential NOE number starting from 1 to NNOE
RESI[1..NNOE]	Residue number of atom I of the NOE distance
NAMEI[1..NNOE]	Atom name of atom I of the NOE distance
RESJ[1..NNOE]	Residue number of atom J of the NOE distance
NAMEJ[1..NNOE]	Atom name of atom J of the NOE distance
ANOE[1..NNOE]	XPLOR distance 1
BNOE[1..NNOE]	XPLOR distance 2
CNOE[1..NNOE]	XPLOR distance 3
NUMAMB[1..NNOE]	Number of ambiguous NOEs to which this NOE is linked
AMBNOE[1..NOE,1..NUMAMB]	SEQN of NOEs to which this NOE is linked

XPLOR like NOE specification block

Blockname: NOESPEC

```
DO 10 N=1, NNOE
10  WRITE (unit,11) SEQN[N], RESI[N], NAMEI[N], RESJ[N], NAMEJ[N],
      ANOE[N], BNOE[N], CNOE[N], SEQN[N], NUMAMB[N],
      AMBNOE[N,1], AMBNOE[N,2], ...
11  FORMAT (2I5,A5,I5,A5,3F8.3,I4,I4,10I4)
```

An example of a XPLOR like NOE specification file can be found in `examples/prep.noe`.

For unambiguous NOEs, only the first eight columns of this file are to be specified. For ambiguous restraints, the 9th column repeats the number of the NOE (first column), the 10th column contains the number of NOEs this NOE may be linked to and the remaining columns lists the numbers of the NOEs to which it is linked.

An NOE library file determines what type of virtual or pseudo-atom needs to be used to represent the proton-proton distances. The NOE library is defined by the following variables:

NNLIB	Number of entries in the library file
RSNM[1..NNLIB]	Residue name of for the atom

PNMIU[1..NNLIB] IUPAC name of the proton (replace * with @)
 CANM[1..NNLIB] Name of the central atom of a virtual or pseudo atom description.
 PATP[1..NNLIB] Explicit, virtual or pseudo-atom type as described in section Sec. 3.4.

NOE library block

Blockname: NOELIB

```

DO 10 N=1, NNLIB
10 WRITE (unit,11) RSNM[N], PNMIU[N], CANM[N], PATP[N]
11 FORMAT (3A6,I5)

```

Examples of NOE library files are noelib.45a3 and noelib.53a6.

Virtual and pseudo atoms may require corrections to the upper bounds due to the position of the atom or the multiplicity of the signal. Program `prep_noe` can either add or remove such correction from a given set of distances. The corrections are defined in a NOE correction file, which contains the following variables:

NPAC Number of pseudo-atom corrections in the file
 NSPAC NOE suptype to which the pseudo-atom correction applies (set to 0 if no subtype defined)
 NTPAC[1..NPAC] NOE type to which the pseudo-atom correction applies
 FTPAC[1..NPAC] Distance of the pseudo-atom correction
 NMPC Number of multiplicity corrections in the file
 NTMPC[1..NPAC] NOE type to which the multiplicity correction applies
 FTMPC[1..NPAC] Factor for the multiplicity correction

This information is written in the following blocks:

Pseudo atom correction block

Blockname: NOECORGROMOS

```

DO 10 N=1, NPAC
10 WRITE (unit, 11) NTPAC[N], NSPAC[N], FTPAC[N]
11 FORMAT (I5, F15.8)

```

Multiplicity correction block

Blockname: MULTIPLICITY

```

DO 10 N=1, NMPC
10 WRITE (unit,11) NTMPC[N], NSPAC[N], FTMPC[N]

```

Examples of NOE correction files are called noecor.*.

The program `prep_noe` generates the NOE specification file which can be used as input for programs `noe` and `post_noe`. A NOE distance in the NOE specification file is characterised by the following quantities:

DISH carbon-hydrogen distance

DISC	carbon-carbon distance
NNOE	Number of NOE distances specified
IDR1, JDR1, KDR1, LDR1 [1..NNOE]	atom sequence numbers of the real atoms defining the geometric position of the first atom of a NOE distance pair
ICDR1 [1..NNOE]	geometric code defining the position of the first atom of a distance restraint pair [-2, -1, ..., 7] (see Sec. 3.4)
VACS1 [1..NNOE]	subtype of first virtual atom. Possible subtypes are: 0: no subtype defined (for ICDR1 = -2, 0-7) 1: aromatic flipping ring (for ICDR1 = -1) 2: non-stereospecific NH2 group (for ICDR1 = -1)
IDR2, JDR2, KDR2, LDR2 [1..NNOE]	atom sequence numbers of the real atoms defining the geometric position of the second atom of a NOE distance pair
ICDR2 [1..NNOE]	geometric code defining the position of the second atom of a distance restraint pair [-2, -1, ..., 7] (see Sec. 3.4)
VACS2 [1..NNOE]	subtype of second virtual atom. Possible subtypes are: 0: no subtype defined (for ICDR2 = -2, 0-7) 1: aromatic flipping ring (for ICDR2 = -1) 2: non-stereospecific NH2 group (for ICDR2 = -1)
R0 [1..NNOE]	corrected upper bound for NOE distance.

NOE specification block

Blockname: NOECALCSPEC

```

WRITE (unit,11) DISH, DISC
DO 10 N=1, NNOE
10 WRITE (unit,12) IDR1[N], JDR1[N], KDR1[N], LDR1[N], ICDR1[N], VACS1[N],
    IDR2[N], JDR2[N], KDR2[N], LDR2[N], ICDR2[N], VACS2[N],
    R0[N]
11 FORMAT (2F10.5)
12 FORMAT (12I5,1F10.5)

```

7.9. SASA implicit solvent model

Program `make_sasa_top` adds the atom-specific information required to use the SASA implicit solvent model to the molecular topology file (see Sec. 5-2.16). It reads in an existing molecular topology file created using `make_top`, along with a SASA specification library file, which contains the atom-specific SASA parameters. The specification library file must be for the same force field as was used to create the molecular topology file. The inclusion of hydrogen atoms in the calculation of the SASA during the simulation can also be specified. The following variables are known

NRSASAT Number of atom types with a unique set of SASA parameters (not given in file).

RADI[1..NRSASAT] Atomic radius for each SASA atom type.

PI[1..NRSASAT] Atom type-specific parameter for reduction in SASA.

SIGMAI[1..NRSASAT] Scaling parameter for SASA energy term ($\text{kJ}\cdot\text{mol}^{-1}\cdot\text{nm}^{-2}$)

NRIACI[1..NRSASAT] Number of integer atomic codes corresponding to this SASA atom type.

IAC[1..NRSASAT,1..NRIACI] Integer atomic code for each atom corresponding to this SASA atom type.

These variables are stored in the following block of the library file:

SASA parameter specification block

Blockname: SASASPEC

```

DO 10 I=1, NSASAT
10  WRITE (unit,20) RADI[I], PI[I], SIGMAI[I], NRIACI[I], (IAC[I,J], J=1, NRIACI)
20  FORMAT (F5.3,3X,F5.3,4X,5I,3X,2I,3X,5I3)

```

The 5I3 in format statement 20 is for $\text{NIAC} \leq 5$; this should be altered if $\text{NIAC} > 5$. Examples of a SASA specification library file can be found in `data/sasa45b3.spec` and `data/sasa53a6.spec`. The values of RADI and PI in these files were optimised by Hasel et al.¹ for $\text{RSOLV} = 0.14 \text{ nm}$. They should not be changed without justification. Different values of SIGMAI are required if the SASA implicit solvent model is used alone (see²) or with the VOLUME correction (see³). The IAC values will depend on the force field that is used. They are listed in Vol. 3.

7.10. DISICL angle, region and segment definitions

Program `disicl` classifies protein and nucleic acid secondary structure based on dihedral angles (see Sec. 5-4.11.^{4,5} Angle, region and segment definitions are read in from a user-specified library file.

The library file is defined by the following variables:

DIHNAME Name of the dihedral to define.

ATOM[1..4] Atom names defining a dihedral angle, either simply by the name or by an expression in the following format: D;RES1,RES2:B;RES3:C, where RES1..3 are residue names for which atom B, B and C, respectively will be used, whereas atom D is the default which will be used for all other residues. The most common case will be different atoms for purines and pyrimidines in nucleic acids, e.g. N1;GUA,ADE:N9.

SHIFT[1..4] Relative residue number for each atom.

REGNAME The name of a DISICL region.

REGMIN[1..NDIH] The lower limit of the region.

REGMAX[1..NDIH] The upper limit of the region.

CLASSNAME	The name of a DISICL segment.
SEGDEF1	The region the current residue has to fall into.
SEGDEF2	The region the following residue has to fall into.
CLASSSHORT	Shortname of the class.

Apart from the title block, the disicl library file contains the following blocks:

Dihedral angle definition block

Blockname: DSCLANG

```

DO 10 N=1, NDIH
10 WRITE (unit,11) DIHNAME, (ATOM[N], N=1, 4), (SHIFT[N], N=1, 4)
11 FORMAT (A6, 4A4, 4I4)

```

Region definition block

Blockname: DSCLREG

```

DO 10 N=1, NREG
10 WRITE (unit,11) REGNAME ((REGMIN[M], REGMAX[M]), M=1, NDIH)
11 FORMAT (A8, 16F3.1)

```

Class definition block

Blockname: DSCLCLASS

```

DO 10 N=1, NSEG
10 WRITE (unit,11) CLASSNAME, SEGDEF1, SEGDEF2, CLASSSHORT
11 FORMAT (A20, 2A8, A6)

```

An example of this library file is DISICL_prot_detailed.lib.

Input file for MD++

The data structure of the *input file* (input flag @input for MD++) is as follows:

MD++ QUICK REFERENCE SHEET

- Blocks can appear in any order
- Compulsory blocks are marked by a star
- When an optional block is not given all switches will be set to their "DEFAULT" value.
- Linebreaks in the variable list should match linebreaks in the input file
- Error checking is performed in three phases:
 - Phase I: unknown, duplicate or missing compulsory blocks, switches defining array lengths
 - Phase II: incorrect switch values or variable ranges within the blocks
 - Phase III: incompatible switches within and among the blocks
- EM, MD, SD, or RT denote the energy minimisation, molecular dynamics, stochastic dynamics and trajectory reading modes of the program
- NRP(> 0): number of atoms of the solute
- NRAM(> 0): number of atoms per solvent molecule
- NATTOT=NRP+NSM*NRAM: total number of atoms in the system

TITLE	page
TITLE*	4-88
MOLECULAR SYSTEM	
SYSTEM*	4-105
METHOD EMPLOYED	
(default is to do an MD run, when none of the first four blocks below are present or when the corresponding first switch is set to zero)	
ENERGYMIN	4-93
STOCHDYN	4-104
READTRAJ	4-103
REPLICA	4-103
STEP*	4-104
SPATIAL BOUNDARY CONDITIONS	
BOUNDCOND*	4-89
MULTICELL	4-97
THERMODYNAMIC BOUNDARY CONDITIONS	
MULTIBATH	4-96
PRESSURESCALE	4-101
MULTIGRAIENT	4-97
INTERACTION EVALUATION	
FORCE*	4-93
COVALENTFORM	4-90
CONSTRAINT*	4-90
POLARISE	4-101
INTEGRATE	4-95
CGRAIN	4-89
ROTTRANS	4-103
INNERLOOP	4-95
MULTISTEP	4-98

PAIRLIST*	4-99
NONBONDED*	4-98
INITIALISATION OF THE RUN	
INITIALISE	4-94
RANDOMNUMBERS	4-102
CENTRE-OF-MASS MOTION	
COMTRANSROT	4-89
SPECIAL FORCES	
POSITIONRES	4-101
DISTANCERES	4-91
DIHEDRALRES	4-91
JVALUERES	4-95
ORDERPARAMRES	4-99
DISTANCEFIELD	4-91
QMMM	4-102
LOCALELEV	4-96
PERSCALE	4-100
ELECTRIC	4-92
SASA	4-104
FREE-ENERGY CALCULATION	
PERTURBATION	4-100
LAMBDA	4-96
PRECALCLAM	4-101
EDS	4-92
AEDS	4-88
INPUT-OUTPUT	
PRINTOUT	4-102
WRITETRAJ	4-105
EWARN	4-93

TITLE*

text

- Arbitrary text that can be used to identify the simulation.

AEDS

AEDS

ALPHLJ,ALPHC,FORM,NUMSTATES

EMAX,EMIN

EIR(1..NUMSTATES)

NTIAEDSS,RESTREMIN,BMAXTYPE,BMAX,ASTEPS,BSTEPS

AEDS 0,1 controls accelerated enveloping distribution sampling (A-EDS)

0: no accelerated enveloping distribution sampling (EDS) [DEFAULT]

1: accelerated enveloping distribution sampling

ALPHLJ ≥ 0.0 Lennard-Jones soft-core parameter

ALPHC ≥ 0.0 Coulomb soft-core parameter

FORM 1..4 defines type of A-EDS simulation

1: A-EDS with fixed parameters

2: fixed Emax and Emin parameters, search for offset parameters

3: search for Emax and Emin parameters, fixed offset parameters

4: search for Emax, Emin and offset parameters

NUMSTATES ≥ 2 number of (end)states

EMAX A-EDS parameter Emax

EMIN A-EDS parameter Emin

EIR energy offsets for states

NTIAEDSS 0..1 controls startup of the A-EDS parameter search
 0: read A-EDS parameter search configuration from input configuration
 1: initialize A-EDS parameter search

RESTREMIN 0..1 controls restriction of parameter Emin during parameter search
 0: do not restrict $E_{min} \geq$ minimum average end-state energy
 1: restrict $E_{min} \geq$ minimum average end-state energy before all states have been visited at least once

BMAXTYPE 1..2 controls type of given anticipated maximum energy barrier between the states
 1: absolute maximum energy barrier between the states in energy units
 2: multiples of the standard deviation of the energy of the end-state with the lowest average energy

BMAX maximum energy barrier parameter

ASTEPS have-life in simulation steps of the exponential averaged energy difference between the end-states at the beginning of the run

BSTEPS have-life in simulation steps of the exponential averaged energy difference between the end-states at the end of the run

- The parameter E_{max} must be $\geq E_{min}$
- NBATHS= 0 results in an error, in addition all baths must have the same temperature TEMPO
- A-EDS cannot be applied with replica exchange
- A-EDS cannot be applied to solvent atoms

BOUNDCOND*

NTB,NDFMIN

NTB -1..2 controls type of boundary conditions
 -1: truncated-octahedral periodic boundary conditions
 0: vacuum boundary conditions
 1: rectangular periodic boundary conditions
 2: triclinic periodic boundary conditions

NDFMIN ≥ 0 number of degrees of freedom subtracted for temperature

- $NTM \neq 0$ requires NTB= 1,2
- pressure coupling requires NTB $\neq 0$
- (semi-)anisotropic pressure coupling (SCALE=2,4) requires NTB=1 or 2
- full anisotropic pressure coupling (SCALE=3) requires NTB=2
- $abs(NLRELE) > 1$ requires NTB $\neq 0, -1$
- NTISHI=0 requires NTB $\neq 0$
- $NTRD \neq 0$ and $NTRB \neq 0$ require NTB $\neq 0$
- Initial box parameters (GENBOX) are read from @conf

CGRAIN

NTCGRAN,EPS,EPSM

NTCGRAN = 0..3 Coarse grain selection
 0: No coarse graining [DEFAULT]
 1: Coarse grain simulation using the MARTINI model
 2: Coarse grain simulation using the GROMOS model
 3: Mixed-grain simulation using the GROMOS model

EPS ≥ 0.0 Dielectric constant for coarse grained - coarse grained coulombic interactions

EPSM ≥ 0.0 Dielectric constant for coarse grained - fine grained coulombic interactions

COMTRANSROT

NSCM

NSCM controls system center-of-mass (com) motion removal
 0: no com motion removal [DEFAULT]
 < 0: com translation and rotation are removed every $abs(NSCM)$ steps
 > 0: com translation is removed every NSCM steps

- NSCM≠0 should not be used with roto-translational constraints (RTC=1)

CONSTRAINT*

NTC

NTCP,NTCP0(1),[NTCP0(2),NTCP0(3)]

NTCS,[NTCS0(1),NTCS0(2),NTCS0(3)]

NTC 0,1,2,3,4 controls application of constraints to bonds

1: constraints are applied to solvent only

2: constraints are applied to solvent and solute bonds involving hydrogen atoms and to bonds specified in the topology CONSTRAINT block

3: constraints are applied to solvent and solute bonds

4: constraints are applied to bonds specified in the CONSTRAINT block in the topology and to solvent

NTCP shake, lincs, flexshake controls algorithms to apply solute constraints

shake(1) apply shake for solute

lincs(2) apply lincs for solute

flexshake(3) apply flexible shake for solute

NTCP0(1) > 0 option parameters for constraint algorithm Shake: Tolerance,
Lincs: Order

NTCP0(2..3) ≥ 0 option parameters for flexible shake algorithm: readin, mode
[only supply when flexible shake is selected]

NTCS shake, lincs, flexshake, settle, m_shake, gpu_shake controls algorithm to apply solvent constraints

shake(1) apply shake for solvent

lincs(2) apply lincs for solvent

flexshake(3) apply flexible shake for solvent

settle(4) apply settle for solvent

m_shake(5) apply m_shake for solvent

gpu_shake(6) apply m_shake for solvent using GPU

NTCS0(1) ≥ 0 option parameter for constraint algorithm: (flexible) Shake or M-Shake:
Tolerance; Lincs: order; Settle: do not specify

NTCS0 (2..3) ≥ 0 option parameters for flexible shake algorithm: readin, mode
[only supply when flexible shake is selected]

NTCG ≥ 0 number of GPUs

[only supply when GPU shake is selected]

NTCD ≥ -1 device number of the GPU; if -1 given driver will determine
[only supply when GPU shake is selected]

COVALENTFORM

NTBBH,NTBAH,NTBDN

NTBBH 0,1 controls bond-stretching potential energy function

0: quartic potential energy function [DEFAULT]

1: harmonic potential energy function

NTBAH 0,1 controls bond-angle bending potential energy function

0: cosine-harmonic potential energy function [DEFAULT]

1: harmonic potential energy function

NTBDN 0,1 controls torsional dihedral potential energy function

0: arbitrary phase shifts [DEFAULT]

1: phase shifts limited to 0 and 180 degrees

- A topology containing bond types only in the form of a BONDANGLETYPE block and no BONDANGLEBENDTYPE block requires NTBAH= 0, the HARBONDANGLETYPE block requires NTBAH= 1.

- A topology containing a DIHEDRALTYPE and no TORSDIHEDRALTYPE block requires NTBDN=1
- NTBDN=1 along with the presence of a topology block TORSDIHEDRALTYPE requires that all phase shifts are 0 or 180 degrees in this block

DIHEDRALRES

NTDLR,CDLR,PHILIN

NTDLR 0...3 controls dihedral-angle restraining or constraining
 0: no dihedral restraining [DEFAULT]
 1: dihedral restraining using CDLR (WDLR ignored)
 2: dihedral restraining using CDLR×WDLR
 3: dihedral constraining
 CDLR ≥0.0 force constant for dihedral restraining (multiplied by WDLR)
 PHILIN 0...180 absolute deviation (degrees) after which the potential energy function is linearised.
 If zero no linearisation performed.

- Dihedral restraints and weights WDLR in DIHEDRALRESSPEC and PERTDIHRESSPEC block read from @dihrest

DISTANCEFIELD

NTDFR,GRID,PROTEINOFFSET,PROTEINCUTOFF,PROTECT,UPDATE,SMOOTH,RL,NTWDF,PRINT-GRID

NTDFR 0,1 controls distance-field restraining
 0: no distance-field restraining [DEFAULT]
 1: apply distance-field restraining
 GRID > 0.0 grid size for distance-field
 PROTEINOFFSET > 0.0 penalty for distances through the host
 PROTEINCUTOFF > 0.0 distance to host atoms to be considered inside
 PROTECT ≥ 0 protect grid points within this radius around the zero-distance point from being flagged as protein
 UPDATE > 0 update frequency for grid
 RL ≥ 0.0 potential energy function for distances larger than RL
 SMOOTH ≥ 0 smoothen the host boundary after grid construction by SMOOTH layers
 NTWDF ≥ 0 write distance-field information to special trajectory every NTWDF steps
 PRINTGRID 0,1 write grid to final configuration file

- Distance-field specification read from distance restraints specification file (@distrest)
- To use distance-field coordinate in local elevation, turn off the restraining potential energy function (NTDFR = 0)
- Distance-field restraints require NTB=1.

DISTANCERES

NTDIR,NTDIRA,CDIR,DIR0,TAUDIR,FORCESCALE,VDIR,NTWDIR

NTDIR -2..3 controls distance restraining
 -2: time-averaged restraining using force constant CDIR×W0
 -1: time-averaged restraining using force constant CDIR (W0 ignored)
 0: no distance restraining [DEFAULT]
 1: instantaneous restraining using force constant CDIR (W0 ignored)
 2: instantaneous restraining using force constant CDIR×W0
 NTDIRA 0,1 controls values of initial distance averages
 0: zero initial averages [DEFAULT]
 1: read current averages from startup file
 CDIR ≥ 0.0 force constant for distance restraining

DIR0 ≥ 0.0 distance offset in restraining function
 TAUDIR > 0.0 coupling time for time averaging
 FORCESCALE 0..2 controls approximation of force scaling
 0: approximate $d\langle r \rangle / dr = 1$
 1: approximate $d\langle r \rangle / dr = (1.0 - \exp(-\Delta t / \tau))$
 2: use $d\langle r \rangle / dr = (1.0 - \exp(-\Delta t / \tau)) * (\langle r \rangle / r)^4$
 VDIR 0,1 controls contribution to virial
 0: no contribution
 1: distance restraints contribute to virial
 NTWDIR ≥ 0 write every NTWDIRth step distance restraining information to external file

- NTDIRA= 1 requires NTWDIR ≤ 0
- List of distance restraints and weights W0 (DISTANCERESSPEC) read from @distrest (24)
- Average distances (DISRESEXPAVE) read from @conf if NTDIRA= 1
- NTWDIR > 0 requires the specification of a special trajectory file with @trs

EDS

EDS,ALPHLJ,ALPHC,FORM,NUMSTATES,S,EIR

EDS 0,1 controls enveloping distribution sampling
 0: no enveloping distribution sampling (EDS) [DEFAULT]
 1: enveloping distribution sampling
 ALPHLJ ≥ 0.0 Lennard-Jones soft-core parameter
 ALPHC ≥ 0.0 Coulomb soft-core parameter
 FORM 1..3 defines functional form of the Hamiltonian
 1: Single s Hamiltonian
 2: Hamiltonian with NUMSTATES*(NUMSTATES-1)/2 (pairwise) s parameters
 3: Hamiltonian with (NUMSTATES-1) s parameters
 NUMSTATES ≥ 2 number of (end)states
 S > 0.0 smoothness parameter(s) (number according to functional form)
 EIR ≥ 0.0 energy offsets for states

- FORM=3 requires the specification of a tree: "S" becomes "i j S", where i and j are the pair of states for which the S is applied
- NBATHS= 0 results in an error, in addition all baths must have the same temperature TEMPO
- EDS cannot be applied with replica exchange
- EDS cannot be applied to solvent atoms

ELECTRIC

FIELD,DIPOLE,CURRENT

EF_x,EF_y,EF_z

DIPGRP,NTWDIP

NTWCUR,NCURGRP,CURGRP(1..NCURGRP)

FIELD 0..1 controls the use of applied electric field
 0: not used [DEFAULT]
 1: electric field is applied
 DIPOLE 0..1 controls the calculation of the box dipole
 0: not used [DEFAULT]
 1: box dipole is calculated and written to special trajectory
 CURRENT 0..1 controls the calculation of electric (ionic) currents
 0: not used [DEFAULT]
 1: electric (ionic) current is calculated and written to special trajectory
 EF_x *double* x-component of the electric-field vector
 EF_y *double* y-component of the electric-field vector
 EF_z *double* z-component of the electric-field vector
 DIPGRP 0..2 define the groups for which the box dipole is calculated

- 0: solute only
- 1: solvent only
- 2: all

NTWDIP ≥ 0 write box dipole to special trajectory every NTWDIPth step
 NTWCUR ≥ 0 write box currents to special trajectory every NTWCURth step
 NCURGRP ≥ 0 number of current groups
 CURGRP(1..NCURGRP) ≥ 0 last atom of each current group

ENERGYMIN

NTEM,NCYC,DELE,DX0,DXM
 NMIN,FLIM,CGIC,CGIM

NTEM 0..3 controls energy minimisation mode
 0: do not do energy minimisation [DEFAULT]
 1: use steepest-descent minimisation
 2: use Fletcher-Reeves conjugate gradient minimisation
 3: use Polak-Ribiere conjugate gradient minimisation
 NCYC > 0 number of steps before resetting the conjugate-gradient search direction
 = 0 reset only if the energy grows in the search direction
 DELE > 0.0 energy threshold for convergence
 > 0.0 (conjugate-gradient) RMS force threshold for convergence
 DX0 > 0.0 initial step size
 DXM > 0.0 maximum step size
 NMIN > 0 minimum number of minimisation steps
 FLIM ≥ 0.0 limit force to maximum value
 CGIM > 0 (conjugate-gradient only) maximum number of cubic interpolations per step
 CGIC > 0.0 (conjugate-gradient only) RMSD threshold after interpolation

- $DX0 \leq DXM$
- $NTSD \neq 0$ requires $NTEM = 0$
- $NTRD \neq 0$ requires $NTEM = 0$
- pressure or temperature coupling is not allowed with $NTEM = 0$
- $NSCM \neq 0$ (center-of-mass motion removal) requires $NTEM = 0$

EWARN

MAXENER

MAXENER Issues a warning if the total energy is larger than this value

FORCE*

NTF(1..6)
 NEGR
 NRE(1..NEGR)

NTF(1..6) 0,1 determines terms used in force calculation
 NTF(I)= 0 do not include terms of type I
 NTF(I)= 1 include terms of type I
 NTF (1) bonds
 NTF (2) bond angles
 NTF (3) improper dihedrals
 NTF (4) dihedrals
 NTF (5) nonbonded electrostatic interactions
 NTF (6) nonbonded van der Waals interactions
 NEGR ≥ 0 number of energy groups
 0: no energy groups

- initial shift vectors (LATTICESHIFTS) read from @conf if NTB≠ 0 and NTISHI= 0
- initial positions and orientations (ROTTRANSREFPOS) read from @conf if RTC=1 and NTIRTC= 0
- stochastic integrals and seed (STOCHINT) read from @conf if NTSD≠ 0 and NTISTI= 0
- If NTIVEL= 0 and NTISTI= 0, IG is irrelevant
- If NTIVEL= 0, TEMPI is irrelevant
- NTIVEL≠ 0 results in a warning if VELOCITY block is found in @conf
- NTISTI≠ 0 results in a warning if STOCHINT block is found in @conf
- NTIRTC≠ 0 results in a warning if ROTTRANSREF block is found in @conf

INNERLOOP

NTILM 0..4, acceleration method used

- 0: use standard solvent loops [DEFAULT]
- 1: use fast generic solvent loops
- 2: use solvent loops with hardcoded parameters
- 3: use solvent loops with tabulated forces and energies
- 4: use solvent loops with CUDA library

NTILS 0..1, solvent used

- 0: use topology [DEFAULT]
- 1: use SPC

NGPUS number of GPUs to use

NDEVG which GPU device number to use; if not given driver will determine

INTEGRATE

NINT

NINT = 0, 1 selects integration method

- 0: No integration takes place
- 1: Leap-frog integration scheme is used [DEFAULT]

JVALUERES

NTJVR,NTJVRA,CJVR,TAUJVR,NJVRTARS,
NJVRBIQW,LE,NGRID,DELTA,NTWJV

NTJVR -3...2

- 3: biquadratic using CJVR×WJVR
- 2: time-averaged using CJVR×WJVR
- 1: time-averaged using CJVR (WJVR ignored)
- 0: no ³J-value restraining [DEFAULT]
- 1: instantaneous using CJVR (WJVR ignored)
- 2: instantaneous using CJVR×WJVR

NTJVRA 0,1 controls reading of averages from startup file

- 0: start from initial values of J0 [DEFAULT]
- 1: read time-averages from startup file (for continuation of time-averaged run)

CJVR ≥0 ³J-value restraining force constant (weighted by individual WJVR)

TAUJVR >0 coupling time for time-averaging

NJVRTARS 0,1 controls scaling of force in time-averaging

- 0: omit factor $[1 - \exp(\Delta t/\tau_{Jr})]$, i.e. set it to one
- 1: scale force by $[1 - \exp(\Delta t/\tau_{Jr})]$

NJVRBIQW 0...2 controls weighting of contributions for biquadratic restraining

- 0: equal weights of \mathbf{f}_i^{tav} and \mathbf{f}_i^{inst}
- 1: multiply \mathbf{f}_i^{tav} with $[1 - \exp(\Delta t/\tau_{Jr})]$
- 2: multiply \mathbf{f}_i^{tav} with zero

LE 0,1 local-elevation restraining [md++ only]

0: local-elevation off [DEFAULT]
 1: local-elevation on
 NGRID >1 number of grid points in local-elevation restraining
 DELTA ≥0.0 no elevation of potential if J is within DELTA of J0
 NTWJV ≥0 write ³J-value averages and LE grid to special trajectory
 =0: do not write [DEFAULT]
 >0: write every NTWJV-th step

- NTJVRA ≠ 0 requires NTJVR < 0
- NTJVRA = 0 and NTJVR < 0 results in a warning

LAMBDAS

NTIL
 NTLI(1..),NILG1(1..),NILG2(1..),ALI(1..),BLI(1..),CLI(1..),DLI(1..),ELI(1..)
 NTIL off, on, 0, 1
 off,0: no special treatment of interactions with individual λ-values
 on,1: interactions are treated with special individual λ-values
 NTLI(1..) interaction type to treat with individual λ: bond(1), angle(2),
 dihedral(3), improper(4), vdw(5), vdw_soft(6), crf(7), crf_soft(8),
 distanceres(9), dihedralres(10), mass(11)
 NILG1, NILG2 energy groups of interactions that are treated with individual
 λ-values
 ALI, BLI, CLI, Polynomial coefficients linking the individual λ-values to the
 DLI, ELI overall λ-value

- Input for this block is read linewise, i.e. you need to specify each interaction type within one separate line.

LOCALELEV

NTLES,NLEPOT,NTLESA,NTWLE,NLEPID,NTLEFR

NTLES 0..2 controls application of local-elevation
 0: no local-elevation potential energy function [DEFAULT]
 1: local-elevation using linear build up
 automatic force-constant update
 NLEPOT Number of applied potential energy functions
 NTLESA 1..2 controls reading of local-elevation potential energy functions
 1: read averages and parameters from startup file [DEFAULT]
 2: read averages and parameters from LEUS database file (@lud)
 NTWLE ≥ 0 write potential energy to special trajectory
 NLEPID(1..) ID of potential energy function to read and apply
 NTLEPFR(1.0),1 controls build up vs freezing of memory
 0: do memory build up (time-dependent potential energy function)
 1: freeze memory (no build up)

- List of local-elevation dihedrals (LOCALELEVSPEC) read from @led

MULTIBATH

NTBTYP (NUM)
 NBATHS
 TEMP0 (1..NBATHS) TAU(1..NBATHS)
 DOFSET
 LAST(1..DOFSET) COM-BATH(1..DOFSET) IR-BATH(1..DOFSET)

NTBTYP controls temperature coupling algorithm to use
 weak-coupling(0) use weak coupling scheme
 nose-hoover(1) use Nosé Hoover scheme
 nose-hoover-chains(2) use Nosé Hoover chains scheme
 NUM ≥ 0 number of chains in Nosé Hoover chains scheme [only specify when needed]
 NBATHS ≥ 0 number of temperature baths to couple to
 TEMP0() ≥ 0.0 bath reference temperature per bath
 TAU() ≥ 0.0 or -1 coupling time per bath, -1 turns coupling off
 DOFSET ≥ 0 number of distinguishable sets of degrees of freedom
 LAST() ≥ 0 last atom for set of degrees of freedom
 COM-BATH() ≥ 1 temperature bath to couple centre-of-mass motion of this set of d.o.f. to
 IR-BATH() ≥ 1 temperature bath to couple internal and rotational degrees of freedom of this set to

- LAST should be \leq NATTOT
- COM-BATH and IR-BATH should be between 1 and NBATHS

MULTICELL

NTM,NCELLA,NCELLB,NCELLC,
 TOLPX,TOLPV,TOLPF,TOLPFW

NTM 0,1 switch for multiple-unit-cell simulation
 0: single-unit-cell simulation [DEFAULT]
 1: multiple-unit-cell simulation
 NCELLA ≥ 1 number of subdivisions along a-axis
 NCELLB ≥ 1 number of subdivisions along b-axis
 NCELLC ≥ 1 number of subdivisions along c-axis
 TOLPX > 0.0 relative tolerance for coordinate periodicity check (not supported)
 TOLPV > 0.0 absolute tolerance for velocity periodicity check (not supported)
 TOLPF > 0.0 absolute tolerance for force periodicity check (not supported)
 TOLPFW > 0.0 absolute tolerance for force periodicity fix and warning (not supported)

- The indexing of subcells goes along *c* (fastest index), then *b*, then *a*
- Solvent molecules are reset to initial subcell
- Solute molecules may drift across subcells
- $NTB \neq 1,2$ requires $NTM = 0$

MULTIGRAIENT

NTMGRE, NTMGRP,
 NTMGRN,
 MGRVAR(1..NTMGRN), MGRFRM(1..NTMGRN), MGRNCP(1..NTMGRN),
 MGRCP(1..NTMGRN), MGRCPV(1..NTMGRN)

NTMGRE 0, 1 disables/enable multiple gradients
 0: disable gradients
 1: enable gradients
 NTMGRP 0..3 printout of the gradient curves in the output file
 0: don't print
 1: plot the curves
 2: print the values of the curves
 3: plot and print the curves
 NTMGRN ≥ 0 number of gradients
 MGRVAR() name of the variable to be affected, available are:
 TEMP0, CPIR, CDIR, RESO, CXR, COPR
 MGRFRM() 0..3 functional form of the gradient

- 0: linear interpolation between control points
- 1: cubic spline interpolation between control points
- 2: Bezier curve
- 3: Oscillation: $A \sin \left[\frac{2\pi}{T} (d - dt) \right] + b$

Note: MGRNCP is 2, $A = \text{MGRCP}[1]$, $T = \text{MGRCPV}[1]$, $dt = \text{MGRCP}[2]$, $b = \text{MGRCPV}[2]$

MGRCP() ≥ 2 number of control points

MGRCP() ≥ 0 time of the control point

MGRCPV() value of the variable at the control point

MULTISTEP

STEPS,BOOST

STEPS ≥ 0 calculate non-bonded every STEPSth step

BOOST 0,1 switch to control the method:

0: stored forces of STEPSth step are added every step

1: stored forces of STEPSth setp are multiplied by STEPS and added every STEPSth step [DEFAULT]

NONBONDED*

NLRELE

APPAK,RCRF,EPSRF,NSLFEXCL

NSHAPE,ASHAPE,NA2CLC,TOLA2,EPSLS

NKX,NKY,NKZ,KCUT

NGX,NGY,NGZ,NASORD,NFDORD,NALIAS,NSPORD

NQEVAL,FACCR,NRDGRD,NWRGRD

NLRLJ,SLVDNS

NLRELE -1.3 method to handle electrostatic interactions

-1: reaction-field method (LSERF compatibility mode)

0: no electrostatic interactions

1: reaction-field method

2: Ewald method

3: P³M method

APPAK ≥ 0.0 reaction-field inverse Debye screening length

RCRF ≥ 0.0 reaction-field radius

0.0: set reaction-field radius to infinity

> 0.0: reaction-field radius

EPSRF = 0.0 or ≥ 1.0 controls reaction-field permittivity

0.0: set reaction-field permittivity to infinity

≥ 1.0 : reaction-field permittivity

NSLFEXCL 0,1 contribution of excluded atoms to reaction field

0: contribution turned off

1: contribution considered [DEFAULT]

NSHAPE -1.10 lattice-sum charge-shaping function (-1: Gaussian)

ASHAPE > 0.0 width of the lattice-sum charge-shaping function

NA2CLC 0.4 controls evaluation of the lattice-sum A_2 term

0: $A_2 = \tilde{A}_2 = 0$

1: \tilde{A}_2 exact, $A_2 = \tilde{A}_2$

2: A_2 numerical, $\tilde{A}_2 = A_2$

3: \tilde{A}_2 exact from Ewald or from mesh and atom coords, A_2 numerical

4: \tilde{A}_2 averaged from mesh only, A_2 numerical

TOLA2 > 0.0 relative tolerance for numerical A_2 evaluation

EPSLS = 0.0 or ≥ 1.0 controls lattice-sum (external) permittivity

0.0: set lattice-sum permittivity to infinity (tin foil)

≥ 1.0 : lattice-sum permittivity

NKX,NKY,NKZ > 0 maximum absolute Ewald k-vector components
 KCUT > 0.0 Ewald k-space cutoff
 NGX,NGY,NGZ > 0 P³M number of grid points along the three box axes (even)
 NASORD 1.5 order of the mesh charge-assignment function
 NFDORD 0.5 order of the mesh finite-difference operator
 (0: *ik*-differentiation)
 NALIAS > 0 number of mesh alias vectors considered
 NSPORD order of the SPME B-spline function (not available)
 NQEVAL ≥ 0 controls accuracy reevaluation
 0: do not reevaluate accuracy
 >0: reevaluate accuracy every NQEVAL steps
 FACCUR > 0.0 rms force error threshold to recompute influence function
 NRDGRD 0,1 read initial influence function (and derivatives) from file (not implemented)
 NWRGRD 0,1 write final influence function (and derivatives) to file (not implemented)
 NLRLJ 0,1 controls long-range Lennard-Jones correction (not implemented)
 SLVDNS > 0.0 average solvent density for long-range Lennard-Jones cor-
 rection (not implemented)

- Numerical A_2 : by Ewald summation up to relative tolerance TOLA2
- Exact \tilde{A}_2 : by Ewald(abs(NLRELE)= 2) or based on mesh and exact atom coordinates (abs(NLRELE= 3,4))
- Average \tilde{A}_2 : based on mesh for atom coordinates averaged over box
- For a truncated octahedron box, NGA, NGB and NGC refer to the axes of the transformed triclinic cell.
- Choices for NSHAPE are found in Tab. 2-7.1
- NA2CLC= 1 requires abs(NLRELE)= 2
- NA2CLC= 4 requires abs(NLRELE)= 3,4
- NGX,NGY and NGZ must be even
- NTB= 0 requires NLRELE=-1,0,1
- NLRELE≠ 0,1 and ASHAPE>RCUTP results in a warning
- NA2CLC= 0 and NLRELE= 2,3 results in a warning
- P3M and Ewald require atomistic cutoff scheme
- P3M and Ewald can not be used with multiple energy groups

ORDERPARAMRES

NTOPR,NTOPRA,COPR,TAUJVR,UPDOPR,NTOPW

NTOPR -2...2
 -2: time-averaged using COPR×WOPR
 -1: time-averaged using COPR (WOPR ignored)
 0: no S^2 -order parameter restraining [DEFAULT]
 1: window-averaged using COPR (WOPR ignored)
 2: window-averaged using COPR×WOPR
 NTOPRA 0,1 controls reading of averages from startup file
 0: start from initial values [DEFAULT]
 1: read time-averages from startup file (for continuation of time-averaged run)
 COPR ≥ 0 S^2 -order parameter restraining force constant (weighted by individual WOPR)
 TAUOPR ≥ 0 coupling time for time-averaging, length of averaging window for window averaging
 UPDOPR > 0 update order parameters only every UPDOPR steps (only relevant for window averaging)
 NTWOP ≥ 0 write S^2 -value averages to special trajectory
 =0: do not write [DEFAULT]
 >0: write every NTWOP-th step

PAIRLIST

algorithm NSNB RCUTP RCUTL SIZE TYPE

algorithm standard, grid method for generating pairlist

standard(0) GROMOS96 like pairlist
 grid(1) md++ grid pairlist
 grid_cell(2) grid-based algorithm using a mask⁶
 NSNB >0 frequency (number of steps) a pairlist is constructed
 RCUTP >0.0 cut-off used in pairlist construction
 RCUTL >0.0 cut-off used in long range interaction
 SIZE >0.0, auto size of grid cell
 auto: 0.5*RCUTP
 TYPE chargegroup (0), atomic (1) type of cut-off
 chargegroup: chargegroup based cut-off
 atomic: atom based cut-off

PERSCALE

RESTYPE
 KDIH,KJ,T,DIFF,RATIO,READ

RESTYPE Special energy term to which periodic scaling should be applied
 0: Do not apply periodic scaling.
 1: Apply periodic scaling to ³J-value restraints
 KDIH ≥ 0.0 Maximum scaling factor for dihedral angle potential
 KJ ≥ 0.0 Maximum scaling factor for ³J-value restraint potential
 T > 0 Period of cosine scaling function
 DIFF ≥ 0.0 Minimum deviation from target value to start a scaling period
 RATIO > 0.0 Minimum fraction of T that needs to be passed before starting a new scaling period
 READ = 0,1 Read scaling parameters from coordinate file for continuation simulation

- RESTYPE=1 requires NTBDN=1

PERTURBATION

NTG,NRDGL,RLAM,DLAMT
 ALPHLJ,ALPHC,NLAM
 NSCALE

NTG 0,1 controls use of free-energy calculation
 0: no free-energy calculation is performed [DEFAULT]
 1: calculate $\frac{\partial \mathcal{H}(\lambda, \mu)}{\partial \lambda}$
 NRDGL 0,1 controls reading of initial values
 0: use initial λ parameter from PERTURBATION input block
 1: read initial λ value from startup file
 RLAM 0.0..1.0 initial value for λ
 DLAMT ≥ 0.0 rate of λ increase in time
 ALPHLJ ≥ 0.0 Lennard-Jones soft-core parameter
 ALPHC ≥ 0.0 Coulomb soft-core parameter
 NLAM > 0 power dependence of λ coupling
 NSCALE 0,1,2 turn energy group scaling on
 0: no scaling [DEFAULT]
 1: scaling
 2: scaled interactions only

- NTWG ≠ 0 requires NTG ≠ 0
- Perturbation topology file read from @pttopo
- λ (PERTDATA) read from @conf if NRDGL=1

PRECALCLAM

NRLAM, MINLAM, MAXLAM

NRLAM ≥ 0 determines calculation of \mathcal{H} and $\partial\mathcal{H}/\partial\lambda$ at alternative values of λ
0: off
> 0: precalculating energies and derivatives for NRLAM extra λ values
MINLAM 0.0 .. 1.0: minimum λ value to precalculate energies and derivatives
MAXLAM MINLAM .. 1.0: maximum λ value to precalculate energies and derivatives

POLARISE

COS,EFIELD,MINFIELD,DAMP,WRITE

COS 0,1,2 controls explicit inclusion of electronic polarisation effects
0: do not explicitly include electronic polarisation [DEFAULT]
1: use charge-on-spring model for dipolar polarisation
2: use charge-on-spring model for dipolar polarisation with off atom site
EFIELD 0,1 controls evaluation site for electric field
0: evaluate at atomic position of polarisable centres
1: evaluate at position of charges-on-spring
MINFIELD > 0.0 convergence criterium in iterative procedure to determine positions of charges-on-spring
DAMP 0,1 controls polarisability damping
0: use linear relationship between induced dipole moments and electric field
1: damp polarisability (with parameters from topology)
WRITE ≥ 0 write COS positions to special trajectory file
0: do not write COS positions
> 0: write COS positions every WRITEth step

POSITIONRES

NTPOR,NTPORB,NTPORS,CPOR

NTPOR 0..3 controls atom position restraining or constraining
0: no position re(con)straining [DEFAULT]
1: restraining with force constant CPOR (no B-factor weighting)
2: restraining with force constant CPOR weighted by atomic B-factors
3: position constraining
NTPORB 0,1 controls reading of reference positions and B-factors
0: read reference positions from startup file (@conf) [DEFAULT]
1: read reference positions and B-factors (if required) from special file (@refpos)
NTPORS 0,1 controls scaling of reference positions upon pressure scaling
0: do not scale reference positions [DEFAULT]
1: scale reference positions together with box parameters
CPOR ≥ 0.0 position restraining force constant

- NTPOR= 2 requires NTPORB= 1
- List of re(con)strained atoms (POSRESSPEC) read from @posresspec
- Reference positions in REFPOSITION blocks
- Without pressure coupling, NTPORS has to be 0

PRESSURESCALE

COUPLE,SCALE,COMP,TAUP,VIRIAL

SEMI(1..3)

PRES0(1, 3, 1..3)

COUPLE off,calc,scale controls calculation and scaling of pressure
off(0) no pressure calculation or scaling
calc(1) calculate pressure but no scaling

scale(2) calculate and couple pressure to a pressure bath
 SCALE off,iso,aniso,full controls isotropy of pressure scaling
 off(0) no pressure scaling
 iso(1) isotropic pressure scaling
 aniso(2) anisotropic pressure scaling (x-, y-, z-axes, no angle deformation)
 full(3) fully anisotropic pressure scaling
 semianiso(4) semi-anisotropic pressure scaling
 COMP > 0.0 isothermal compressibility
 TAUP ≥ 0.0 coupling relaxation time
 VIRIAL none,atomic,group controls type of virial for pressure calculation
 none(0) no pressure calculation
 atomic(1) atomic virial
 group(2) group-based virial according to PRESSUREGROUPS
 SEMI 0..2,0..2,0..2 (semianisotropic couplings: x-, y-, and z-axes)
 PRES0(,) ≥ 0.0 reference pressure in Tensor format

PRINTOUT

NTPR,NTPP

NTPR ≥ 0 controls printing of energies
 0: no printing out of energies [DEFAULT]
 > 0: print out energies every NTPR steps
 NTPP 0,1 controls dihedral angle transition monitoring
 0: no dihedral angle transition monitoring [DEFAULT]
 1: perform dihedral angle transition monitoring

- Data is printed to standard output.
- Dihedral angle transitions are printed to @trs.

QMMM

NTQMMM,NTQMSW,RCUTQ,NTWQMMM

NTQMMM 0,1 controls application of QM/MM
 0: do not apply QM/MM [DEFAULT]
 1: perform QM/MM simulation
 NTQMSW 0,1 software package to use for QM calculation
 0: MNDO
 1: TURBOMOLE
 RCUTQ ≥ 0.0 cutoff for electrostatic QM/MM interactions, inclusion of MM charge groups in QM Hamiltonian
 0.0: include all MM atoms
 > 0.0: include only atoms of charge groups closer than RCUTQ
 NTWQMMM ≥ 0 write QM/MM related data to special trajectory
 0: do not write [DEFAULT]
 > 0: write every NTWQMMMth step (not yet available)

- Note: QM/MM currently only applicable to systems with non-covalent interactions between QM and MM region.

RANDOMNUMBERS

NTRNG, NTGSL

NTRNG 0,1 random number generator
 0 use GROMOS 96 algorithm
 1 use GSL library (DEFAULT)
 NTGSL ≥ -1 GSL random number generation algorithm

-1 use default algorithm (mt 19937)
> = 0 run contrib/rng_gsl for a list of possible arguments

READTRAJ

NTRD,NTSTR,NTRB,NTSHK

NTRD 0,1 controls trajectory-reevaluation mode
0: do not use trajectory-reevaluation mode [DEFAULT]
1: use trajectory-reevaluation mode
NTSTR ;0 stride: should be the NTWX used to produce the analyzed trajectory
NTRB 1 obsolete option to control reading of box parameters (must be 1)
NTSHK 0..2 controls application of constraints
0: apply constraints with respect to previous coordinates [default]
1: apply constraints with respect to current coordinates
2: do not apply constraints (neither solute nor solvent)

- For consistency, a pairlist should have been made every NTSTR steps (or a divisor thereof) in the generating run
- Velocities and dependent quantities are zeroed
- NTEM \neq 0 requires NTRD= 0
- NTSD \neq 0 requires NTRD= 0
- NTB= 0 requires NTRD= 0
- NSCM> 0 and RTC> 0 (centre-of-mass removal and roto-translational constraints) are ignored when NTRD= 1.
- Coordinate trajectories (POSITIONRED and GENBOX) are read from @anatrj if NTRD \neq 0

REPLICA

NRET
RET(1..NRET)
LRESCALE
NRELAM
RELAM(1..NRELAM)
RETS(1..NRELAM)
NRETRIAL,NREQUIL,CONT

NRET \geq 1 Number of replica exchange temperatures
RET() \geq 0.0 Temperature for each replica
LRESCALE = 0,1 Scale temperatures after exchange trial
NRELAM \geq 1 Number of replica exchange lambda values
RELAM() \geq 0.0 Lambda value of each lambda-replica
RETS() \geq 0.0 Timestep of each lambda-replica
NRETRIAL \geq 0 Number of overall exchange trials
NREQUIL \geq 0 Number of exchange periods to equilibrate (disallow switches)
CONT = 0,1 Continuation run
0 start from one configuration file
1 start from multiple configuration files

- if CONT=1, the name specified for @conf will be split before the last "." and replica numbers inserted, e.g. input.cnf will be expanded to input_1.cnf .. input_n.cnf where n is the number of replicas
- NRESCALE \neq 0 requires NRET>1

ROTTRANS

RTC,RTCLAST

RTC = 0,1 Turn roto-translational constraints on (1)
RTCLAST > 0 Last atom of subset to be roto-translationally constraint

- Use either centre of mass removal or roto-translational constraints but not both!

SASA

NTSASA,NTVOL,P_12,P_13,P_1X,SIGMAV,RSOIV,AS1,AS2

NTSASA 0,1 controls use of SASA implicit solvent model

0: do not use SASA [DEFAULT]

1: use SASA

NTVOL 0,1 controls use of VOLUME correction to SASA implicit solvent model

0: do not use VOLUME correction [DEFAULT]

1: use VOLUME correction (requires NTSASA = 1)

P_12 > 0, < 1 pair parameter for SASA reduction for first neighbours

P_13 > 0, < 1 pair parameter for SASA reduction for second neighbours

P_1X > 0, < 1 pair parameter for SASA reduction for third and higher neighbours

SIGMAV > 0 scaling parameter for volume energy term ($\text{kJ}\cdot\text{mol}^{-1}\cdot\text{nm}^{-3}$)

RSOLV > 0 radius of solvent molecule for SASA calculation (nm)

AS1 > 0 an atom with SASA below this contributes to the VOLUME correction (nm^2)

AS2 > 0 an atom with SASA above this is not considered for the VOLUME correction (nm^2)

- NTSASA \neq 0 requires NTB=0
- NTVOL= 1 requires NTSASA= 1
- Suitable values of P_12, P_13 and P_1X for the SASA and SASA/VOL implicit solvent models are given in² and³
- SIGMAV is required if NTVOL= 1. Its parameterisation is discussed in³
- AS1 and AS2 are required if NTVOL= 1. Atoms with AS1<SASA<AS2 have a partial contribution determined by a switching function, thus AS1 and AS2 should in most cases be close to each other and close to zero.

STEP*

NSTLIM,T,DT

NSTLIM > 0 number of steps

T \geq 0.0 time at beginning of simulation

DT > 0.0 timestep

- Final configuration (POSITION,VELOCITY,GENBOX) written to @fin
- If NTRD \neq 0, NSTLIM is the total number of configurations, T the initial time of the first file, and DT the time interval between successive records on file
- If NTEM \neq 0, T and DT are irrelevant

STOCHDYN

NTSD,NTFR,NSFR,NBREF,RCUTF,CFRIC,TEMPSD

NTSD 0,1 controls stochastic dynamics mode

0: do not do stochastic dynamics [DEFAULT]

1: do stochastic dynamics

NTFR 0..3 defines atomic friction coefficients γ

0: set γ to 0.0 [DEFAULT]

1: set γ to CFRIC

2: set γ to CFRIC*GAM0

3: set γ to CFRIC* ω_i from Eq. 2-13.30

NSFR > 0 recalculate γ every NSFR steps

NBREF > 0 threshold number of neighbour atoms for a buried atom

RCUTF \geq 0.0 interatomic distance considered when calculating γ

CFRIC ≥ 0.0 global weighting for γ
TEMPSD ≥ 0.0 temperature of stochastic bath

- NTEM $\neq 0$ requires NTSD= 0
- NTRD $\neq 0$ requires NTSD= 0
- NTISTI $\neq 0$ requires NTSD $\neq 0$
- Atomic friction coefficients GAM0 (FRICTIONSPEC) read from @friction if NTSD $\neq 0$ and NTFR= 2
- If NTFR= 0, CFRIC is irrelevant
- If NTFR $\neq 3$, NSFR, NBREF and RCUTF are irrelevant

SYSTEM*

NPM,NSM

NPM 0,1 number of solute molecules
NSM ≥ 0 number of identical solvent molecules

- NPM= 0 and NSM= 0 are not allowed simultaneously
- Data on the system topology is read from @topo
- Note that MD++ as well as GROMOS++ do not accept NPM > 1 (solute molecules have to be explicitly replicated in the topology file)

WRITETRAJ

NTWX,NTWSE,NTWV,NTWF,NTWE,NTWG,NTWB

NTWX controls writing of coordinate trajectory

0: no coordinate trajectory is written [DEFAULT]
> 0: write solute and solvent coordinates every NTWX steps
< 0: write solute coordinates every abs(NTWX) steps

NTWSE ≥ 0 selection criteria for coordinate trajectory writing

0: write normal coordinate trajectory [DEFAULT]
> 0: write minimum-energy coordinate and energy trajectory (based on the energy entry selected by NTWSE and as blocks of length NTWX)

NTWV controls writing of velocity trajectory

0: no velocity trajectory is written [DEFAULT]
> 0: write solute and solvent velocities every NTWV steps
< 0: write solute velocities every abs(NTWV) steps

NTWF controls writing of force trajectory

0: no force trajectory is written [DEFAULT]
> 0: write solute and solvent forces every NTWF steps
< 0: write solute forces every abs(NTWF) steps

NTWE ≥ 0 controls writing of energy trajectory

0: no energy trajectory is written [DEFAULT]
> 0: write energy variables every NTWE steps

NTWG ≥ 0 controls writing of free energy trajectory

0: no free energy trajectory is written [DEFAULT]
> 0: write free energy variables every NTWG steps

NTWB ≥ 0 controls writing of block-averaged energy trajectory

0: no block-averaged energy trajectory is written [DEFAULT]
> 0: write block-averaged energies (and free energies if NTWG>0) every NTWB steps

- NTWSE $\neq 0$ requires NTWX $\neq 0$, NTWV= 0, NTWF= 0, NTWE= 0 or abs(NTWX), NTWG= 0, NTWB= 0
- NTWSE denotes a potential energy term (Sec. 4.17)
- NTG= 0 requires NTWG= 0
- NTEM $\neq 0$ requires NTWV= 0
- NTRD $\neq 0$ requires NTWV= 0
- Coordinates (POSITIONRED) written to @trc if NTWX $\neq 0$
- Velocities (VELOCITYRED) written to @trv if NTWV $\neq 0$

- Forces (FREEFORCERED,CONSFORCERED) written to @trf if NTWF \neq 0
- Energies (ENERGY03) written to @tre if NTWE \neq 0
- Volume and pressure quantities (VOLUMEPRESSURE03) written to @tre if NTWE \neq 0
- Free energy quantities (FREEENERDERIVS03) are written to @trg if NTWG \neq 0
- Block-averaged energies and fluctuations (BAENERGY03 and BAEFLUCT03) written to @bae if NTWB \neq 0
- Block-averaged volume and pressure quantities (BAVOLUMEPRESSURE03) written to @bae if NTWB \neq 0
- X(t),V(t-dt/2), and F_{uc}(t) are written at the beginning of a timestep, F_c(t) right after SHAKE
- If NTWSE \neq 0, a minimum-energy trajectory is written, i.e. only the configuration and energy corresponding to the lowest NTWSE component within a block of length abs(NTWX) steps is reported

Input data are described in Vol. 5 (Program Library Manual) and Vol. 7 (Tutorials, Benchmarks, Test Sets).

Examples of MD input files are named:

*.imd

CHAPTER 9

Output files for MD++

The data structure of the *output file* of the (simulation) programs will not be given here.

Output of programs is discussed in Vol. 5 (Program Library Manual) and Vol. 7 (Tutorials, Benchmarks, Test Sets).

Examples of MD output files are named:

*.omd

CHAPTER 10

Files accessed by MD++ for reading or writing

Indicated are: files for reading (R), files for write-up (W) and compulsory blocks (*).

@input Standard input (control) file (R; always)

TITLE*
SYSTEM*
ENERGYMIN
STOCHDYN
READTRAJ
STEP*
REPLICA
BOUNDCOND*
MULTICELL
MULTIBATH
PRESSURESCALE
MULTIGRAIENT
FORCE*
COVALENTFORM
CONSTRAINT*
POLARISE
INTEGRATE
CGRAIN
ROTTRANS
INNERLOOP
MULTISTEP
PAIRLIST*
NONBONDED*
INITIALISE
RANDOMNUMBERS
COMTRANSROT
POSITIONRES
DISTANCERES
DIHEDRALRES
JVALUERES
ORDERPARAMRES
DISTANCEFIELD
QMMM
LOCALELEV
PERSCALE
ELECTRIC
SASA
PERTURBATION
LAMBDA
PRINTOUT
WRITETRAJ
EWARN
EDS

@out Standard output file (W; always)
MD++ output

@fin Final configuration file (W; if NTRD= 0)
TITLE
POSITION (if no SHAKE failure)
SHAKEFAILPOSITION (if SHAKE failure)
SHAKEFAILPREVPOSITION (if SHAKE failure)
VELOCITY (if NTEM= 0 and NTRD= 0)
STOCHINT (if NTSD≠ 0)
GENBOX (BOX; if NTB≠ 0)
LATTICESHIFTS (if NTB= 0)
ROTTRANSREFPOS (if NTT≠ 0 and NTCNS(J)≠ 0 for at least one J)
REFPOSITION (if NTPOR≠ 0)
DISRESEXPAVE (if NTDIR= -1, -2)
JVALRESEXPAVE (if NTJVR= -1, -2)
ORDERPARAMRESEXPAVE (if NTOPR= -1, -2)
ORDERPARAMRESWINAVE (if NTOPR= 1, 2)
LEMEMORY (if NTLES≠ 0)
PERTDATA (if NTG≠ 0)

@trc Coordinate trajectory (W; if NTWX≠ 0)
TITLE
TIMESTEP
POSITIONRED
GENBOX (if NTB≠ 0)

@trv Velocity trajectory (W; if NTWF≠ 0)
TITLE
TIMESTEP
VELOCITYRED

@trf Force trajectory (W; if NTWF≠ 0)
TITLE
TIMESTEP
FREEFORCERED
CONSFORCERED

@tre Energy trajectory (W; if NTWE≠ 0)
TITLE
TIMESTEP
ENERGY03
VOLUMEPRESSURE03

@trg IOTRJG Free-energy trajectory (W; if NTWG≠ 0 and NTG≠ 0)
TITLE
TIMESTEP
FREEENERGYDERIVS03

@bae Energy block-average trajectory (W; if NTWB≠ 0)
TITLE

TIMESTEP
BAENERGY03
BAEFLUCT03

@topo Topology file (R; always)
TITLE (compulsory, first)
PHYSICALCONSTANTS (compulsory, second)
TOPVERSION (compulsory, third)
ATOMTYPENAME (compulsory)
RESNAME
SOLUTEATOM (compulsory)
CGSOLUTE
BONDSTRETCHTYPE or BONDTYPE or HARBONDTYPE (one of them is compulsory if
BONDH or BOND)
BONDH
BOND
BONDDP
BONDANGLEBENDTYPE or BONDANGLETYPE or BONDANGLEBENDTYPE (one of them
is compulsory if BONDANGLEH or BONDANGLE)
BONDANGLEH
BONDANGLE
IMPDIHEDRALTYPE
IMPDIHEDRALH
IMPDIHEDRAL
TORSDIHEDRALTYPE or DIHEDRALTYPE (either of the two; compulsory if DIHEDRALH or
DIHEDRAL)
DIHEDRALH
DIHEDRAL
LJPARAMETERS
CGPARAMETERS
SOLUTEMOLECULES
TEMPERATUREGROUPS
PRESSUREGROUPS
SOLVENTATOM (compulsory)
SOLVENTCONSTR
SASAPARAMETERS (if NTSASA= 1)

@conf Initial configuration (startup) file (R; always, except if NTRD= 1 and NTRB= 1)
TITLE (compulsory, first)
TIMESTEP
POSITION or POSITIONRED (either of the two; compulsory)
VELOCITY or VELOCITYRED (either of the two; if NTEM= 0 and NTRD= 0 and NTIVEL= 0)
LATTICESHIFTS (if NTB≠ 0 and NTISHI= 0)
STOCHINT (if NTSD≠ 0 and NTISTI= 0)
GENBOX (if NTB≠ 0)
ROTTRANSREFPOS (if NTT≠ 0 and NTCNS(J)≠ 0 for at least one J)
REFPOSITION (if NTTPOR≠ 0 and NTPORB= 0)
PERTDATA (if NTG≠ 0 and NRDGL≠ 0)
DISRESEXPAVE (if NTDIR= -1, -2 and NTDIRA≠ 0)
JVALRESEXPAVE (if NTJVR= -1, -2 and NTJVRA≠ 0)
ORDERPARAMRESEXPAVE (if NTOPR= -1, -2)
ORDERPARAMRESWINAVE (if NTOPR= 1, 2)
LEMEMORY (if NTLES≠ 0 and NTLESA≠ 0)

@refpos Reference coordinates for position re(con)straining (R; if NTPOR \neq 0 and NTPORB= 1)
TITLE (compulsory)
REFPOSITION (compulsory)

@posrespec Atom specification for position re(con)straining (R; if NTPOR \neq 0)
TITLE (compulsory)
POSRESSPEC (compulsory)

@distrest Distance specification for distance re(con)straining (R; if NTDIR \neq 0)
TITLE (compulsory)
DISTANCERESSPEC
PERTDISRESSPEC
MDISRESSPEC
DFRESSPEC
PERTDFRESSPEC

@dihrest Dihedral specification for dihedral-angle re(con)straining (R; if NTDLR \neq 0)
TITLE (compulsory)
DIHEDRALRESSPEC
PERTDIHRESSPEC

@jval 3J -value specification for 3J -value restraining (R; if NTJVR \neq 0)
TITLE (compulsory)
JVALRESSPEC (compulsory)

@order S^2 -value specification for S^2 -order parameter restraining (R; if NTOPR \neq 0)
TITLE (compulsory)
ORDERPARAMRESSPEC (compulsory)

@qmmm QM/MM specification file (R; if NTQMMM \neq 0)
TITLE (compulsory)
QMZONE (compulsory)
QMUNIT (compulsory)
MNDOFILES (if NTQMSW= 0)
MNDOHEADER (if NTQMSW= 0)
TURBOMOLEFILES (if NTQMSW= 1)
TURBOMOLETOOLCHAIN (if NTQMSW= 1)
TURBOMOLEELEMENTS (if NTQMSW= 1)

@led Coordinate specification for local-elevation (R; if NTLES \neq 0)
TITLE (compulsory)
LOCALELEVSPEC (compulsory)

@lud LEUS biasing potential database (R; if NTLES= 2)
TITLE (compulsory)
LEUSGRID (one or more)

@friction Atomic friction coefficients for stochastic dynamics (R; if NTSD \neq 0 and NTFR= 2)

TITLE (compulsory)

FRICCTIONSPEC (compulsory)

@pttopo Data determining perturbation (R; if NTG \neq 0)

TITLE (compulsory)

PERTATOMPARAM

MPERTATOM PERTATOMPAIR

PERTATOMGROUPS

PERTPOLPARAM

PERTBONDSTRETCHH

PERTBONDSTRETCH

PERTCONSTRAINT03

PERTBONDANGLEH

PERTBONDANGLE

PERTIMPROPERDIHH

PERTIMPROPERDIH

PERTPROPERDIHH

PERTPROPERDIH

@anatrj Input coordinate trajectories (R; if NTRD \neq 0)

TITLE

series of

TIME

POSITIONRED

BOX (if variable box)

CHAPTER 11

Other non-GROMOS formats

Some GROMOS programs can read non-GROMOS data and formats, e.g. protein data bank data and formats, see Vol. 5 (Program Library Manual).

CHAPTER 12

List of GROMOS blocknames

Three *categories of blocks* are distinguished:

- data blocks,
- MD input blocks,
- molecular topology blocks.

The *current GROMOS blocknames* are listed below. In addition to the following reserved names, no block may be called 'END'.

Data blocks

ACCEPTORMASS	HYDROGENMASS
ANGLETYPECONV	IMPDIHEDRALTYPECODE
ATOMNAMELIB	IMPROPERTYPECONV
ATOMTYPECONV	JOBSCRIPTS
BFACTOR	JVALUERESEPS
BFACTORANISO	JVALUERESEXPAVE
BONDANGLEBENDTYPECODE	JVALUERESSPEC
BONDSTRETCHTYPECODE	LEDIHSPEC
BONDTYPECONV	LEMEMORY
BOX	LINKADDITION
CONSFORCE	MASSATOMTYPECODE
CONSFORCERED	MISCELLANEOUS
DIFFSTAT	MIXEDATOMLJPAIR
DIHEDRALTYPECODE	MPERTATOM
DIHEDRALTYPECONV	MTBUILDBLEND
DIHRESSPEC	MTBUILDSOLUTE
DIPMSTAT	MTBUILDSOLVENT
DISRESEXPAVE	MULTIPLICITY
DISRESSPEC	NOECALCSPEC
DISTANCERESSPEC	NOEGORGROMOS
ENERGIES	NOELIB
ENERGY	NOESPEC
ENERGY03	ORDERPARAMRESEXPAVE
ENERTRJ	ORDERPARAMRESWINAVE
FILENAMES	PERTATOM
FORMAT	PERTATOMPAIR
FOURDIMATOMSPEC	PERTBONDANGLE
FREE3D4DDATA	PERTBONDANGLEH
FREEENERGY3D4	PERTBONDSTRETCH
FREEENERGYDERIVS03	PERTBONDSTRETCHH
FREEENERGYLAMBDA	PERTDATA
FREEFORCE	PERTDIHRESSPEC
FREEFORCERED	PERTDISRESSPEC
FREELAMBADATA	PERTIMPROPERDIH
FRENERTRJ	PERTIMPROPERDIHH
FRICCTIONSPEC	PERTPROPERDIH
GENBOX	PERTPROPERDIHH

POSITION
POSITION4THD
POSITION4THDRED
POSITIONFOURTHM
POSITIONOF
POSITIONRED
POSITIONSECONDM
POSITIONSECONDMT
POSITIONTHIRDM
POSRES
POSRESSPEC
QUANDISTRIB
QUANENERAVER
QUANSUMENERAVER
QUANTIMECORR
QUANTIMECORRSPE
QUANTIMESERIES
QUANTITYAVER
REFPOSITION
RESIDUENAMELIB

RUNDATA
SASASPEC
SHAKEFAILPOSITION
SHAKEFAILPREVPOSITION
SINGLEATOMLJPAIR
SOLVSTAT
STOCHINT
STOCHINT4THD
TIMESTEP
TITLE
TRANSFORM
TRICLINICBOX
VARIABLES
VELOCITY
VELOCITY4THD
VELOCITY4THDRED
VELOCITYRED
VOLUMEPRESSURE
VOLUMEPRESSURE03

MD input blocks

BOUNDCOND
CGRAIN
COMTRANSROT
CONSTRAINT
COVALENTFORM
DIHEDRALRES
DISTANCEFIELD
DISTANCERES
EDS
ELECTRIC
ENERGYMIN
EWARN
FORCE
INITIALISE
INNERLOOP
INTEGRATE
JVALUERES
LAMBDA
LOCALELEV
MULTIBATH
MULTICELL
MULTIGRAIENT

MULTISTEP
NONBONDED
ORDERPARAMRES
PAIRLIST
PERSCALE
PERTURBATION
POLARISE
POSITIONRES
PRESSURESCALE
PRINTOUT
QMMM
RANDOMNUMBERS
READTRAJ
REPLICA
ROTTRANS
SASA
STEP
STOCHDYN
SYSTEM
WRITETRAJ

Topology blocks

ATOMTYPENAME
BOND
BONDANGLE
BONDANGLEBENDTYPE
BONDANGLEH
BONDANGLETYPE
BONDDP
BONDH

BONDSTRETCHTYPE
BONDTYPE
CGSOLUTE
CONSTRAINT
DIHEDRAL
DIHEDRALH
DIHEDRALTYPE
HARMBONDANGLETYPE

HARMBONDTYPE
IMPDIHEDRAL
IMPDIHEDRALH
IMPDIHEDRALTYPE
LJPARAMETERS
CGPARAMETERS
PATHINTSPEC
PHYSICALCONSTANTS
RESNAME

SASAPARAMETERS
SOLUTEATOM
SOLVENTATOM
SOLVENTCONSTR
SUBMOLECULES
TITLE
TOPVERSION
TORSDIHEDRALTYPE

CHAPTER 13

Recommendations for standard input and output file names

molecular building blocks	*.mtb
interaction-function parameters	*.ifp
script to run the program	*.run
input file	*.imd
output file	*.omd
configuration	*.cnf
topology	*.top
perturbation topology	*.ptp
reference positions and possibly B-factors for position	
re(con)straining (if in a file separate from .cnf)	*.rpr
position restraints	*.por
distance restraints	*.dsr
dihedral restraints	*.dhr
3J -value restraints	*.jvr
S^2 -order parameter restraints	*.opr
crystallographic restraints	*.xrs
LEUS database file	*.lud
local-elevation dihedrals	*.led
atomic friction coefficients	*.frc
P3M optimal influence function (G_{hat})	*.ght
gromos++ specific libraries	*.lib

trajectories:

coordinates	*.trc
velocities	*.trv
forces	*.trf
energies	*.tre
free energies	*.trg
special trajectories	*.trs
block average energies	*.bae
block average free energies	*.bag

Bibliography

- [1] W. Hasel, T. F. Hendrickson, and W. C. Still. A rapid approximation to the solvent accessible surface areas of atoms. *Tetra. Comput. Method.*, 1:103–116, 1988.
- [2] F. Fraternali and W.F. van Gunsteren. An Efficient Mean Solvation Force Model for Use in Molecular Dynamics Simulations of Proteins in Aqueous Solution. *J. Mol. Biol.*, 256:939–948, 1996.
- [3] J.R. Allison, K. Boguslawski, F. Fraternali, and W.F. van Gunsteren. A refined, efficient mean solvation force model that includes the interior volume contribution. *J. Phys. Chem. B*, 115:4547–4557, 2011.
- [4] G. Nagy and C. Oostenbrink. Dihedral-based segment identification and classification of biopolymers I: Proteins. *J. Chem. Inf. Model.*, 54:266 – 277, 2014.
- [5] G. Nagy and C. Oostenbrink. Dihedral-based segment identification and classification of biopolymers II: Polynucleotides. *J. Chem. Inf. Model.*, 54:278 – 288, 2014.
- [6] T. Heinz and P.H. Hünenberger. A fast pairlist construction algorithm for molecular simulations under periodic boundary conditions. *J. Comput. Chem.*, 25:1474, 2004.