

ORIENT version 4.9

by

Anthony J. Stone

with contributions from

Andreas Dullweber, Ola Engkvist, Elena Frascini, Matthew P. Hodges,
Andrew Meredith, Paul L. A. Popelier and David J. Wales

Contents

1	Introduction	1
1.1	Background and bibliography	1
1.2	Acknowledgements	2
1.3	Program overview	2
2	Data conventions	3
3	Specifying program limits	4
4	Defining variables and constants	5
5	Various control commands	5
6	Defining atoms, molecules and crystals	8
6.1	General non-linear molecules	8
6.1.1	Molecule definitions	8
6.1.2	Site definitions	9
6.2	Linear molecules	11
6.3	Atoms	11
6.4	Lattice	11
6.5	Layers	14
6.6	Crystal growth and aggregation	15
7	Defining site properties and site-site interactions	16
7.1	Site types	16
7.2	Defining the interactions between sites	17
8	Modifying a molecule description	20
9	Distributed polarizabilities	22
10	Displaying information	25
11	Calculations of energy and related properties	25
12	Geometry optimization	28
13	Monte Carlo Simulations in Orient	32

13.1	Adjustment period	33
13.2	Container	34
13.3	Thermodynamical properties	34
13.4	Histogram	35
13.5	Jump Walking	35
13.6	Lennard-Jones systems	37
13.7	Quenching	37
13.8	Statistics	38
13.9	Step control	38
13.10	Temperatures	39
14	Basin-Hopping in Orient	40
14.1	Syntax	40
14.2	Interrupting a basin-hopping job	42
15	Molecular Dynamics in Orient	42
15.1	Integration accuracy	43
15.2	Energy control	44
15.3	Lennard-Jones systems	44
15.4	Integration method	45
15.5	Plotting configurations	45
15.6	Quenching	46
15.7	Dumping and restarting a trajectory	46
15.8	Statistics	47
15.9	Time step control	47
16	Calculation of the second virial coefficient	48
17	Optimizing potential parameters	50
17.1	Repulsion parameters	50
17.2	Dispersion damping parameters	52
18	Graphical displays	53
18.1	Setting up the grid	53
18.2	Defining maps	54
18.3	Displaying maps	55

18.4 Interacting with the display	57
19 Using Orient as a subroutine for other programs	58

1 Introduction

ORIENT is a program for carrying out calculations of various kinds for an assembly of interacting molecules. It uses a site-site potential specified by the user, including electrostatic, induction, repulsion, dispersion and charge-transfer interactions if required. The electrostatic interactions may be described by simple point charges or by more elaborate descriptions involving distributed multipoles up to rank 5 if required. Distributed polarizabilities may be used, and the site-site repulsion and dispersion and charge-transfer terms may be anisotropic. The energy of the assembly can be calculated at specified configurations, and the geometry can be optimized to find minima and transition states and the paths between them. Molecular dynamics and Monte Carlo calculations can be performed on the assembly, and properties such as the virial coefficient can be calculated. Systems involving molecular clusters or periodic adlayers adsorbed on crystal surfaces can also be studied. The program has facilities for optimizing dispersion and repulsion parameters by reference to *ab initio* data.

IMPORTANT: The program is not in the public domain and may not be distributed without permission. Please ask any colleagues who are interested in using the program to send email to ajs1@cam.ac.uk. Please acknowledge use of the program in any publications arising from its use, by citing appropriate references from the bibliography below. A suitable form of reference to the program itself is also listed (?).

1.1 Background and bibliography

The theoretical ideas concerning intermolecular potentials that provided the motivation for the ORIENT program were reviewed by [Stone and Price \(1988\)](#). The method of distributed multipole analysis that is usually used to determine the multipole moments assigned to each site was described by [Stone \(1981\)](#) and by [Stone and Alderton \(1985\)](#). The program for calculating them forms part of the CAMCASP package ([Misquitta and Stone, 2011](#)), and is also available in MOLPRO ([Werner and Knowles, 1998](#)). Distributed multipoles can also be calculated using the Gaussian program system ([Frisch *et al.*, 1998](#)) together with the GDMA program ([Stone, 1998, 2005](#)).

The theory of distributed polarizabilities was given by [Stone \(1985\)](#), and the practicalities of calculating them were discussed by [Le Sueur and Stone \(1993, 1994\)](#). All of these issues are also discussed in *The Theory of Intermolecular Forces* ([Stone, 1996](#)). Molecular polarizabilities are listed in Appendix D of [Gray and Gubbins \(1984\)](#) for a number of small molecules, and these can be used if distributed polarizabilities are not available. Repulsion parameters for exponential repulsion are available for some atoms; for instance [Mirsky \(1978\)](#) gave recommended values. [Williams \(1965, 1967\)](#) gave values for hydrocarbons. In fact these are tabulations of ‘exp-6’ parameters, so they include atom-atom dispersion coefficients too. Other sources of dispersion coefficients include the values derived from dipole oscillator strength distributions by Meath and his colleagues; see for example [Kumar and Meath \(1994\)](#). Repulsion parameters have been calculated for a number of small molecules by [Stone and Tong \(1994\)](#) using a test-particle technique; other methods include the overlap model of [Wheatley and Price \(1990\)](#). Repulsion and dispersion energies can be calculated point by point using intermolecular perturbation theory (IMPT) ([Hayes and Stone, 1984](#)) or symmetry-adapted perturbation theory (SAPT) ([Jeziorski *et al.*, 1994](#)), and a suitable model fitted to the data using the ORIENT program ([Fraschini and Stone, 1998](#); [Hodges and Stone, 2000](#)).

Applications of the ORIENT program to the calculation of structures of complexes of small molecules have been described in a number of papers, including [Buckingham and Fowler \(1985\)](#), [Buckingham *et al.* \(1986\)](#) and [Price and Stone \(1987\)](#). The program has also been used in the calculation of induced dipole moments in molecular complexes ([Fowler and Stone \(1987\)](#), [Le Sueur *et al.* \(1991\)](#)).

The optimization procedure used by the current version of the program is the Eigenvector Following method of [Cerjan and Miller \(1981\)](#) with refinements by [Wales \(1991\)](#). It requires first and second derivatives of the potential, and the techniques used to obtain these were described by [Popelier and Stone](#)

(1994). Some applications of these optimization methods have been described by Popelier *et al.* (1994) and Wales *et al.* (1995).

The program has been developed recently to permit the study of molecules, molecular clusters and molecular adlayers adsorbed on ionic crystal surfaces (Meredith and Stone, 1996; Engkvist and Stone, 1999a,b). It is also possible to study crystal growth morphology by calculating attachment energies and related quantities; the ORIENT program makes it possible to use much more accurate potentials than those previously used for this purpose (Engkvist *et al.*, 2000). The program can also calculate interaction energies between different crystal faces for use in studies of crystal aggregation.

1.2 Acknowledgements

The program uses a number of LAPACK routines (Anderson *et al.*, 1999), and several Numerical Recipes routines (Press *et al.*, 1992), some slightly modified, which are incorporated in the program by permission.

1.3 Program overview

The system is defined in a hierarchical fashion. At the lowest level is the *site*, which is typically an atom or a small group of atoms such as a methyl group. A site has its own local coordinate frame, and normally carries a set of multipole moments, defined with respect to that frame. It may also have a set of local polarizabilities. Finally it may have properties in common with other sites of the same *type*; such properties include the van der Waals radius (which may be zero), the atomic number and the atomic mass. There may also be properties defined for *pairs* of sites: these include non-local polarizabilities, and repulsion and dispersion interactions. The repulsions may be hard-sphere or exponential. The repulsion and dispersion properties are defined in terms of site types; all sites of the same type have the same repulsion-dispersion properties. A site need not have a specified type; in that case it is assigned type zero, and can have no repulsion or dispersion interactions, though it may carry multipole moments. Sites of type zero may also be used for such purposes as defining the molecular centre of mass.

A *molecule* is then defined as an assembly of sites. It is possible for a molecule to consist of a single site, which may be isotropic (*i.e.*, an atom) or anisotropic (for example, a methane molecule could be represented by a single site.) More usually, the molecule has a base site, used to specify the position and orientation of the molecule, and the positions and orientations of the sites within the molecule are defined with respect to a local coordinate frame attached to the base site. It is possible, in an extension of this approach, for a site position and orientation to be defined in the local coordinate system of some other 'parent' site in the same molecule, and a molecule can be built up step by step, each site being 'linked' to (*i.e.* defined in the local coordinate frame of) one of the sites already defined. The 'links' are for purposes of definition only, and do not necessarily have anything to do with chemical bonds. However, if they do correspond to chemical bonds, the geometry can be defined in terms of dihedral rotations around the bonds.

Finally the system as a whole is defined as a list of molecules, each with a specified position and orientation in the global coordinate frame.

It is possible to define a *lattice*, which like a molecule is an assembly of sites. A lattice in ORIENT is a representation of a crystal surface, and is periodic in the global *xy* plane and in negative *z*. The sites in a lattice have fixed positions. An *adlayer* (or *layer*) is periodic in the *xy* plane but not in *z*, and is composed of molecules – all of the molecules in the system – repeated with specified periodicity. A *nanocrystal* comprises the assembly of all molecules in the system, repeated periodically for a finite number of unit cells in *x*, *y* and *z*. Consequently the complete system may be

- An assembly of one or more molecules;

- An assembly of one or more molecules interacting with a periodic surface;
- A periodic layer of molecules interacting with a surface;
- A nanocrystal interacting with a surface; or
- An isolated periodic layer of molecules.

The distances and angles needed to define the geometry may be specified numerically, or in terms of named constants and variables. The values of named variables can be changed by input commands, so that (for example) the energy of a system can be evaluated at a number of geometrical configurations. Variables are used for values that might change, for example during an optimization, and constants for values that are guaranteed to remain fixed.

Properties that can be calculated using the program include the energy of the system, which can be optimized to find stationary points, especially minima and saddle points on the intermolecular potential energy surface. Harmonic intermolecular vibration frequencies can be calculated at stationary points, and the path to optimization can be displayed as an xyz file and viewed using the Xmol or XMakeMol programs (Hodges, 1998). By finding paths from a saddle point to minima on either side, interconversion pathways can be displayed. Monte Carlo calculations may be carried out; and molecular dynamics can be performed for molecular clusters, though not at present for systems involving surfaces. Potential energy surfaces can be explored in a systematic way by carrying out high-temperature Monte Carlo simulations and quenching at intervals to find the nearest potential energy minimum. The pressure and dielectric virial coefficients can also be computed.

2 Data conventions

In the following, square brackets [. . .] are used to indicate optional elements in a command. The brackets themselves should not appear in the data. The vertical bar | is used to separate alternatives. If alternatives are enclosed in braces, { . . . }, one of them must be present. Items in *italic* must be replaced by a suitable data value; items in typewriter style (e.g. MOLECULE) should appear as given, except that upper and lower case are synonymous in all such keywords.

In commands that contain a number of subsidiary options, the options may normally occur in any order. Lines in the input dataset are limited to 128 characters (though this limit can be changed). If a command is too long to fit on one input line, it may be continued on to as many lines as necessary, provided that every line but the last ends with the concatenation sequence '+++'. The concatenation of such a set of lines (with the concatenation sequences removed) is referred to below as a 'logical line'. Most commands occupy a single logical line, but some are followed by a list of sub-commands, terminated by a line containing just the word 'END'. Within a logical line, the data is generally given in keyword–value form, and is always in free format, with data items separated by white space.

The concatenation sequence '+++' is used in this document where the description of a complex command runs on to two or more lines.

Within any data line, items enclosed in parentheses, '()', are ignored, provided that they are separated from other items by white space.

Lines in the data of the form

```
#include "file"
```

are replaced by the contents of the specified file. The file-name should be given in a form which is appropriate for the Fortran OPEN statement. On Unix systems this usually means that shell meta-characters will not be interpreted. Included files may be nested.

```

Allocate
Molecules 5
Sites 50 Polarizable 0
Alphas 256
S_functions 400
Variables 1500
Types 20
Pairs 100
Parameter-sets 200
Layers 0
Fitting parameters 32 points 500
End

```

Figure 1: Default limits for the Orient program.

3 Specifying program limits

It is possible to specify the values for certain array sizes in the program, so that arrays of an appropriate size can be allocated. This means that the memory used by the program can be kept small for small problems, but can be increased to cope with larger systems without the need to recompile. The ALLOCATE command specifies the values, and must precede any items in the data file that define molecules or invoke calculations. It is recommended that it appear immediately after any TITLE and NOTES describing the calculation.

```

ALLOCATE
name value
:
END

```

The keywords PARAMETERS or LIMITS can be used instead of ALLOCATE. The limits that can be set in this way are

MOLECULES *n*

The maximum number *n* of molecules that can be defined.

SITES *n* [POLARIZABLE *p*]

The maximum number *n* of sites, of which at most *p* may be polarizable. In estimating the number required, remember that a site is used to define the reference position of a molecule, and also that an extra site is used for the centre of mass of each molecule in an optimization, even if a site is specified for it in the molecule definition.

ALPHAS *n*

The maximum number *n* of polarizability matrices. Every set of polarizabilities α^{ab} for sites *a* and *b* requires one such matrix. The matrix for α^{ba} is stored separately from α^{ab} . The default value of 256 should normally be adequate.

S-FUNCTIONS *n*

The value *n* must be at least the maximum number of *S* functions required to describe the interaction between any single pair of sites. However, if induction energies are to be calculated, and the RECALC_SFNS switch is off (see below), *n* should be at least the number of *S* functions needed to describe the interactions between *all* pairs of sites.

VARIABLES *n*

The maximum number of geometrical variables and constants. (The space is shared between them.) An entry in this space is needed for every geometrical parameter (position or rotation) used in the definition of a molecule or site, whether it is given in terms of a named variable or constant or as a literal numerical constant. The default of 1500 should normally be enough.

TYPES *n*

The maximum number of site types that may be used by the program.

PAIRS *n*

The maximum number of pair interactions that may be defined. See the description of the PAIRS data section on p. 17 for further explanation.

PARAMETER-SETS *n*

The maximum number of pair interaction parameter-sets that may be defined. See the description of the PAIRS data section on p. 17 for further explanation.

FITTING [PARAMETERS *ng*] [POINTS *nd*]

Maximum number of fitted parameters and data points for the fitting procedure (§17, p.50)

LAYERS *n*

The maximum number of adsorbed layers of molecules on a crystal surface.

The default values of the parameters correspond to the declaration shown in Fig. 1.

4 Defining variables and constants

The geometry of a molecule can be defined in terms of named constants and variables. As the terms imply, constants have fixed values for the duration of the calculation, while variables may be changed either by the program or by subsequent commands in the data. The name of a constant or variable can be any string of characters; it can even include spaces, provided that it is enclosed in single or double quotes, though this is not recommended.

```
CONSTANTS
name value [A|B|D]
:
END
VARIABLES
name value [A|B|D]
:
END
```

The quantity *name* is defined to be a constant or variable with the value given. The value may be followed by a letter specifying a unit: A for Angstrom, B for Bohr, D for degrees of angle, or R for radians. Use N for a dimensionless quantity. If no unit is given, the value is assumed to be a distance in the current length units (so 'D' or 'R' must be present for angles, even if zero). An example of a declaration of variables and constants is given in Fig. 2.

The values of variables can be changed subsequently by the following commands:

```
SET name [=] value [unit] [ name [=] value [unit] ...]
```

Set the variable *name* to the value specified. A unit must be given if the value is not in the current units for the variable type (A, B or D for Å, bohr or degrees respectively), or if the next variable name in the list could be confused with a unit name. Several variables may be set in the same command.

```
ADD name increment [unit] [name increment [unit] ...]
```

Increment the variable '*name*' by the amount specified. Several variables may be incremented if required. For example:

```
Set  alpha1 = 90  beta1 = 120
Add  x1 0.2 A
```

5 Various control commands

```
TITLE title
SUBTITLE subtitle
```

```

Constants
C-C  1.54 A  ( bond lengths )
C=C  1.34 A
C-H  1.08 A
T    109.54 D
End
Variables
x1  0 A      ( Position of molecule 1 )
y1  0 A
z1  0 A
alpha1  0 D  ( Orientation of molecule 1 )
beta1   0 D
gamma1  0 D
x2  0 A      ( Position of molecule 2 )
y2  0 A
z2  6 A
alpha2  0 D  ( Orientation of molecule 2 )
beta2   0 D
gamma2  0 D
End

```

Figure 2: Example of a declaration of variables and constants.

Specify a title, and subtitle if required, to be printed at appropriate places in the output. The text should be enclosed in quotes (single or double) if it contains spaces. TITLE alone causes the current title and subtitle to be printed.

DEFAULT RANK L_{def}

In the absence of a RANK L option in a site definition, the site is assumed to have multipoles up to rank L_{def} . If there is no DEFAULT command, $L_{\text{def}} = -1$, meaning that the usual default is for sites to have no multipoles.

UNITS [kJ/mol | HARTREE | mH | eV | meV | cm-1 | K] [ANGSTROM | BOHR]

This declaration controls the input and output of distance and energy values. After an occurrence of this declaration in which ANGSTROM or BOHR is specified, subsequent distances in the input data will be assumed to be in that unit, and output of distance values will also be in the same unit except for certain diagnostic data that are always given in atomic units. The default unit for distances is BOHR.

Similarly the declaration can be used to specify the unit used for printing interaction energy values. The default is HARTREE.

This command also affects values read by the PAIRS declaration. For instance, after a UNITS ANGSTROM eV declaration, values of C_6 will be assumed to be given in units of $e\text{V}\text{\AA}^6$.

Multipole moments and polarizabilities are always assumed to be in atomic units.

UNITS POP { LENGTH | ENERGY }

Revert to the units of length or energy previously in use. This is recommended for use in included data files to avoid inadvertently changing the units used by the main data file.

ESLIMIT n

Limit electrostatic interactions to terms in R^{-k} that have $k \leq n$. The default is 5. It is possible to calculate all terms up to R^{-6} , but the higher terms are numerous and complicated, so in optimizations it is advisable to carry out an initial stage using a limit of 5 or even less, and to calculate the higher terms, if required, only for the final refinement.

SUPPRESS [MOLECULE] *name*

Mark molecule *name* temporarily as inactive. It will be ignored in all calculations until a corresponding RESTORE command is encountered.

RESTORE [MOLECULE] *name*

Cancel a previous SUPPRESS command.

RESTORE GEOMETRY *filename*

Restore the geometry of the system dumped to the specified file in a previous calculation or a previous step of this one. If the geometry was dumped in a previous calculation, the system definition must be given as in the previous calculation, before the RESTORE command.

SWITCH [*name* ON/OFF] . . .

This command allows internal switches to be set. The switches available are:

REFLECT

If ON the specification of the site data and the molecule positions and orientations are reflected to the output when they are read from the data.

INDUCE

If ON, the induction energy is calculated and added to the electrostatic energy in all energy calculations. Note that this will slow the calculation down substantially; the ratio will depend on the number of polarizable sites and the rank of polarizabilities specified. If there are no polarizable sites, there will be no reduction in speed.

ITERATE

If ON, the induction energy calculation, and the calculation of induced moments, is iterated to convergence. If OFF, a single iteration only is performed, to give the first-order induced moments and the corresponding induction energy.

RECALC_SFNS

In a calculation of the induction energy with iteration to convergence, the S functions needed, and their derivatives, are normally held in memory and re-used at each iteration. This can take up too much memory when large systems are involved. If RECALC_SFNS is switched ON, the S functions are recalculated at each iteration, which is slower but uses much less memory. It is OFF by default.

REFLECT and ITERATE are ON by default. INDUCE is OFF unless polarizabilities have been defined.

DEBUG [*n*₁ *n*₂ . . .]

Set or unset various switches that control debugging. *n*₁, *n*₂, *etc.*, are integers between -16 and +16. If *n* > 0, debug switch *n* is turned on; if *n* < 0, switch -*n* is turned off. The switches have different effects in different parts of the program, and should not be needed in normal use.

PRINTLEVEL *n*

n determines the amount of output produced by the program. *n* = 0 (the default) gives the minimum amount of information; *n* = 2 or 4 may be used for more detailed printing. Higher values will usually produce a great deal of output and should be avoided except for diagnostic purposes.

TIME

Print the cpu time used since the last TIME command, and the total time used so far.

NOTE *text*

Ignored. Can be used to insert comments in the data. Blank lines in the data are also ignored (except within TABULATE, where they are copied to the output).

IGNORE

:

IGNORE END

Ignore all data lines between IGNORE and IGNORE END. This may be used to enclose an extended comment, or to 'comment out' parts of the data file that are to be omitted in a particular run, without having to delete them from the file.

PLOT *options*

Plot the current system configuration in a form suitable for viewing with ‘xmol’ or ‘xmakemol’ (Hodges, 1998). This command can be used by itself as a way of checking the system configuration before embarking on a lengthy calculation, but it is more usually used in connection with geometry optimization, and the syntax is described in detail on p. 31.

STOP or FINISH

Stop.

6 Defining atoms, molecules and crystals

6.1 General non-linear molecules

The definition of a molecule consists of a list of sites. Each site carries a coordinate frame of its own, whose position and orientation are specified by reference to some previously defined coordinate frame. The first site specified for a molecule defines a local origin and coordinate frame for the molecule, in terms of the global coordinate system, and normally the position and orientation of every other site are defined with respect to this local frame. However it is also possible to define the position and orientation of any site by reference to the local frame for any site in the same molecule that has already been defined. Each site may also have a *type*, undefined by default, that specifies the nature of its repulsion and dispersion interactions. A site with undefined type has only electrostatic interactions.

6.1.1 Molecule definitions

Each molecule is defined by a declaration of the form

```
MOLECULE name AT {x y z | POLAR r θ φ} [molecule attributes] +++
  [base site attributes]
  base site multipoles, if any
  ⋮
  site definitions
  ⋮
END
```

The molecule name, like site names generally, is any convenient character string; if it contains spaces it must be enclosed in single or double quotes. To avoid ambiguities in subsequent references, each molecule should have a different name. The position of the molecule can be specified either in Cartesian coordinates (x, y, z) or in polar coordinates (r, θ, ϕ) , and is given relative to the global coordinate system in either case. The coordinates may be literal numerical values, or the names of constants or variables previously defined. It is common to vary the position and orientation of molecules in a cluster to optimize the energy, so if these are given as literal numerical values they are treated as the initial values of anonymous variables, rather than constants.

The molecule attributes describe properties of the molecule as a whole, while the base site attributes describe the site at the molecular origin. They must all be given on the same logical line as the MOLECULE command. The possible molecule attributes are

LINEAR

This molecule is linear. The data for the individual sites take a slightly different form in this case; see below.

```
ROTATED { α β γ | BY χ ABOUT l m n }
```

The local molecular coordinate frame is rotated relative to the global coordinate frame. The orientation of the molecule is specified either by giving a set of Euler angles (α, β, γ) , or in terms of a rotation through an angle χ about a vector (l, m, n) in the global frame. (It need not be a unit vector.) The ROTATED attribute must be specified if an optimization is to be carried out, even if the rotation is initially zero.

```
COPY name
```

If COPY *name* is specified, all details of the molecular sites are copied from molecule *name*. No END line is needed.

The possible base site attributes are:

RANK *L*

Normally the molecular origin is a site of rank -1 , i.e. it is assumed to have no electrostatic moments of its own, even if the default rank for other sites has some other value. If the base site is required to carry multipole moments, an explicit RANK declaration will be needed, and the multipole moments then follow as for any other site (see below).

TYPE *type-name*

This specifies that the site is of type *type-name*, as defined by a TYPE command elsewhere in the data. A site must be assigned a type if it is to take part in repulsion, dispersion or charge-transfer interactions.

6.1.2 Site definitions

Each site definition for a nonlinear molecule takes the form

```
name AT { x y z | POLAR r θ φ } [site attributes]  
site multipoles, if any
```

The name of the site is any convenient character string; if it contains spaces it must be enclosed in single or double quotes. Site names are not required to be unique, though they may need to be unique within a particular molecule to prevent ambiguity in subsequent references. There is no need for sites in different molecules to have different names. The site position can be specified either in Cartesian coordinates (x, y, z) or in polar coordinates (r, θ, ϕ). The site attributes follow on the same logical line. They can include any of the following, in any order:

RELATIVE TO *parent-name*

By default, the site position and orientation are given in the molecule's local coordinate system, as specified in the MOLECULE declaration. That is, if no parent is specified explicitly, the parent site is the molecular origin. This attribute asserts that they are given in the local coordinate system of the specified parent site, which must be in the same molecule and must have been defined already. This can be used, for example, to define atom site positions in terms of bond lengths and bond angles.

LINK TO *parent-name*

Here the position and orientation are taken to be given in the molecular coordinate frame, but are then converted to a local reference relative to the specified parent. This allows crystallographic or *ab initio* data to be used for defining a reference geometry, while still allowing internal rotations to be specified.

ROTATED { $\alpha \beta \gamma$ | BY χ ABOUT $l m n$ }

The local coordinate frame of this site is rotated as specified relative to the coordinate frame of its parent.

TWIST ω

A twist is a rotation about the axis to the site in question from its parent, and is intended for specifying dihedral angles. A rotation may be specified by 'ROTATED ...' or by 'TWIST ...' or both. If both a rotation and a twist are specified, the rotation is considered to be applied first, to give a reference orientation, and the twist second. A positive rotation is taken according to the usual convention, i.e. corkscrew-wise about an axis from the parent site to this one.

TYPE *type-name*

RANK *L*

These attributes have the same form as for the molecule's base site. The type is undefined by default. The default rank is as specified in a previous DEFAULT RANK declaration, if any; otherwise it is -1 (no multipoles), and in this case an explicit rank must be specified for any site that is to carry multipoles.

BOND TO *name* [AND *name*]

This specifies actual chemical bonds. The site at the other end of each bond must have been defined already. This information is not normally required, but it is used in constructing a ball-and-stick display.

The EDIT command (p. 20) provides simpler methods for specifying chemical bonds, if they are needed.

```

Molecule Formaldehyde at x1 y1 z1 rotated alpha1 beta1 gamma1
C at 0.0 0.0 0.0 type C rank 4
  0.655041
  0.354816 0.0 0.0
  0.025542 0.0 0.0 -0.385910 0.0
 -0.595851 0.0 0.0 2.346798 0.0 0.0 0.0
  2.172953 0.0 0.0 -0.568807 0.0 0.0 0.0 -0.149474 0.0

O 0.0 0.0 C-H type O rank 4
 -0.581386
  0.020940 0.0 0.0
  0.326349 0.0 0.0 -0.276484 0.0
 -0.200280 0.0 0.0 -0.829297 0.0 0.0 0.0
  0.224333 0.0 0.0 1.196370 0.0 0.0 0.0 0.016740 0.0

H1 pol C-H O-C-H 180 type H rank 1
  Q00 = -0.036827 Q10 = -0.121165 Q11c = -0.199957

H2 pol C-H O-C-H 0 type H rank 1
  Q00 = -0.036827 Q10 = -0.121165 Q11c = 0.199957

Axes z from C to H1 x from H1 to H2 rotate for H1
Axes z from C to H2 x from H2 to H1 rotate for H2

End

```

Figure 3: Molecule definition for formaldehyde.

If there are multipole moments for this site (that is, if its rank is 0 or higher) they follow the site attributes, starting on a new line. Two alternative formats may be used. The numerical values can all be given in the order Q_{00} , Q_{10} , Q_{11c} , Q_{11s} , Q_{20} , ..., Q_{LLc} , Q_{LLs} . Alternatively, the moments may be given in the form $Q_{00} [=] \textit{value}$ $Q_{10} [=] \textit{value}$... In this case, any zero moments may be omitted, and the non-zero moments may be given in any order, but the list must be terminated by a blank line. In either case, several values can be given on each line, and the list may continue on as many lines as necessary. Multipole moments are always given in atomic units.

An alternative way to define site orientations, which avoids the need to puzzle over Euler angles and so forth, is to use the following command, which can occur anywhere in a molecule definition after the definition of the sites to which it refers:

```
AXES axis1 axis2 [ ROTATE | DEFINE ] FOR site site . . .
```

where *axis1* specifies the direction of one local axis, in one of the forms

```
{x|y|z} FROM site1 TO site2
```

or

```
{x|y|z} FROM site1 BETWEEN site2 AND site3
```

In the second case the axis concerned bisects the angle between the *site1-site2* and *site1-site3* directions.

axis2 specifies the second axis in the same way, except that it is then orthogonalized to the first axis. The remaining axis forms a right-handed set.

If this approach is used, any rotation previously specified for the sites whose axes are being defined is ignored, but if a twist is specified it is applied to the newly-defined axes.

If ROTATE is specified, any multipole moments given for the sites whose axes have been changed are rotated into the new axes. This is the usual case, where distributed multipoles have been calculated ab initio and are all given in global axes, and it is the default. DEFINE, on the other hand, means that the multipole moments are already given relative to the rotated axis system.

Fig. 3 shows a molecule definition for formaldehyde. It uses the constants C-O, C-H and O-C-H for the bond lengths and bond angles, and variables x_1 , y_1 , z_1 , α_1 , β_1 , and γ_1 for the position and orientation. They must have been defined previously. It also shows the two alternative ways to give

the multipole moments, and local axes are defined for each H atom with the z axis along the C–H bond and the x axis perpendicular to the bond and in the plane of the molecule. The H atom multipole moments are given with respect to global axes, but will be rotated into the new axis system. The program prints the rotated multipole moments.

Fig. 4 shows a more complicated example. Here the three methyl groups are assumed to have local C_{3v} symmetry — probably not a good approximation, and made here just for illustrative purposes. In local axes with z along the local C_3 axis, the C atom moments Q_{lm} are zero unless $m = 0 \pmod{3}$. Moreover, if the H atom local axes are all defined with z along the CH bond and x in the NCH plane, the local moments are the same for every H atom.

6.2 Linear molecules

For a linear molecule, the molecule definition should include the LINEAR keyword, and the site specifications take the simpler form

```
name AT z [RELATIVE TO parent-name] [ROTATED] [TYPE type-name] [RANK L]
site multipoles, if any
```

The only function of a RELATIVE clause in this case is to permit geometries to be specified in terms of bond lengths rather than positions relative to the molecular origin. For a linear molecule, only the moments Q_{00} , Q_{10} , Q_{20} , ..., Q_{L0} should be given. Sites in a linear molecule may be rotated, but the only rotation that makes sense is one that reverses the direction of the local z axis, *i.e.*, one with Euler angles $(0, \pi, 0)$. Accordingly the ROTATED keyword on its own is sufficient, and is assumed to specify this particular rotation. Fig. 5 shows a molecule definition for acetylene and the atom positions and local axes that result.

There are several points to note in this definition. First, CC/2 is the name of a constant specifying half the C–C bond length; names generally may include any character, and the ‘/’ doesn’t imply division. (The use of names like this is not recommended in general, because of the possibility of confusion.) There is one exception to this, however: a minus sign before a name, as in -CC/2, negates the constant. The second point to note is that atom C2 is rotated, so its local z axis points in the opposite direction to the global Z axis, and its dipole moment too points in the opposite direction to that of C1. Moreover, H2 is defined relative to C2, so its position is a distance CH along the local z axis from C2, and its local axes are in the same direction as those of C2.

6.3 Atoms

```
ATOM name AT {x y z | POLAR r  $\theta$   $\phi$  } [TYPE type-name] [RANK 0]
```

This declaration defines an atom called ‘name’ at the position specified. The definition of an atom is similar to that of a molecule, but no further sites are allowed and no rotation may be specified. The only multipole that may be defined for an atom is the charge; if this is required the rank needs to be defined as 0, since the default is -1 . A site type may be specified, and will usually be needed in order that repulsion/dispersion interactions can be defined. None of the other site options is relevant for an atom, but polarizabilities may be defined subsequently (see below). An atom specification must be followed by END as for a molecule.

6.4 Lattice

A lattice, like a molecule, comprises a number of sites, but in most respects it is treated differently. Sites are specified for the lattice unit cell, and are considered to be replicated by translation along the specified lattice vectors. The lattice vectors \mathbf{a} and \mathbf{b} are restricted to lie in the global xy plane. The unit cells are replicated to $\pm\infty$ in the \mathbf{a} and \mathbf{b} directions, but to $-\infty$ only in the \mathbf{c} direction. Consequently there are no lattice sites with $z > 0$, and the plane $z = 0$ is a crystal surface onto which molecules can be adsorbed.

```

Constants
  CN      1.47 A
  zNC    104 D
  C-H    1.09 A
  N-H    1.011 A
  N-C-H  109.4 D
End
Molecule Trimethylamine at x1 y1 z1 rotated alpha1 beta1 gamma1

N at 0.0 0.0 0.0 type N rank 3
  Q00 = -0.328424  Q10 = -0.519333  Q20 = -0.364515
  Q30 = -1.022369  Q33c = -0.949586

Note Define the C atom of a methyl group, rotated so that its z axis
Note is along the NC bond. The x axis will still be in the zNC plane
Note following the rotation. In this orientation, many of the moments
Note of the methyl group are zero (approximately) because of the
Note local C3v symmetry.
C1 polar CN zNC 0.0 relative to N type C rank 3 rotated 0.0 zNC 0.0
  Q00 = -0.036161  Q10 = -0.397953  Q20 = 0.731520
  Q30 = 1.515602  Q33c = -2.073542

Note Now the H atoms are defined relative to the C atom, but with the
Note local coordinates further rotated so that the local H-atom z
Note axis is along the CH bond in each case. If we continue to assume
Note C3v symmetry for the methyl group, each H atom has the same
Note moments in its local axis system
H11 polar C-H N-C-H 0.0 relative to C1 rotated 0 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

H12 polar C-H N-C-H 120.0 relative to C1 rotated 120 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

H12 polar C-H N-C-H 240.0 relative to C1 rotated 240 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912
Note The C atom of the second methyl group is defined in a similar
Note way. In these local axes the moments are the same as for C1.
C2 polar CN zNC 120.0 relative to N type C rank 3 rotated 120.0 zNC 0.0
  Q00 = -0.036161  Q10 = -0.397953  Q20 = 0.731520
  Q30 = 1.515602  Q33c = -2.073542

Note The definition of the hydrogen atoms is also the same except for
Note the labels.
H21 polar C-H N-C-H 0.0 relative to C2 rotated 0 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

H22 polar C-H N-C-H 120.0 relative to C2 rotated 120 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

H22 polar C-H N-C-H 240.0 relative to C2 rotated 240 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

Note Similarly for the third methyl group
C3 polar CN zNC 120.0 relative to N type C rank 3 rotated 120.0 zNC 0.0
  Q00 = -0.036161  Q10 = -0.397953  Q20 = 0.731520
  Q30 = 1.515602  Q33c = -2.073542

H31 polar C-H N-C-H 0.0 relative to C3 rotated 0 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

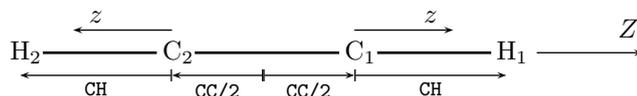
H32 polar C-H N-C-H 120.0 relative to C3 rotated 120 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

H32 polar C-H N-C-H 240.0 relative to C3 rotated 240 N-C-H 0 type H rank 1
  Q00 = 0.048545  Q10 = 0.106785  Q11c = 0.045912

End

```

Figure 4: Molecule definition for trimethylamine.



```

Constants
  CC/2  0.601 A
  CH    1.060 A
End
Molecule Acetylene at 0.0 0.0 0.0 linear
C1 at CC/2 rank 2
Q00 = -0.023753392   Q10 = 0.8135023887   Q20 = -1.4862906534

C2 at -CC/2 rank 2 rotated
Q00 = -0.023753392   Q10 = 0.8135023887   Q20 = -1.4862906534

H1 at CH relative to C1 rank 1
Q00 = 0.023753392    Q10 = 0.3095037496

H2 at CH relative to C2 rank 1
Q00 = 0.023753392    Q10 = 0.3095037496

End

```

Figure 5: Molecule definition for acetylene.

The crystal lattice, if required, is defined by a declaration of the form

```

LATTICE
  options
CELL
  A  ax ay
  B  bx by
  C  cx cy cz
END
SITES
  :
  site definitions
  :
END
END

```

The vectors **a**, **b** and **c** are the lattice vectors of the lattice unit cell. **a** and **b** must lie in the global *xy* plane. It may be necessary to redefine the unit cell of the crystal to achieve this. The sites of the lattice cell follow in the same format as for a molecule, except that the site positions are always given in global coordinates, and they can be specified in the form `FRAC fa fb fc`, where *f_a*, *f_b* and *f_c* are fractional displacements along the cell vectors.

Once the lattice has been defined, it may not be changed, but the `LATTICE` command may be repeated without the cell and site definitions to change the options. The options are:

```
RANGE DIRECT s RECIPROCAL g
```

`RANGE` specifies the range to which summations are carried out. Summations in direct space (which normally include the dispersion and repulsion energy as well as the electrostatic energy) are taken to a distance *s* (in the current length units), while summations in reciprocal space for the electrostatic interaction are taken to a distance *g* in reciprocal space. The default values are *s* = 40 bohr, *g* = 6.

```
EWALD n
NOEWALD
```

`EWALD n` specifies that an Ewald summation is to be used. The interactions between the adsorbed molecules and the surface are calculated directly for the lattice cells with centre of mass up to a distance *s* away from the global origin (0,0,0). The electrostatic interactions are calculated using distributed multipoles in the usual way. If an Ewald summation is required, it is carried out using the total moments of molecule and lattice cell, referred to their respective centres of mass. The interactions between these moments are evaluated up to terms in R^{-n} . From the result are subtracted the interactions between

```

Lattice
  Range direct 25.0 reciprocal 4
  Ewald 5
  Cell
    a 5.627 0.0      (Lattice parameter 5.627 A)
    b 0.0 5.627
    c 0.0 0.0 5.627
  End
  Sites
  Na  frac   0.0  0.0  0.0  Rank 0  Type Na
  1.0
  Na  frac   0.5  0.5  0.0  Rank 0  Type Na
  1.0
  Na  frac   0.5  0.0 -0.5  Rank 0  Type Na
  1.0
  Na  frac   0.0  0.5 -0.5  Rank 0  Type Na
  1.0
  Cl  frac   0.5  0.0  0.0  Rank 0  Type Cl
 -1.0
  Cl  frac   0.0  0.5  0.0  Rank 0  Type Cl
 -1.0
  Cl  frac   0.0  0.0 -0.5  Rank 0  Type Cl
 -1.0
  Cl  frac   0.5  0.5 -0.5  Rank 0  Type Cl
 -1.0
  End
End

```

Figure 6: Lattice definition for sodium chloride.

centre-of-mass moments involving those lattice cells that have already been evaluated in direct space. The default is EWALD 4, which includes interactions between the net molecular dipoles and the lattice cell quadrupole. (Note that the lattice cell must be defined in such a way that it has no net dipole moment; otherwise there is a contribution from the surface charges on the crystal which does not vanish as the crystal is extended to infinity.) NOEWALD specifies that no Ewald sum is to be carried out; it is equivalent to EWALD 0.

DIELECTRIC ϵ

Here ϵ is a dielectric constant to be applied to interactions between the lattice and the molecules. It provides a crude way to account for polarization effects in the crystal, in the absence at present of a proper treatment.

The example in Fig. 6 shows a lattice definition for sodium chloride.

6.5 Layers

If a lattice and a set of molecules is defined, it is assumed that the molecules are to be treated as an isolated cluster on the crystal surface. The LAYER or MONOLAYER or ADLAYER declaration (the keywords are equivalent) specifies that the set of molecules is to be replicated across the surface to form an infinite layer.

```

LAYER
  options
  CELL
    A  $a_x$   $a_y$ 
    B  $b_x$   $b_y$ 
  END
END

```

The vectors \mathbf{a} and \mathbf{b} , which must lie in the xy plane, define the 2-dimensional unit cell of the surface layer. If there is an underlying crystal, the layer must be commensurate with the crystal lattice, and in this case \mathbf{a} and \mathbf{b} are defined in the basis of the unit cell vectors for the underlying lattice. For the layer to be commensurate with the lattice, the components of \mathbf{a} and \mathbf{b} must then be integers. For example, Fig. 7 defines the layer unit cell shown when the crystal lattice vectors are \mathbf{a} and \mathbf{b} .

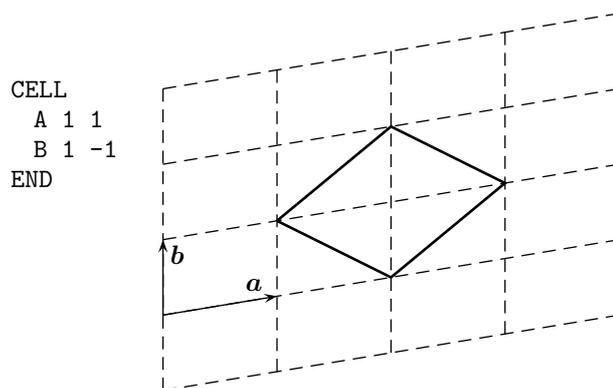


Figure 7: Example of the definition of an adlayer unit cell

If there is no crystal lattice, there is no restriction on the unit cell of the layer, and the components are just lengths in the current unit.

The options include RANGE and EWALD, as for the LATTICE declaration. The default range for real-space is 40 bohr, but the range for the reciprocal-lattice summation is determined by the program. The default for Ewald summation is EWALD 3 in this case, as it is usually necessary to carry out an Ewald summation of the dipole–dipole interaction. If NOEWALD or EWALD 0 is specified, the Ewald summation is skipped.

If there is more than one molecule per unit cell in the adlayer, the interaction energy between them is calculated by one of two routines, one of which is suitable when the difference in global z coordinate between the centres of mass is small, and one when it is large. The switch between the two is controlled by an additional option to the EWALD subcommand, namely SWITCH z . The default is to change at $z = 0.25$ bohr. The two routines will give slightly different results because of differing convergence characteristics, which may cause optimizations to thrash between them, in which case it would be helpful to change the switch value.

The LAYER command may be repeated to change the options during the course of a calculation, though the unit cell may not be redefined. Any options not mentioned remain unchanged. There is a DEFAULTS option, which resets default values.

6.6 Crystal growth and aggregation

Many theories of crystal growth are based on calculations of the attachment energy, which for an (hkl) crystal face is the energy per molecule released when a slice of crystal parallel to (hkl) is added to the growing face. The thickness of the slice depends on the indices (hkl) ; for (100) , for example, it would be 1 unit cell thick in the a crystal direction, while for (200) it would be half a unit cell thick (which may or may not be meaningful in this context). This approach uses static energy calculations; it is assumed that a large attachment energy is associated with fast growth. Fast-growing faces generally grow out, so that the faces seen in actual crystals are the slowest-growing ones.

The ORIENT program can calculate the attachment energy between a crystal slice and the bulk crystal. Because of the periodicity of the system, it is only necessary to calculate the interaction energy between the bulk crystal surface and a single unit cell of the slice.

It is also possible to determine the interaction energy between two different crystal faces. This is treated approximately by considering the interaction between an (hkl) crystal surface and a ‘nanocrystal’ with an exposed $(h'k'l')$ face in contact with the (hkl) surface. In this case the position of the nanocrystal, and its orientation with respect to the surface normal, are not known as they are for the attachment energy, so they have to be determined by optimizing the energy. Moreover the optimum geometry may depend on the size of the nanocrystal, though it rapidly settles down as the size of the nanocrystal is increased.

In both types of calculation a lattice must be defined as described above. The nanocrystal is defined by a declaration of the form

```
NANOCRYSTAL
  CELL
    A  $a_x$   $a_y$ 
    B  $b_x$   $b_y$ 
    C  $c_x$   $c_y$   $c_z$ 
  END
  [ SIZE A  $n_a$  B  $n_b$  C  $n_c$  ]
END
```

The atoms in the unit cell are defined as for an ordinary molecule. If the attachment energy is to be calculated the CELL definition for the nanocrystal should be the same as in the LATTICE command. SIZE, if present, specifies the number of unit cells in the nanocrystal. If it is omitted, a single cell is assumed, i.e. $n_a = n_b = n_c = 1$.

7 Defining site properties and site-site interactions

7.1 Site types

```
TYPES
  type-name [Z atomic-number] [RADIUS r] [MASS m] +++
  [LABEL name | NOLABEL] [COLOUR data]
  :
END
```

type-name is a character string, enclosed in single or double quotes if it contains spaces. *r* is the van der Waals radius for this type of site. The mass is needed for optimizations, and for Ewald summations, where the total moments used in the summation are referred to the centre of mass of the molecule or lattice cell. The radius must be specified if a hard-sphere repulsion is required, or if it is needed to define the molecular surface, for example to tabulate the electrostatic potential on the surface. The atomic number may be specified as shown. If it is given, a standard mass and van der Waals radius are used for the site, unless explicit values are given. ORIENT knows the masses of the most common isotopes of atoms up to $Z = 36$ (krypton) except in a few cases where it uses the mean atomic weight. The program knows values of some van der Waals radii for atomic numbers up to 82, as listed by [Bondi \(1964\)](#). For atoms for which values are not given by [Bondi](#) the standard radius is arbitrarily taken to be 2.5 bohr.

A colour may be specified, but is used only for graphical displays (see §18). It can be expressed in any of the forms illustrated by the following examples:

```
COLOUR GREY 0.8
COLOUR RGB 0.5 0.25 1.0
COLOUR RGB255 128 64 255
COLOUR RGBX 8040FF
```

The last three of these are equivalent. COLOR is accepted as a synonym for COLOUR.

A label, if given, is used only to identify the atom type in xyz files produced by the program; otherwise, if the atomic number is given the standard chemical symbol is used for this purpose, and failing that the first four characters of the type name are used. If NOLABEL is specified, atoms of this type do not appear in xyz files. If the atomic number is omitted or given as zero, the mass and radius are assumed to be zero and NOLABEL is assumed, unless explicit MASS or RADIUS or LABEL values are given. For example, the following defines types C and H straightforwardly as carbon and hydrogen atoms, shown as such in xyz files, while cm (presumably the centre of mass, and perhaps the interaction site for dispersion interactions) has no mass of its own and is not shown in xyz files:

```
Types
  C Z 6
  H Z 1
  cm Z 0
End
```

Up to 20 site types may be defined, unless the limit is increased in the ALLOCATE command.

7.2 Defining the interactions between sites

There is always an electrostatic interaction between sites on different molecules that have any non-zero multipole moments. Other interactions can be added through the PAIRS declaration, where the parameters for the interaction are also defined. The repulsion potential takes the form $U \exp[-\alpha(r-\rho)]$, where U is a constant, normally 10^{-3} Hartree. Dispersion coefficients C_n may be specified for $n \leq 12$.

The orientation-dependent quantities α , ρ and the C_n coefficients are expressed as \bar{S} -function expansions (Stone, 1996). For example:

$$\rho = \sum_{l_a \kappa_a l_b \kappa_b j} \rho_{l_a l_b}^{\kappa_a \kappa_b j} \bar{S}_{l_a l_b}^{\kappa_a \kappa_b j}.$$

At present the permitted \bar{S} -functions are limited to those for which $l_a + l_b = j$ (i.e., those that occur in the electrostatic energy) except for a few special cases, in particular when $l_a = l_b = 1$ and $j = 0$.

The form of the specification is

```
PAIRS
[ general options ]
t1 t2 [ALPHA] [RHO] [C6] [GAMMA6] [C7] [GAMMA7] ...
[ options for this pair ]
:
l_a \kappa_a l_b \kappa_b j [\alpha] [\rho] [C6] [\gamma6] [C7] [\gamma7] [C8] [\gamma8]...
:
End
etc.
END
```

Here t_1 and t_2 are the names of the two site types concerned. Following them is a list of the parameters that will be specified subsequently. They may be listed in any order, but the numbers in subsequent lines must be in the same order. Subsequent lines specify the terms in the \bar{S} -function expansions of each parameter. In these lines, l_a and κ_a are specified together as a character string of the form '00', '20', '22s', etc., and l_b and κ_b likewise. The corresponding expansion parameters follow. Omitted values at the end of the line are taken to be zero, so if for example ρ is anisotropic but α and C_6 are isotropic, it is more convenient to give the ρ values first. The dispersion anisotropies may be specified either in terms of the anisotropic coefficients, or in terms of the anisotropy parameters γ , defined by $(C_n)_{l_a l_b}^{\kappa_a \kappa_b j} = (\gamma_n)_{l_a l_b}^{\kappa_a \kappa_b j} (C_n)_{000}^{00}$. In this case the value provided for $(\gamma_n)_{000}^{00}$ is taken to be $(C_n)_{000}^{00}$. Note that the isotropic coefficients are positive for attractive interactions.

The maximum number of pair interactions is normally 100, but the limit can be increased by a PAIRS declaration in the ALLOCATE command. (See p. 4.) If we call the set of pair interaction parameters associated with a particular \bar{S} -function a parameter-set, the maximum number of parameter-sets, for all site pairs, is normally limited to 200, but this limit may be increased by a PARAMETER-SETS declaration in the ALLOCATE command.

Note that the pair $A \dots B$ is treated by the program as different from $B \dots A$. Only one of them needs to be defined in the data file, and the other will be created automatically. However both need to be counted when working out the total number of pairs, and their parameter-sets also have to be counted twice.

The following options may occur in a PAIRS section, either before a pair definition, in which case they supply defaults for subsequent pairs and also for undefined interactions, or inside a pair definition, in which case they supply values for that pair only.

```
UNIT u
```

UNIT u defines the pre-exponential factor to be u in the current energy units. The default is 10^{-3} Hartree.

```
[ INDUCTION | DISPERSION ] DAMPING type [ parameters ]
```

This option defines the treatment of damping. If either INDUCTION or DISPERSION is specified, the treatment applies to that type of interaction only, and otherwise to both. DAMPING OFF is the default.

Several types of damping function can be used. The most usual is Tang–Toennies, but there are several varieties. In all of them, the damping function is a Tang–Toennies damping function $f_n(bR)$, *i.e.*, an incomplete Gamma function of order $n + 1$, but the coefficient b can take various forms. In all of the following, “TANG–TOENNIES” can be abbreviated to “TT”.

DAMPING TANG–TOENNIES [b]

Here b is an optional constant. It has dimensions of reciprocal length and is assumed to be given in the current units. If it is omitted, b is set equal to the alpha value in the repulsion for the same pair of sites, in which case it will usually be different for different site pairs. The dispersion and repulsion parameters do not need to be given in the same PAIRS command.

DAMPING TANG–TOENNIES–SLATER [b]

DAMPING TANG–TOENNIES–SCA [b]

These are versions of the modified Tang–Toennies damping function (Tang and Toennies, 1992), which is recommended when the repulsion energy V_{er} is not a pure exponential. In this case, b depends on the site separation R , according to

$$b(R) = -\frac{\partial}{\partial R}(\ln V_{er}).$$

Orient can handle two such cases. The Tang–Toennies–Slater version is derived from

$$V_{er} = K((bR)^2 + 3bR + 3)\exp(-bR),$$

while the Tang–Toennies–SCA version is derived from the asymptotic form of the exchange–repulsion obtained by Smirnov and Chibisov (1965) and Andreev (1973), namely

$$V_{er} = KR^{(7/b)-1}\exp(-bR).$$

In both cases the parameter b can be specified explicitly, or if omitted is taken from the alpha parameter of the repulsion for the same site pair.

Other damping functions are provided but are less commonly used. HFD s and KMA s specify Hartree–Fock–Dispersion or Koide–Meath–Allnatt damping functions respectively, and ERF s specifies that the damping function is an error function. (The last of these is used in the Sprik–Klein water potential, but is probably not of much use otherwise.) DAMPING SBCS specifies the damping function used by Szczęśniak *et al.* (1990). In each case but the last a scale factor s is needed; for the SBCS damping function the scale factor is the sum of the van der Waals radii.

The remaining damping functions need more parameters. They are

DAMPING GE b c t

DAMPING GG b c t

DAMPING PADE m b_m b_{m-2} ... b_2 a_{m-6} a_{m-4} ... a_2

These are the new damping functions described by Hodges and Stone (2000). The first two are the exponential and gaussian combination damping functions, which are

$$g_e(R) = [1 + c\exp(-bR) - (1 + c)\exp(-tbR)]^6,$$

$$g_g(R) = [1 + c\exp(-bR^2) - (1 + c)\exp(-tbR^2)]^3.$$

The third is the g_m function

$$g_m(R) = \sum_{i=0}^{m-6} a_i R^{-i} / \sum_{i=0}^m b_i R^{-i}$$

with a_i and b_i zero for odd i .

Note that induction damping is applied only to pairs of sites for which damping has been defined. For consistency, the same damping should be used for all interactions involving a polarizable site and

```

PAIRS
  CH N          rho  alpha
    00 00 0    6.37  1.930
    10 00 1    0.098
    00 10 1   -0.075
    20 00 2    0.185
    00 20 2    0.107
    30 00 3    0.270
  end
  CH CH        rho  alpha
    00 00 0    6.83  1.819
    10 00 1    0.098
    20 00 2    0.185
    30 00 3    0.270
  end
  N N          rho  alpha
    00 00 0    5.91  2.055
    10 00 1    0.098
    20 00 2    0.185
  end
End

```

Figure 8: Example of a definition of pair interactions.

some other site carrying multipoles. To achieve this it is sufficient to define the induction damping as a general option rather than for particular pairs of site types.

An example of an anisotropic repulsion specification is shown in Fig. 8, for acetylene and N_2 . The repulsion coefficients are from Stone and Tong (1994). α is isotropic here. In this example the program automatically sets up appropriate data for the $N\cdots CH$ interaction too; it is not necessary to define it explicitly. Similarly the program automatically includes 00 10 1 and 00 20 2 parameters for the $N\cdots N$ interaction, with the same values as for the 10 00 1 and 20 00 2 terms given.

The units for all of the quantities in a PAIRS command are those implied by the units currently defined for length and energy (see the UNITS command). Note that α has units of reciprocal length.

In this version of the program the Lennard-Jones potential is not provided, but it can be fudged by providing suitable C_6 and C_{12} coefficients. If you do this, note that the C_6 coefficient should be positive and the C_{12} coefficient negative. Both of these terms are currently regarded as dispersion, so if the dispersion energy comes out positive, don't be alarmed. Charge-transfer can be described by an attractive exponential term, *i.e.*, by defining the pre-exponential factor for the pair interaction to be negative using UNIT. Only one kind of 'repulsion' interaction can be defined for each pair of site types, however, so if there is repulsion as well as charge transfer, you will need to define several sites of different types at the same position. Morse potentials can be defined in a similar way, as a sum of a repulsive exponential and an attractive one. There is no separate hard-sphere potential, but a pseudo-hard-sphere model can be obtained by using a large value of α . 25 bohr⁻¹ should be sufficient; higher values may cause problems with the optimization, because of the steepness of the potential. For example,

```

PAIRS
  C N          alpha
    00 00 0   25.0
  end
End

```

The same result can be achieved for all pairs using the syntax

```

PAIRS
  [UNIT unit]
  HARD-SPHERE [ $\alpha$ ]
End

```

The value of α specified here is used for repulsions between every pair of types, provided that both types in the pair have non-zero radii. If it is omitted, a value of 25.0 is used. The value of the pre-exponential

constant may be adjusted using the syntax shown; the standard value of 0.001 hartree gives separations that are slightly too large, and a value of 0.0005 hartree or thereabouts may be better.

This example assumes that the site radii have been previously defined using the TYPES declaration, in which case ρ is the sum of the site radii. However ρ can be defined in the pairs declaration for a hard-sphere potential just as for an ordinary exponential potential. Indeed it *is* an ordinary exponential potential except for the unusually large value of α . Notice that an anisotropic ‘hard-sphere’ potential is easily obtained by defining the ρ expansion appropriately.

It is permissible to use several PAIRS commands referring to the same pair of sites, so that if, for example, the ρ and α values are known in atomic units and the C_6 coefficients in $\text{\AA}^6 \text{kJ mol}^{-1}$, two separate PAIRS commands may be used, with a UNITS command between them to change units appropriately.

8 Modifying a molecule description

The EDIT command provides various ways of modifying the description of a molecule after it has been read in. It is not very often needed; the most common situation where it may be useful arises when a Distributed Multipole Analysis has been carried out by an *ab initio* program such as CAMCASP or MOLPRO. In such cases the multipoles for each site are all expressed in the global coordinate system, and it may be necessary to re-express them in suitable local coordinate systems, possibly different for each site. Another possibility is that the rank of multipole on some sites may need to be limited, and multipoles above the limiting rank transferred to other sites.

The command takes the form

```
EDIT [molecule]  
  ⋮  
  subcommands  
  ⋮  
END
```

The subcommands are:

```
MOLECULE molecule
```

Subsequent commands apply to the specified molecule. If no molecule is specified on the EDIT line, this command must appear before anything else. It can be used more than once, to switch between molecules.

```
VERBOSE
```

Print more details.

```
QUIET
```

Print fewer details. (Default)

```
AXES  
   $s_0$  {x|y|z} [ FROM  $s_1$  ] TO  $s_2$  {x|y|z} [ FROM  $s_3$  ] TO  $s_4$   
  ⋮  
END
```

Define local axes for site s_0 . The first axis mentioned is in the direction from s_1 to s_2 ; the second axis is initially from s_3 to s_4 but is Schmidt orthogonalised to the first axis. The third axis forms a right-handed set with the first two. If the FROM part is omitted, FROM s_0 is assumed. The sites are referenced by name, so the names need to be unique.

An alternative specification that can be used for either axis is

```
{x|y|z} FROM  $s_1$  BETWEEN  $s_2$  AND  $s_3$ 
```

In this case the axis concerned bisects the angle between the s_1 – s_2 and s_1 – s_3 directions.

In linear molecules, there are no reference points to define directions perpendicular to the molecular axis, and the perpendicular axes are arbitrary anyway. If the molecule has been declared linear in its definition, only the local z axis of a site may be redefined. If this is done, the local x and y axes are

defined to form an orthogonal right-handed set but are arbitrary. If the molecule has not been defined as linear, even though it is, a local x axis may be defined as 'GLOBAL { X | -X | Y | -Y | Z | -Z }', i.e. parallel or antiparallel to the global x , y or z axis. It will be orthogonalized to the local z axis in the usual way.

```
BONDS AUTO
BONDS s1 -> s2 [& s3 ... & sn] [-> s'2 [& s'3 ... & s'n]] ...
```

The first form assigns chemical bonds automatically on the basis of covalent radii. The second form describes the bonding explicitly: site s_1 is bonded to sites s_2, s_3, \dots, s_n ; then the last of these, s_n , is bonded to sites s'_2, s'_3 , etc., and so on. The command can be repeated to specify further bonds. For example, the bonds in propane, with a natural atom labelling, could be specified by

```
BONDS C1 -> H11 & H12 & H13 & C2 -> H21 & H22 & C3
BONDS C3 -> H31 & H32 & H33
```

```
SHIFT FROM s1 TO s2 [RANK n]
```

Shift multipoles of ranks n and above from s_1 to s_2 . This is done in such a way that the overall multipoles of the molecule are preserved. However the accuracy of the multipole description will be reduced if multipoles are shifted too far. It is often useful to shift multipoles on hydrogen atoms to the neighbouring heavy atom to simplify the multipole description of the molecule; for hydrogen atoms on polar groups such as OH it is advisable to leave the charge and perhaps the dipole on the hydrogen and to shift only the higher moments.

In the following commands, the action specified is applied to all sites with the names given:

```
DELETE s1 s2 ...
```

Remove the specified sites altogether.

```
CLEAR s1 s2 ...
```

Set the multipoles to zero for the specified sites.

```
LIMIT [RANK] j [FOR] s1 s2 ...
```

Ignore multipoles above rank j . They can be restored by a subsequent LIMIT command with a higher value of j .

```
SUPPRESS s1 s2 ...
```

Temporarily suppress all multipoles on the sites listed.

```
RESTORE s1 s2 ...
```

Restore multipoles previously suppressed.

```
NEGATE s1 s2 ...
```

Change the sign of all moments on the sites listed.

```
MOMENTS [FOR] s Qlm [=] ⟨Qlm⟩...
```

Set particular moments for site s explicitly. Moments not mentioned are left unchanged.

```
TYPE type [FOR] s1 s2 ...
```

Set or change the site type.

```
COMBINE molecule
```

Add the multipole moments of the specified molecule to those of the current molecule. Usually the moments of one molecule or the other will have been negated first, so that the result is to construct the difference. This provides a way to compare different multipole descriptions of the same molecule. The site names of the two molecules must match exactly — it is assumed that they are in fact different descriptions of the same molecule — so even if some sites in one description or the other carry no moments, the sites must still be present.

9 Distributed polarizabilities

A number of commands are provided for manipulating distributed polarizabilities. The molecule whose polarizabilities are to be specified must have been defined previously.

It is not necessary for polarizabilities to be specified for all sites in a molecule; for instance it may be convenient to specify hydrogen atoms separately as DMA sites but to include them with the neighbouring heavy atom for polarizability purposes. It is possible to specify polarizabilities for each atom of a molecule separately, but note that the program assumes that the mutual polarization of sites within the same molecule has been taken into account in the site-site polarizabilities, and it does not attempt to calculate any such effects, as would be required for instance in an Applequist model. Nor does the program attempt to take any account of the effects of molecular flexibility on polarizabilities, and only local polarizabilities should be used in flexible molecules.

Polarizabilities are specified or manipulated for one molecule at a time, using the command as follows:

```
POLARIZABILITIES FOR molecule
:
(subcommands)
:
END
```

The subcommands are:

VERBOSE	Print more details
QUIET	Print fewer details
READ	Read polarizability data
COPY	Copy polarizabilities from one site to another
DELETE	Delete polarizabilities
LIMIT	Suppress polarizabilities temporarily above a given rank
LOCALIZE	Calculate a set of localized polarizabilities
SUM-RULE	Check that the polarizabilities satisfy the charge-flow sum-rules.
MOVE	Move polarizabilities to a new site
TOTAL	Print the total molecular dipole–dipole polarizability
PRINT	Print specified polarizabilities
WRITE	Write the polarizabilities to a binary (unformatted) file.

The detailed syntax is as follows:

For the READ command, there are several possibilities. The syntax that is needed depends on the form of the polarizability data.

Often the polarizabilities are taken directly from the CAMCASP program. For versions 5.n of CAMCASP the syntax is

```
READ [LIMIT l] [FILE file] [SITES a b ... z]
```

where polarizabilities are to be provided for sites *a, b, ..., z*, all belonging to the molecule specified. If a file is specified, it should contain the polarizability data; otherwise the polarizabilities should follow this

command. In either case the polarizabilities are given on as many lines as necessary, in the order

$$\begin{aligned} &(((\alpha_{tu}^{pq}, \quad u = 00, 10, 11c, 11s, 20, \dots, lls), \\ &\quad p = a, b, \dots, z), \\ &\quad t = 00, 10, 11c, 11s, 20, \dots, lls), \\ &\quad q = a, b, \dots, z) \end{aligned}$$

Line breaks in the data are ignored. This is the data format used for the output from the distributed polarizability procedure in versions 5.n of the CAMCASP program, so the output from CAMCASP can be used directly as the input to ORIENT. The distributed polarizability file may also include labels in parentheses identifying the numbers; these labels are ignored by ORIENT and there is no need to edit them out of the file. If LIMIT L is specified, polarizabilities involving ranks higher than L are set to zero.

Another format is used in versions 5.n of CAMCASP, in particular from the localization procedure. The READ command takes the same form as above, except that no site list is given:

```
READ [RANK] L [FILE file]
```

If a file is specified, the polarizabilities are read from that; otherwise they are expected to follow the READ command. In either case the polarizabilities for each site pair a and b are given as a matrix:

$$\begin{array}{cc} a & b \\ \alpha_{00,00}^{ab} & \alpha_{00,10}^{ab} & \alpha_{00,11c}^{ab} & \alpha_{00,11s}^{ab} & \dots \\ \alpha_{10,00}^{ab} & \alpha_{10,10}^{ab} & \alpha_{10,11c}^{ab} & \alpha_{10,11s}^{ab} & \dots \\ \vdots & \vdots & \vdots & \vdots & \end{array}$$

The site pairs may be given in any order, and site pairs carrying no polarizabilities should be omitted. Note carefully that this array is not symmetric, in general, if $a \neq b$. The row labels should refer to site a and the column labels to site b . It is not necessary to provide α^{ba} as well as α^{ab} , but it does no harm to do so. The entire list of polarizable site pairs is terminated by a line containing only END, or by end of file if reading from a file.

In version 6.0 of ORIENT the format is different. In this case the polarizability matrix for each site pair is given separately, preceded by a header giving the names of the sites and other details. The format of the READ command in this case is

```
READ PAIRS [LIMIT L] FILE file
```

where the specified file contains the polarizabilities. Again, if LIMIT L is specified, polarizabilities involving ranks higher than L are set to zero.

Finally, it is possible to use a binary (unformatted) file previously written by the WRITE subcommand (see below) or by another program.

```
READ BINARY FILE filename [SITES] a b ... z
```

See the specification of the WRITE command for details of the file structure.

Note that the axes for each site are assumed to be the same as when the file was originally written. No axis information is contained in the binary file, so it is important to ensure that the axes are correctly defined before the polarizabilities are read.

```
COPY SITES a [b ... ] FROM [MOLECULE m] SITES a' [b' ... ]
```

Copy the polarizabilities already given for sites a' , b' , ..., possibly on another molecule m , to sites a , b , ... on the present molecule. The site names should identify the sites uniquely within each molecule. If polarizability matrices have not been defined for any site or pair of sites in the source list, the corresponding target matrix is left unchanged (which will probably mean that it is left undefined). The polarizabilities are not actually copied — the new sites refer to the same stored values as the old — so this is more efficient than specifying the new polarizabilities again using the READ command.

```
LIMIT [RANK] l [ FOR | SITES ] s1 s2 ...
```

Sometimes it is necessary to suppress the higher-rank polarizabilities and to perform calculations only with those up to rank l . This command suppresses temporarily the polarizabilities $\alpha(k, s, k', s')$ for which k or k' is of higher rank than l , for sites s listed and for all polarizable sites s' in the molecule. The original

rank can be reinstated by a further LIMIT command with a higher value of l , but if the polarizabilities have been manipulated, for example by LOCALIZE, this may not be a sensible thing to do.

SUM-RULE [TEST ϵ] [CORRECT]

Check that the charge-flow sum rules are satisfied to within the criterion ϵ . If no criterion is specified, a default value is used. If CORRECT is specified, the polarizabilities with $s = s'$ are adjusted so that the sum rules are satisfied. This should not be done unless it is known that the failures are due only to rounding errors.

LOCALIZE [[EXCEPT | ONLY] CHARGEFLOW] [TEST ϵ] +++
[ALL | SITES $a b \dots$ | PAIR $a b$ TO c]

Move polarizabilities between sites in such a way as to remove non-local polarizabilities α^{ab} with $a \neq b$. (See [Le Sueur and Stone \(1993, 1994\)](#).) If EXCEPT CHARGEFLOW is specified, charge-flow polarizabilities are left unchanged; if ONLY CHARGEFLOW is specified, only the charge-flow polarizabilities are localized.

A list of sites may be specified (and must come last); in this case, only non-local polarizabilities involving those sites or pairs of sites are removed. If no list is given, or if ALL is specified, all cross-polarizabilities are removed.

If PAIR $a b$ TO c is specified, the non-local polarizabilities α^{ab} and α^{ba} are replaced by local polarizabilities on site c , which need not already carry polarizabilities. The polarizabilities α^{aa} and α^{bb} are also changed by this procedure.

The localization is carried out in such a way that the sum rules for charge-flow polarizabilities are preserved. This is checked; if TEST ϵ is specified, the check is that the sum-rule value, which should be zero, is smaller than ϵ . The default value of ϵ is 10^{-7} ; a larger value may be used when it is known that the polarizability values are not very accurate.

In order for the localization procedure to work correctly, the rank of polarizabilities must be the same on every polarizable site. If a LIMIT command is used to reduce the rank of the polarizabilities before localization, it must be applied to all the sites.

MOVE [FROM] a [b] [CROSS] TO c [RANK [MIN l_{\min}] [KEEP l_{\max}]]

Replace the polarizabilities α^{ab} and α^{ba} , and possibly α^{aa} , α^{bb} , by equivalent polarizabilities at site c . If the optional keyword CROSS is present then only the non-local polarizabilities α^{ab} and α^{ba} are moved; if it is absent the polarizabilities α^{aa} , α^{bb} are moved too. Only polarizabilities of ranks l_{\min} and above are moved; polarizabilities of lower ranks are left unchanged. Polarizabilities at the new site above rank l_{\max} are discarded.

It is not possible to move charge-flow polarizabilities using this command. Either leave them where they are (using $l_{\min} \geq 1$) or use the LOCALIZE command first.

TOTAL [LIMIT l]

Print the total molecular dipole-dipole polarizability. If LIMIT 0 is specified, only the charge-flow polarizabilities are taken into account. Otherwise all local and non-local polarizabilities up to dipole-dipole are included.

DELETE [CROSS] s [s']

Delete polarizabilities $\alpha(k, s, k', s')$. The arguments have the same significance as in MOVE.

PRINT [LIMIT l_{\max}] [NON-LOCAL | LOCAL] $a b \dots z$

Print polarizabilities α_{tu}^{pq} for all p and q in the list specified. If LIMIT l_{\max} is specified, only the components up to rank l_{\max} are printed. If NON-LOCAL is specified, only non-local polarizabilities involving the listed sites are printed; if LOCAL is specified, only local polarizabilities are printed. These options may appear anywhere in the line, but apply to all the sites listed. If both are specified, all polarizabilities are printed, but this is the default.

WRITE FILE *filename* [LIMIT l_{\max}] $s_1 s_2 \dots s_n$

Write the polarizabilities α_{tu}^{ab} , for all a and b in the list specified, to the specified file in binary form. The structure of the file is as follows. Each polarizability matrix $\alpha^{s_i s_j}$ is preceded by a record of the form

$i j l_{ij} S_{ij}$

where $S_{ij} = (l_{ij} + 1)^2$ is the number of multipoles up to rank l_{ij} . This record is followed by S_{ij} records,

containing the first S_{ij} elements of each row of $\alpha^{S_{ij}}$ in turn.

The rank l_{ij} is the actual rank of the matrix $\alpha^{S_{ij}}$, unless LIMIT l_{\max} has been specified, in which case it is either l_{\max} or the actual rank, whichever is smaller.

Note that the initial record contains i and j , the ordinal number of the sites in the specified list — not the site number as assigned by Orient. The assigned site numbers may be different in other runs of the program, and this usage makes it possible to read the data back in without needing to know the site numbers.

10 Displaying information

SHOW *options*

Display information about the system. Any number of options may be specified on one line. The options are:

VARIABLES [REINPUT]

Print a list of the variables known to the program, and their current values.

CONSTANTS [REINPUT]

Print a list of constants. In both these cases, if REINPUT is specified, the names and values are printed in a form suitable for reinput to the program. However the DUMP and RESTORE commands are more suitable for that purpose.

SWITCHES

Print a list of switches with their current settings.

SYSTEM [INERTIA [MULTIPOLES]]

Calculate the system centre of mass, moments of inertia, and rotational constants. Also, if MULTIPOLES is specified, calculate the overall multipole moments of the whole system, relative to the centre of mass. The ‘system’ comprises all the molecules as defined, but not the lattice if there is one, and none of the periodic repetitions implied by NANOCRYSTAL or LAYER.

molecule key

ALL *key*

Show information about the specified molecule, or about all molecules in the system. The available keys are:

POSITION

Print the molecule’s position and orientation in global coordinates.

SITES

Print out the site positions in the global coordinate system.

GEOMETRY

Print the molecular geometries, i.e. site positions and orientations.

DATA

Print the molecular geometries, together with the site multipoles, in a form suitable for subsequent re-input. If any variables or constants were used in the original definitions it will be necessary to use SHOW VARIABLES or SHOW CONSTANTS to print their values.

WRITE FILE *filename*

Same as DATA, but write the information to the specified file.

MULTIPOLES

Print the total molecular multipole moments, referred to the molecular axes and origin.

11 Calculations of energy and related properties

ENERGY [DETAILS]

Evaluate the energy of the assembly of molecules. If the optional keyword DETAILS is present, full details of site positions and individual site-site interactions are printed. Note that calculations involving

a subset of the defined molecules can be performed by using SUPPRESS to mark the remaining molecules as inactive. They can be restored afterwards using the RESTORE command.

```
ENERGY TABLE [FORMAT fmt] PRINT [es] [er] [ind] [disp] [total]
  :
  items
  :
  :
END
```

Construct a table of energies evaluated at a set of specified configurations. The format of the tabulated energy values may be specified; the default is the standard format corresponding to the energy units currently in force. A list of components to be printed may be specified; the default is to print all of them and the total.

The items following the command may be

```
VARIABLES n1 n2 ...
v1 v2 ...
v'1 v'2 ...
:
:
END
```

The VARIABLES line specifies a list n_1, n_2 , etc. of names of one or more previously-declared variables. Each following line specifies values for this set of variables. Length variables are given in the current length units, angles in degrees. Any blank lines in the list of sets of values are echoed to the output. Lines of the form COMMENT "string" may also occur, in which case the string is printed. The set of data lines is terminated by END on a line by itself.

Other items that may occur are:

```
CURRENT
```

Show the variable values for the current configuration (must come first).

```
COMMENT "string"
```

Print the specified string.

The following command is provided for analysing many-body effects.

```
ANALYSE [n]
```

For any finite collection of molecules, the interaction energy can be expressed in the form

$$U = \sum_{i>j} U_{ij} + \sum_{i>j>k} U_{ijk} + \sum_{i>j>k>l} U_{ijkl} + \dots,$$

where U_{ij} is the interaction energy of molecules i and j alone in the absence of all the others, U_{ijk} is the correction to $U_{ij} + U_{ik} + U_{jk}$ required to get the energy of molecules i, j and k right in the absence of all the others, and so on. The ANALYSE command carries out interaction energy calculations for every subset of a system of up to 30 molecules, and analyses the results into pair interaction terms U_{ij} , three-body corrections U_{ijk} , etc., up to n -body terms. If n is omitted, the default is 6 or the total number of molecules present, whichever is smaller.

```
TRIPLE-DIPOLE C9 [m1 m2 m3]
```

Estimate the triple-dipole dispersion energy of the system. If a set of three molecules is listed, the energy is evaluated for that set only. Otherwise it is evaluated for every set of three molecules in the system, using the same C_9 triple-dipole dispersion coefficient in each case. It is assumed that the relevant interaction origin is the centre of mass, and no account is taken of any possible anisotropy in the C_9 coefficient.

```

SLICE
  [FILE file1]
  [GRID file2]
  name1 FROM min1 TO max1 STEP step1
  [ name2 FROM min2 TO max2 STEP step2]
END

```

The SLICE keyword allows the user to sample a potential energy surface along one or two coordinates. The coordinates correspond to named VARIABLES and are specified using the NAME keyword. Minimum and maximum values are specified with the FROM and TO keywords respectively. The step size along each coordinate is specified with the STEP keyword. The energies are written to standard output and to files specified by the FILE and GRID keywords. The output in *file1* is in the format:

$$\begin{array}{cc}
v_1[0] & E[0] \\
\vdots & \\
v_1[i] & E[i] \\
\vdots & \\
v_1[m] & E[m]
\end{array}$$

or

$$\begin{array}{ccc}
v_1[0] & v_2[0] & E[0,0] \\
\vdots & & \\
v_1[i] & v_2[j] & E[i,j] \\
\vdots & & \\
v_1[m] & v_2[n] & E[m,n]
\end{array}$$

for sampling one or two coordinates respectively. The output in *file2* is in the format:

$$\begin{array}{c}
E[0] \\
\vdots \\
E[m]
\end{array}$$

or

$$\begin{array}{cccc}
E[0,0] & \dots & E[0,j] & \dots & E[0,n] \\
\vdots & & & & \\
E[i,0] & \dots & E[i,j] & \dots & E[i,n] \\
\vdots & & & & \\
E[m,0] & \dots & E[m,j] & \dots & E[m,n]
\end{array}$$

for sampling one or two coordinates respectively. Here, the indices *i* and *j* span the values between the minimum and maximum quantities specified.

```

TABULATE [POTENTIAL] [FIELD] [GRADIENT] [E10] ... +++
  [COMPARE [UNIT unit-spec]]
  subcommands
  \vdots
END

```

Tabulate the electrostatic potential, and specified components of the electric field and the field gradient. If POTENTIAL is specified, the electrostatic potential is evaluated; V00 is a synonym. The components of the electrostatic field can be specified individually by the keywords E10, E11c or E11s (or Ez, Ex or Ey respectively); FIELD is equivalent to specifying all three components, in that order. Similarly individual components of the field gradient may be specified by E20 etc., or all five by the keyword GRADIENT. Note that the components of the electric field and field gradient tabulated by the program are minus the derivatives of the electrostatic potential, i.e., they are the electric field and field gradient as usually

defined. The spherical components of the field gradient are

$$\begin{aligned} E_{20} &= \frac{1}{2}(2E_{zz} - E_{xx} - E_{yy}), \\ E_{21c} &= \sqrt{3}E_{xz}, & E_{21s} &= \sqrt{3}E_{yz} \\ E_{22c} &= \frac{1}{2}\sqrt{3}(E_{xx} - E_{yy}), & E_{22s} &= \sqrt{3}E_{xy} \end{aligned}$$

The points at which these quantities are to be tabulated are specified by the subcommands following the TABULATE command itself:

RADII [*type radius type radius ...*] [ADD *d*] [SCALE *f*]

Specify van der Waals radii to be used subsequently to define a van der Waals surface. The radius for each site is initially the value specified in a previous TYPES command, if any. ADD adds the value *d* to all radii; SCALE multiplies all radii by *f*. The radii are restored to their original values at the end of the TABULATE command.

SURFACE [FOR] *molecule* [STEP *s*]

Generate points on the van der Waals surface of the specified molecule. The separation between points is approximately *s*. The points for each site are generated on the surface of a sphere with the radius currently in force for that site; this is repeated in turn for each site, and any points that fall within the sphere for another site are omitted.

LATTICE A *a_x a_y a_z* B *b_x b_y b_z* [C *c_x c_y c_z*] ORIGIN *o_x o_y o_z* +++
ALIMITS *i_a k_a* BLIMITS *i_b k_b* [CLIMITS *i_c k_c*]

Generate points in a lattice defined by lattice vectors *a*, *b* and *c*, with origin at the point specified. *i_a* and *k_a* specify the range of lattice points in the lattice *a* direction; similarly for *i_b* and *k_b* and for *i_c* and *k_c*. The definition of *c* and the CLIMITS may be omitted, in which case a single plane of points is generated. All of these quantities are defined in the global coordinate system.

x y z [V]

Specifies a single point at position (*x, y, z*) in global coordinates. *V*, if given, is the value of the electrostatic potential at the specified point; if COMPARE is specified, the program prints these values out together with its own values and the differences, so that other models may be compared with the distributed multipole model. COMPARE may not be specified with GRADIENT, SURFACE or LATTICE. The unit in which *V* is given is normally the energy unit set by default or by a UNITS command, but a different unit may be defined by specifying, on the TABULATE command line, UNIT *u* [*unit*]. *u* may be omitted, in which case it is assumed to be 1. The *unit* is one of HARTREE, kJ/mol, eV or cm⁻¹, or it may be omitted, in which case *u* must be present and the unit for *V* is *u* times the current energy unit.

12 Geometry optimization

```
OPTIMIZE (or MINIMIZE or SADDLE or NEWTON-RAPHSON)
:
options
:
END
```

Optimise the energy of the system of molecules, starting from the current configuration. The options can often be omitted, since there is a sensible default in all cases. The keyword OPTIMIZE may be replaced by MINIMIZE for an optimization with search type 0, *i.e.*, a minimization, by SADDLE for a saddle-point search, or by NEWTON-RAPHSON for a Newton-Raphson search.

When the geometry is optimized, the internal values for the geometrical parameters are updated. Molecules are treated as rigid during an optimization, so that no internal coordinates are optimized, but all intermolecular coordinates are normally optimized, whether they were declared initially as named variables, or specified as literal numerical values.

The optimization options may also be specified separately:

```
OPTIONS
  :
  options
  :
END
```

This is not necessary for a simple optimization, but it is needed for basin-hopping or for Monte Carlo or molecular dynamics simulations with quenching (see below).

Whether specified in an OPTIONS block or in the OPTIMIZE command, the options remain in force unless changed subsequently. This is an incompatible change from earlier versions of the program, where the default options were reset for each OPTIMIZE or MC or MD command. There is one exception to this: if the DUMP option is set and an optimization or simulation is carried out, it is unset after the dump so that the dump file is not accidentally over-written by later optimizations.

```
EVF
```

Use the eigenvector-following method for optimization. This is the default.

```
LMBFGS
```

Use the limited-memory BFGS (Broyden–Fletcher–Golfarb–Shanno) minimization algorithm. This cannot be used to find saddle points, but it should find minima more efficiently. It is also possible to carry out aminimizations with more flexible constraints than are possible with eigenvector-following.

```
RESET
```

Reset all options to the default.

```
SEARCH n
```

Carry out a search of type *n* for a stationary point. The search types are

- 0 (Default) Find a minimum by eigenvector following.
- 1 Carry out a Newton-Raphson search for a stationary point. This can be used to find stationary points of index higher than 1, provided that the starting-point is close enough.
- 2 Find a saddle-point of index 1 (1 imaginary frequency) by eigenvector following.
- 3 Find a minimum, using a pseudo-third-derivative correction.
- 4 Find a saddle-point of index 1, using a pseudo-third-derivative correction.

MINIMUM is a synonym for SEARCH 0, NEWTON-RAPHSON for SEARCH 1 and SADDLE for SEARCH 2.

```
ITERATIONS n
```

Allow up to *n* iterations of the optimization procedure. Default 100.

```
CONVERGENCE c
```

Set the convergence threshold for eigenvector-following. The optimization stops when the magnitude of the energy gradient is less than *c*. Default is 10^{-6} hartree/bohr. The value of *C* is taken to be given in terms of the current energy and length units.

```
MODE n
```

Start the search along the *n*th mode of the Hessian, counting from the one with the most negative eigenvalue as mode 1, and subsequently follow the mode that has the largest overlap with the direction of the previous step. MODE 0 follows the mode with the most negative eigenvalue at every step, and is the default.

```
STEP x
```

(Eigenvector-following only.) The maximum step-length is to be *x* units. *x* is treated as a distance in the current length units. Internally the units in which the steps are measured are bohr for distances and radians for angles. The default is 0.1 bohr.

```
SHOW [ENERGY] [VARIABLES] [GRADIENT] [HESSIAN]
```

Display information at each step of the optimization. Any or all of the options may be specified.

ENERGY: display the energy.

VARIABLES: display the values of the geometry variables.

GRADIENT: display the components of the gradient.

HESSIAN: display the components of the Hessian (second derivative matrix).

```
HESSIAN
```

(BFGS optimization only.) Evaluate the Hessian at the starting geometry.

```
FIX
  molecule [POSITION] [X] [Y] [Z] [ORIENTATION]
  :
END
```

Constrain the optimization by fixing the specified variables at their current values. It is possible to fix any one or more of the components *X*, *Y* and *Z* of the molecular (centre-of-mass) position; POSITION is equivalent to *X Y Z*. It is not possible to constrain individual orientational variables. If a molecule is specified without any coordinates, they are all constrained.

When using this option it is necessary to ensure that enough variables are constrained to define the geometry of the stationary point completely. That is, there should be no direction in the unconstrained space in which the system can move without changing the energy. For non-linear molecules, this can be achieved by constraining one molecule completely, but for linear molecules or atoms additional constraints would be needed.

For LMBFGS optimization, the DEFINE and POSITION options provide a more flexible way to define constraints.

```
DEFINE [NEW]
  name {[LENGTH] value | ROTATION BY angle ABOUT nx ny nz}
  :
END
```

This defines the names and initial values of the variables that define the optimization space. The *name* is an arbitrary string of characters (without any spaces), and the definition specifies either a length, with the initial value given, or a rotation. In the latter case the name specifies the three angle-axis variables defining the rotation. (*nx,ny,nz*) is a vector (not necessarily a unit vector) specifying the rotation axis, and *angle* is the rotation angle in degrees. If NEW is specified, any existing variable definitions are deleted.

```
POSITIONS
  molecule [AT] xspec yspec zspec ROTATED name
  :
END
```

The expressions defining the position of the specified molecule each comprise in general a constant plus a linear combination of the defined variables, written in the form

*[const {+|-}] name1 [*factor1] [{+|-} name2 [*factor2] ...*

Each name must be a previously defined variable. The operators “+”, “-” and “*” must be surrounded by spaces. For rotations, the specification comprises the name of a defined rotation.

```
INDUCTION firstind [maxind]
```

If the iterated induction energy is being calculated, the calculation is much more time-consuming. However, it may not be necessary to converge it completely at each step, and time may be saved by limiting the number of iterations that are performed. When the optimization is close to convergence, the geometry changes very little for several iterations, and the induction energy can reach convergence even though there may not be enough iterations for convergence in a single step. *firstind* is the maximum number of iterations to be permitted at the first step of the optimization. Default 40. *maxind* is the maximum number of iterations to be permitted in each of the remaining steps of the iteration. Default 40. If this option is used with a small value of *maxind* it is important to check that the induction energy has converged at the final geometry, for example by carrying out an ENERGY DETAILS calculation, which will converge the induction energy fully.

```
RESTORE file
```

Read the geometrical parameters from the specified file before starting the optimization or simulation. These values over-ride any values already given in the molecule declaration or in SET commands.

DUMP *file*

At the end of the optimization or simulation, write the geometrical parameters to the file specified, for subsequent re-input. The parameters comprise the position of each site, followed by the orientation matrix for any site that is defined to have a different orientation from its parent. As noted above, this option is unset after the dump has been carried out, to avoid over-writing the dumpfile inadvertently in a later stage of the calculation.

```
PLOT [[NEW] FILE file] [EVERY n | LAST] [TITLE title] +++  
      [LATTICE a1 a2 b1 b2] [LAYER a1 a2 b1 b2]
```

Write the geometry to the specified file in xyz format. If NEW is specified, a new file is opened with the name given, overwriting any existing file; otherwise the output is appended to the specified file. If no file is specified, and a plot file is already open, the output is appended to that. If 'EVERY *n*' is specified, and xyz format is used, a frame of plot data is produced at every *n*th iteration. 'LAST' calls for a plot of the final geometry only. The default is 'EVERY 1'. The title, if given, is used as a heading in the xmol 'animate' window, together with the iteration number and the energy. Remember to enclose the title in quotes if it contains spaces. The LATTICE and LAYER options specify a range of unit cells for plotting in the crystal and adsorbed layer respectively. They are irrelevant if there is no lattice or periodic layer, but their arguments are read anyway and could affect subsequent plots. The default in each case is to plot a single unit cell, i.e. for example LATTICE 0 0 0 0.

REALIGN *molecule*

If REALIGN is specified, the system is moved as a whole at the end of the optimization so that the named molecule is in its original position and orientation. This option is now redundant, since the same effect can be achieved using the FIX command.

```
FREQUENCIES [FILE file] [ANIMATIONS [FRAMES nframes]] [INTENSITIES] [VECTORS]
```

Print the intermolecular vibration frequencies (in cm^{-1}) at the end of the optimization. Note that these will include five or six frequencies close to zero, corresponding to the overall vibrations and rotations. If a file is specified, the frequencies are written to that file also.

If the ANIMATIONS keyword is specified, normal mode animations are written in xyz format to the files *file_moden.xyz* (or *moden.xyz* if no filename has been given). A separate file is written for each mode, including the translations and rotations. Each file contains *nframes* frames; the default value of *nframes* is 36.

If the INTENSITIES keyword is specified, the infrared intensities of the normal modes are calculated and written to the file *file_intens.dat* (or *intens.dat* if no filename has been given).

If the VECTORS keyword is specified, the Cartesian coordinates and normal mode displacement vectors are written in xyz format to the file *file_vec.xyz* (or *vec.xyz* if no filename is given).

THERMOFUNCS *options*

Calculate the change in the thermodynamical functions of optimized configurations, or quenched configurations in a simulation, compared to the separated constituents. The polyatomic ideal gas approximation (McQuarrie, 1976) is used. All the translational and rotational contributions, and the intermolecular vibrational contributions, are taken into account, but not the changes in intramolecular vibrations and electronic contributions. In a simulation, the calculation is done after every quench. The options are

FILE *file*

The file in which the thermodynamical functions are stored (default *thermofuncs.dat*). The columns are

D_e	kJ mol^{-1}	dissociation energy from the bottom of the well
D_0	kJ mol^{-1}	ZPE-corrected dissociation energy
T	K	Temperature
ΔS	$\text{J mol}^{-1} \text{K}^{-1}$	Entropy change
$T\Delta S$	J mol^{-1}	Temperature \times Entropy change
ΔU	kJ mol^{-1}	Thermodynamical Energy change
ΔH	kJ mol^{-1}	Enthalpy change ($H = U + RT$)
ΔA	kJ mol^{-1}	Helmholtz Free Energy change ($A = U - TS$)
ΔG	kJ mol^{-1}	Gibbs Free Energy change ($G = H - TS$)

PRESSURE *press*

The external pressure which is used in the ideal gas approximation. The unit is Pascal and the default is 101,325 (= 1atm).

TEMPMIN *tmin*

The minimum temperature for which the thermodynamical functions are estimated. If no temperature grid is defined, i.e. *tgrid*= 1, the calculation is done at this temperature only. The temperature is measured in Kelvin and the default is 298.0.

TEMPMAX *tmax*

The maximum temperature of the calculations. If a grid is defined the changes in the functions are estimated between *tmin* and *tmax*. The unit is Kelvin and the default is 298.0.

TEMPGRID *tgrid*

The number of temperature points from *tmin* to *tmax* inclusive. This parameter provides for calculating the thermodynamical functions over a given temperature interval. The default is 1 (no temperature grid).

13 Monte Carlo Simulations in ORIENT

The energies calculated by the ORIENT program can be used directly for Monte Carlo simulations of the canonical ensemble.

A simulation may start with an adjustment period to determine the sampling parameters automatically. Position and orientation variables (Evans, 1977; Evans and Murad, 1977; Allen and Tildesley, 1987) can be treated independently. To perform ergodic sampling accurately the *Jump Walking* approach is available (Frantz *et al.*, 1990; Tsai and Jordan, 1993). Every run can use configurations from a separate higher temperature run with the appropriate stochastic transition matrix (*jumping*). Similarly the results of a lower temperature run can be used in the same way (*diving*). The independent simulations are run at the same time and communicate via buffer files which are filled at random to prevent correlation effects.

The resulting density of configurational energies may be stored in a histogram and fitted by a Gaussian expansion. Thermodynamical observables are estimated with Ferrenberg and Swendsen's histogram method (Ferrenberg and Swendsen, 1988, 1989; Tsai and Jordan, 1993). All calculations can be run in a loop over a range of temperatures. An optional spherical container prevents evaporation.

The Monte Carlo routine in ORIENT is called by the MC environment

```
MC
:
subcommands
:
END
```

An example of a Monte Carlo data section is shown in Fig. 9. The following commands control the calculation. Many of them have a number of options, which are given one to a line, terminated by a line containing just "end". Energy values are always measured in the current energy unit. The temperature unit is Kelvin.

```

MC
  minimum
  induction 1
  maxstep 100
  niter 500
  dump md01.dump
  restart minimum.dump

  temp initial 35.0
  steps
    number 1000000
    maxdisp 0.3 openend
  end

  container radius 15
  adjust
    steps 100000
    frequency 10000
    rateconverge 4
  end
  jumpwalk
    jumper dtemp01.com
    jumpertemp 0.0002
    diver dtemp02.com
    diveevery 10
  end

  histogram
    file mc01.hist
    size 2000
  end
  density
    caloriccurve mc01.cal
    fitfile mc01.fit
    fitconverge 10
  end
  statistics
    all
    every 5000.0
    file mc01.stat
  end
  quenching
    every 10000.0
    file mc01.quench
    geometry mc01.xyz
    title Ar13-Quenching
    globalmin mc01.gmin
  end
end

```

Figure 9: Example of a Monte Carlo data section.

13.1 Adjustment period

```

ADJUST
  :
  options
  :
END

```

A simulation may start with an adjustment period for determining the maximum displacements in the configuration space. These limits are responsible for the acceptance rate of the sampled geometries and with it for the efficiency of the simulation (Metropolis *et al.*, 1953). Afterwards the initial configuration will be reset. The adjustment procedure is independent of the *Jump Walking* technique. The options are:

STEPS *number*

Maximum number of steps in the adjustment period. The default is 100,000.

FREQUENCY *freq*

The adjustment procedure measures the acceptance rate in partial simulations of *freq* steps and changes the displacements if necessary afterwards. The default is 10,000.

ACCEPTRATE *rate*

The given target acceptance rate which should be achieved with the converged maximum displacements. The default is 0.5.

TOLERANCE *tol*

The tolerance limit for the acceptance rate. Within this interval around the target value *rate* the displacements will not be changed and the adjustment is considered to be converged. The default is 0.05.

RATECONVERGE *conv*

After *conv* partial simulations are converged to the specified acceptance rate the adjustment procedure ends and the main simulation starts. If *conv* is set to zero (default) the adjustment period always continues for the specified number of steps.

DELTADISP *ddisp*

Specifies the fraction *ddisp* by which the maximal displacement in the position variable is increased or decreased if the acceptance rate is too high or low after a partial simulation. The default is 0.3 (i.e. $\pm 30\%$).

DELTAROT *drot*

Specifies the fraction by which the maximal displacement in the rotation variable is increased or decreased if the acceptance rate is too high or low after a partial simulation. The default is 0.3 (i.e. $\pm 30\%$).

13.2 Container

CONTAINER [RADIUS *radius*]

The command defines a spherical container around the system to prevent evaporation. This is appropriate in most Monte Carlo simulations. *radius* is the radius of a spherical container around the centre of mass of the system. All particle positions are restricted to this sphere. The default is 0.0 (no container).

13.3 Thermodynamical properties

The sampled probability distribution of the potential energy can be used to estimate the thermodynamical properties of the system like the caloric curve. This calculations are done with Ferrenberg and Swendsen's histogram method (Ferrenberg and Swendsen, 1988, 1989; Tsai and Jordan, 1993). The command also provides smoothing and fitting to an analytical expression of the density distribution for further calculations. The histogram is fitted to a sum of displaced Gaussians with the Levenberg-Marquardt method (Press *et al.*, 1992). In addition an old density distribution can be loaded from a file.

The subcommand is

```
DENSITY
:
options
:
END
```

and the options are:

CALORICCURVE *ccfile*

Specifies the name of a file to which the caloric curve data are written. Default `mcrun.ca1`. The expectation values for the entire system of the potential energy $\langle E_{\text{pot}} \rangle$ and the heat capacity $\langle C_V \rangle$ are calculated from the sampled density distribution of potential energy as a function of temperature. The histogram method of Ferrenberg and Swendsen is used to get the approximation for the range between *ccmin* and *ccmax* around the simulation temperature. The data columns in the file are

$$T/\text{K} \quad \langle E_{\text{pot}} \rangle \quad \langle E_{\text{pot}}^2 \rangle \quad \langle C_V / \text{J K}^{-1} \text{ mol}^{-1} \rangle$$

CALORICMIN *ccmin*

The start temperature in Kelvin for the calculation of the caloric curve. The default is set 5% below the minimum temperature of all the loops.

CALORICMAX *ccmax*

The end temperature in Kelvin for the calculation of the caloric curve. The default is set 5% above the maximum temperature of all the loops.

CALORICGRID *ccgrid*

The number of temperature points between *ccmin* and *ccmax* for which the caloric curve is calculated. The default is 100.

FITFILE *fitfile*

If this filename is set a histogram due to the analytical fit is stored in *fitfile*. The histogram size corresponds to the original one. In addition the analytical expression of the fit is written to *fitfile.an1*. If the simulation consists of more than one temperature loop the respective loop number is added to every filename automatically.

FITORDER *order*

The histogram is fitted to a sum of Gaussians,

$$\Omega_{\text{pot}}^{\text{fit}}(E) = \sum_{k=1}^{\text{fitorder}} a_k \exp \left[- \left(\frac{E - b_k}{c_k} \right)^2 \right].$$

So *fitorder* sets the order of the expansion in the analytical model. The default is 3, the maximum is 16.

FITQUALITY *quality*

The Levenberg-Marquardt method defines a *merit function* to measure the agreement between the fit and

the histogram. Every fitting step should improve the analytical model and changes the *merit function*. The quantity *quality* defines a relative alteration limit in which the *merit function* and with it the fit is considered to be converged. It controls the quality of the analytical model. The default is 10^{-5} .

A check of the fit by the user is always recommended because automatic fitting can converge to an absurd result. A change in *fitorder* or *quality* usually helps in such a case.

FITCONVERGE *converge*

The fitting procedure stops if the *merit function* is *converge* times in a row converged, *i.e.*, the relative fluctuation is below the limit *quality*. The default is 20.

FITMAXITER *maxiter*

The maximum number of steps in the fitting procedure. If the analytical model has not achieved the required quality after *maxiter* steps, the fitting stops. The default is 1000.

FITREPORT

If this option is set the program prints out information about the convergence during the smoothing iterations.

TAKEOLD *oldfile*

If *oldfile* is set the program loads an old histogram from the file *oldfile* and uses it in the calculations. The run stops after the fitting, the estimation of the thermodynamical properties and/or the file storage, respectively. The FITREPORT option is set automatically.

13.4 Histogram

The density of potential energies is stored in a histogram at each temperature. Details are controlled by the HISTOGRAM command. See also the DENSITY command (previous section). The options are:

SIZE *size*

The number of finite intervals (bins) in the histogram to which the different potential energies are classified. The maximum is 10,000 and the default is 2,500.

FILE *file*

If *file* is set the histogram is saved in the specified file after each temperature. If there is more than one temperature loop in a simulation the number of the respective loop is appended to the filename automatically, so the histograms are stored separately. The default is `mcrun.hist`.

MINIMUM *minimum*

MAXIMUM *maximum*

SAFETYFACTOR *safefac*

These options are used to specify the range of potential energy values which are taken into account in the histogram. If *minimum* is zero, or the MINIMUM option is omitted, it is set to

$$minimum = E_{\min}^{\text{meas}} - safefac \times (E_{\max}^{\text{meas}} - E_{\min}^{\text{meas}}),$$

where E_{\max}^{meas} and E_{\min}^{meas} are the highest and lowest measured energies found during the adjustment period. Similarly, if *maximum* is zero, or the MAXIMUM option is omitted, it is set to

$$maximum = E_{\max}^{\text{meas}} + safefac \times (E_{\max}^{\text{meas}} - E_{\min}^{\text{meas}}).$$

The SAFETYFACTOR option is used to set the multiplication factor which is used if the histogram boundaries are adjusted automatically. The default is 0.5.

13.5 Jump Walking

The *Jump Walking* approach (Frantz *et al.*, 1990; Tsai and Jordan, 1993) makes the Monte Carlo simulation more ergodic. The basic idea is to jump from time to time to an uncorrelated new point in the configuration space during the sampling. This should prevent the run from being trapped in a small area of this space. The new configuration is taken from a higher temperature simulation which should be

more ergodic. In addition it increases the efficiency of the sampling because the important configurations between the energy barriers are more likely than in a uniform distribution. This causes a new statistical transition matrix for the sampling which is taken into account by the program.

If a Monte Carlo run is jumping to a higher temperature distribution from time to time it is called a *Jumper*. If a run at higher temperature is supposed to deliver a *Jumper* at lower temperature it is called a *Diver*. Also every run could be both a *Jumper* and a *Diver*. Hence a simulation which includes *Jump Walking* consists of two or more runs simultaneously. If a run is a *Jumper* it communicates with the next higher temperature run (its *Diver* via a buffer file. Entries are continuously chosen at random and changed by the *Diver*. The access from the *Jumper* is also random. Because of this a reasonably big buffer prevents correlation effects between the two runs. The right choice of the temperature differences and the highest temperature is decisive for the quality and efficiency of the simulation compared to independent runs at the same temperatures.

The command is

```
JUMPWALK
  :
  options
  :
  END
```

and the options are:

JUMPER *jumpcom*

This command characterises a *Jumper*. *jumpcom* is the name of the buffer file for the *Jumper/Diver* communication. An independent *Diver* is filling this buffer constantly with configurations from its higher temperature distribution. If the *Jumper* finds this buffer it starts a simulation and jumps with a preset probability from time to time to a configuration chosen randomly from the buffer. Obviously the *Jumper* and the corresponding *Diver* at higher temperature must use the same communication file. The default is `mcrun.jump`.

JUMPERTEMP *jtemp*

The temperature of the independent *Diver* in Kelvin. *jtemp* must be known to construct the correct transition matrix in the sampling. The temperature difference between the *Diver* and the *Jumper* determines the acceptance rate for a new configuration from the higher temperature run. This rate is reported in the output of the statistics. A recommended rate is 0.5.

JUMPCHANCE *jchance*

The probability for trying a configuration from the higher temperature distribution, *i.e.*, from the buffer file, in a Monte Carlo step. The default is 0.1.

DIVER *divecom*

This command characterises a *Diver*. Like *jumpcom* it defines the name of the buffer file for the *Jumper/Diver* communication. The *Diver* fills the buffer with configurations constantly during the run. Again one must ensure that the corresponding *Jumper* at lower temperature is using the same buffer file. The default is `mcrun.dive`.

DIVEEVERY *devery*

The frequency of saving the configurations (*diving*) by the *Diver*. Every *devery* Monte Carlo steps the current coordinates are stored in *divecom* sequentially. The default is 10.

BUFFER *size*

The number of configurations which are stored in the communication buffer file. The *Jumper* and the *Diver* must use the same buffer size. *size* can be used to optimise the computational costs of the simulation. If the *Diver* is quenching the geometry from time to time *size* should not be too small, otherwise some configurations could be overemphasised in the *Jumper*. The default is 500.

DELAYFACTOR *delay*

After the adjustment procedure a *Jumper* sleeps for a while to wait for an equilibrated filling of the buffer file. This period depends on the buffer size *size* and the diving frequency *devery* of the *Diver*. The factor *delay* offers a chance to manipulate this time delay if the buffer takes longer to equilibrate. The default is 1.0.

13.6 Lennard-Jones systems

```
LJATOMS [EPSILON  $\epsilon$ ] [SIGMA  $\sigma$ ]
```

A simple Lennard-Jones pair interaction can be used for a system of atoms, and makes the simulation much faster. The Lennard-Jones potential is used only for the simulation; if the system is quenched the more complex interaction defined in the ordinary way is used. The options specify the well-depth ϵ and the diameter σ in the Lennard-Jones potential

$$V_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right].$$

13.7 Quenching

A potential energy surface can be explored by performing a high-temperature Monte Carlo simulation, and quenching the system from time to time — that is, by optimizing the geometry, in effect reducing the temperature to zero so that the system falls into the nearest minimum. Afterwards the simulation continues from the configuration from which it was quenched, i.e., as if the quench had not happened.

```
QUENCHING
:
options
:
END
```

This command specifies the number of steps after which the configuration of the simulation is quenched to a stationary point. The optimization procedure (Wales *et al.*, 1995) is controlled by a preceding OPTIONS section, as for the OPTIMIZE environment. The quenched energies and optionally the geometries are written to specified files. If the THERMOFUNCS options has been set in a previous OPTIONS block or optimization, the program calculates the change in the thermodynamical functions of each quenched configuration compared to the separated constituents. See above for details.

The quenching options are:

```
EVERY n
```

After every n Monte Carlo steps the current configuration is quenched. The default of n is 0 (no quenching).

```
FILE file
```

The file where the optimised energies are stored. The output is in columns in the order *time-variable* *total energy*

The default file is `mcrun.quench`.

```
GEOMETRY geofile
```

If this option is set the quenched geometries are stored in *geofile.xyz*. The output is in *xyz*-format, with one frame for each quenched geometry, and includes energy information in the frame title.

```
TITLE title
```

The title of the geometry data. If *title* is set the optimised geometries are stored in *geofile*. The corresponding energy is always appended to the title. The default title is `MC-Quenching`.

```
GLOBALMIN gmfile
```

If this file is set the lowest lying configuration in energy of all quenched geometries is saved separately. This can be used to look for the global minimum during a simulation. The configuration is saved in *xyz*-format (*gmfile*) and in a dump-file (*gmfile.dump*) which is suitable for the RESTART command.

```
TAKEQUENCHED
```

Usually the quenching does not affect the sampling loop. However, if this option is set the Monte Carlo simulation continues with the optimised geometry. This is sometimes useful for searching the global minimum.

13.8 Statistics

```

STATISTICS
:
options
:
END

```

This command controls the calculation of the statistical observables in the Monte Carlo simulation, and causes them to be saved to file at intervals. All canonical observables should converge during the integration.

EVERY *n*

The statistics are updated at every Monte Carlo step, but saved to file only after every *n* steps. The default for *n* is 1,000.

FILE *file*

The name of the file in which the statistical observables are stored. The output is in columns and depends upon whether the ALL option is set or not. If the simulation consists of more than one temperature loop the respective loop number is added to every file automatically. The default file is `mcrun.stat`. The order of the columns in the statistics file is

```

step Paccept Pbump ⟨Epot⟩ σ(⟨Epot⟩) σrel(⟨Epot⟩) σ(Epot) σrel(Epot) δLind σ(δLind) σrel(δLind)

```

The file is an ordinary text file with columns separated by spaces. Notes:

- P_{accept} is the acceptance rate and P_{bump} the container bump rate.
- $\langle x \rangle$ is the average of x , $\sigma(x)$ the standard deviation and $\sigma_{\text{rel}}(x)$ the relative standard deviation compared to the average of the averages.
- E_{pot} is the potential energy. The different deviations correspond to the spread of the respective mean values, $\sigma(\langle E_{\text{pot}} \rangle)$, and to the deviation of all potential energies of the sampled configurations (width of the density profile).
- The Lindemann criterion ([Rose and Berry, 1992](#)),

$$\delta_{\text{Lind}} = \frac{2}{N(N-1)} \sum_{i < j}^N \frac{\left(\langle r_{ij}^2 \rangle - \langle r_{ij} \rangle^2 \right)^{\frac{1}{2}}}{\langle r_{ij} \rangle},$$

is calculated from the fluctuations of the site centre of mass distances r_{ij} .

13.9 Step control

The subcommand

```

STEPS
:
options
:
END

```

controls the number of Monte Carlo steps and the maximum displacement at each step. In *Jump Walking* simulations the actual number of steps might be higher.

NUMBER *number*

The total number of steps in the Monte Carlo simulation for each temperature loop. In *Jump Walking* runs this is a minimum step number for a *Diver*. The default is 1,000,000.

OPENEND

This option is only used in a *Diver* of a *Jump Walking* sampling. If OPENEND is set the program delivers a separate *Jumper* until the *Jumper* sends a stop-signal. But on the other hand the simulation takes at least *number* steps. See also the JUMPWALK command.

MAXDISP *ddisp*

The initial maximum displacement in the position variable for the adjustment period or the main simulation, respectively. The default is 0.1 bohr.

MAXROT *drot*

A new orientation is generated by Euler-type rotation around randomly chosen angles. *drot* sets the relative range of the rotation angles. If it equals 1.0 the rotations are sampled from the entire range of Euler angles, so that the new orientation is completely uncorrelated to the previous one. If *drot* is 0.0 the rotational degrees remain unchanged. In most cases the acceptance rate is very sensitive to the variation of the orientation and *drot* should be set with care. An estimate can be obtained by fixing the positions (i.e. *ddisp* equals 0.0) for various *drot*. Sometimes it is not necessary to change the orientations at all, e.g. if there is only one non-atomic molecule. Sampling without rotations can be more efficient in this case. The default for *drot* is 0.0.

COMMANDFILE *file*

The name of the *Command File*. The *Command File* provides control over a running simulation. It is a simple text file which contains keywords. After the keywords have been processed the *Command File* is deleted automatically, unless the KEEP keyword is present. The default file is `mcommand.txt`.

Keywords for the *Command File* are

STOPLOOP

The simulation will end as soon as the current temperature loop is finished.

SKILOOP

As soon as the current temperature loop is finished the simulation will leave out the next temperature and jump to the next higher value directly. This keyword could be used more than once to jump over several temperatures.

KEEP

The *Command File* will not be deleted after it has been read, so all commands will be taken into account after every temperature loop.

13.10 Temperatures

The command

```
TEMPERATURE
:
options
:
END
```

sets the temperature variable which is used in the importance sampling.

INITIAL *initial*

The absolute initial temperature of the system in Kelvin. It is used in the importance sampling following the canonical simulation scheme of Metropolis et al. ([Metropolis et al., 1953](#)). The default is 0.0.

LOOPS *loops*

Number of simulation loops with stepwise increased temperature. Every loop is always sampled for the number of steps specified in the STEPS command. After finishing a loop the temperature is increased by *delta* and the configuration remains unchanged. *loops* must be 1 in a *Jump Walking* run. The default number of loops is 1.

DELTA *delta*

The temperature value in Kelvin which is added after every loop. The default is 0.0.

SAMESTART

Usually every new temperature loop starts from the last configuration of the preceding one. If SAMESTART is set every loop starts from the same initial configuration.

14 Basin-hopping in ORIENT

Many of the facilities in the Monte Carlo section were provided to assist with finding minima and mapping out the transition paths on a potential energy surface. The recommended procedure for this is now basin-hopping (Wales, 2003), and a basin-hopping routine is now available in ORIENT.

Basin-hopping is a variant of Monte-Carlo in which the energy associated with a point in configuration space is not the true energy but the energy of the minimum of the basin in which the point lies. At each step, therefore, the energy is minimized. If the resulting configuration is a known minimum, a new step is taken from the current point (not from the minimum). If the configuration is a new minimum, the energy is compared with the energy associated with the previous point and the step is accepted or rejected using the Metropolis criterion. If the step is rejected, a new step is attempted from the previous point, and if it is accepted, the next step is taken from the current point. Sometimes the minimization fails, usually because the new point is in a spurious well where undamped attractive forces overwhelm the repulsion, or sometimes because the minimization fails to converge. In this case, a new step is taken from the previous point; see the REJECT option below.

Options for the energy minimization should be set using the OPTIONS command (p. 28). The LMBFGS optimization procedure is recommended, but eigenvector following can be used. If the minimizations often fail to converge, the maximum number of steps for the optimization should be increased.

14.1 Syntax

The syntax for basin-hopping is as follows:

```
BASIN-HOPPING
:
subcommands
:
END
```

The available subcommands are:

```
VERBOSE
```

Print more information.

```
QUIET
```

Print less information (default).

```
ALLOCATE
  MINIMA n
END
```

Reserve space for details of up to n minima. Default 100.

```
TEMPERATURE T
```

Use temperature T kelvin in the Metropolis criterion for accepting or rejecting a possible step. This is not really a physical temperature – just a parameter for the procedure. For a cluster, it should correspond roughly to the energy likely to be needed to dissociate a molecule from the rest of the cluster, relative to the global minimum.

```
kT energy
```

Alternative specification of the temperature by giving kT in current energy units.

```
STEPS
  NUMBER n
  MAXDISP  $\Delta x$ 
  MAXROT  $\Delta\psi$ 
  BLOCKS block-list
END
```

Here n is the maximum number of successful hops, Δx is the maximum displacement in any coordinate,

in current length units, and $\Delta\psi$ the maximum rotation of any molecule, in degrees.

The BLOCKS entry allows for a series of steps to be carried out in which only the positions, or only the orientations, are changed. The *block-list* is a list of items, each of the form POSITIONS *n* or ROTATIONS *n* or ALL *n*. Any number of such items may occur, in any order. For example,
ALL 100 POSITIONS 100 ALL 100 ROTATIONS 100
specifies that both positions and orientations may be changed for 100 steps, then positions only for 100 steps, then both for 100 steps, then rotations only for 100 steps. Then the same cycle starts again. Note that even if only positions or only rotations are changed when a step is taken, both positions and rotations may change during the optimization.

CONTAINER RADIUS *R*

It is usually advisable to specify a container to prevent molecules evaporating from the cluster. The container radius is *R*, in the current length units.

REJECT E_{\min}

Monte-Carlo and related procedures commonly run into trouble when the potential energy surface has spurious negative singularities, usually because the repulsion is bounded but there are undamped attractive terms that may become infinite. This subcommand specifies that if the energy falls below E_{\min} , the system is assumed to be in such a non-physical region. The current step is abandoned, and a new random step is taken from the last acceptable configuration.

DIFFERENCE [ENERGY *etest*] [MOMENTS | MI | M.I.] *mitest*

When a minimum is reached, it is compared with previous minima to determine whether it has been seen before. The energy is compared with the energy of each previous minimum, and if it is the same to within *etest* the principal moments of inertia are calculated and compared. If they are all the same to within *mitest* the minimum is assumed to be the same. The default *etest* is 10^{-6} hartree, and the default *mitest* is 10^{-6} . Energy values specified in the command are assumed to be in current energy units. The M.I. test is based on relative values: if the ratio of corresponding moments of inertia differs from 1 by less than *mitest* they are taken to be the same. If *mitest* is set to zero the moments of inertia are not tested.

FILES

file specifications

END

Possible items here are

PLOT *xyzfile*

File for xyz plot files of each minimum. Default *minima.xyz*.

MINIMA *minima_file*

File for the geometry of each minimum, in a form suitable for re-entry into ORIENT.

DUMP-PREFIX *prefix*

Prefix for dumpfiles. If this is non-null, each minimum geometry is dumped to disc in a file *prefixn.dump* where *n* is the number of the minimum. Default is null. The dumpfile gives a detailed specification of the geometry, including the position and orientation of every site. The dumped geometry can be imported into another Orient job by means of the RESTORE GEOMETRY command (p. 7).

SEED *s*

Specify the seed *s*, an integer between 1 and 9999, for the random number generator.

ADJUST

options

END

The program can be set to adjust the maximum translation or rotation displacement occasionally in order to achieve a satisfactory acceptance ratio in the Metropolis procedure. Options are

TARGET *t*

This is the target acceptance ratio. Default 0.3.

FRACTION *f*

The fraction by which the maximum translation and rotation displacements are to be increased or de-

creased. Default 0.1.

EVERY n

Adjustments are to be applied every n steps. Default is 0, meaning no adjustments.

SWITCH n_{sw}

After every n_{sw} steps, switch between adjusting the maximum translation displacement to the maximum rotation displacement, or *vice versa*.

OFF

Don't adjust the maximum displacements. The default is to make no adjustments.

The procedure for adjusting the maximum displacements is as follows. After n steps the maximum displacement is changed by a random fraction between $-f$ and $+f$. If after another n steps the acceptance ratio has improved (got closer to the target) another random change is made. If it has got worse, the previous step size is restored.

RESTART *file*

Restart the basin-hopping procedure from the last minimum found. At the end of a basin-hopping run, or if the space for new minima is exhausted, the current configuration is saved in a file called `save.cfg`. The procedure can be restarted by using the same data file, with the subcommand `RESTART save.cfg` added to the basin-hopping section. The restart procedure also needs the minima-file, for details of the minima already found. The allocation of space for recording the minima needs to allow for these old minima as well as any new ones that may be found.

14.2 Interrupting a basin-hopping job

If a file called 'stop' is created in the directory where the job is running, the procedure will stop cleanly after the current step. It may then be restarted, with changes to the parameters in the data file if required. The stop file needs to be deleted before the job is restarted. The step count starts from the last step taken, so the maximum number of steps may need to be increased.

15 Molecular Dynamics in ORIENT

The forces and energy calculated by the ORIENT program can be used automatically for Molecular Dynamics simulations of the microcanonical ensemble. Time evolution and time averages of various observables may be calculated to estimate, *e.g.*, the kinetic temperature or the Lindemann δ (Rose and Berry, 1992). The optional stepwise change in the kinetic energy for multiple runs allows the caloric curve of the system to be examined. Furthermore the optimisation routine of ORIENT is still available to quench the system at arbitrary phase points, so that the potential energy surface can be examined. Alternative integration methods are available including an efficient and stable symplectic integration scheme (Dullweber *et al.*, 1997). All statistics, energies, quench results and geometries of the trajectory can be stored in files to give full information about the dynamics.

The Molecular Dynamics routine in ORIENT is called by the MD command

```
MD
:
subcommands
:
END
```

An example of a molecular dynamics data section is shown in Fig. 10. The following subcommands control all important parameters and the output of the integration. Many of them have a number of options, which are given one to a line, terminated by a line containing just "end". Energy values are always measured in the current energy unit (See the UNITS command, p. 5). The temperature unit is

```

MD
  minimum
  induction 1
  maxstep 100
  niter 500
  dump md01.dump
  restart minimum.dump

  time
    start 0.0
    end 1000000.0
    timestep 1.0
  end
  energy
    initial 0.0002533
    loops 5
    delta 0.000015
    every 500.0
    file md01.eng
  end
  quenching
    every 10000.0
    file md01.quench
    geometry md01.xyz
    title BzAr5-Quenching
    globalmin md01.gmin
  end
  statistics
    all
    every 5000.0
    file md01.stat
  end
end

```

Figure 10: Example data section for an ORIENT molecular dynamics simulation

Kelvin, and times are in femtoseconds (fs).

15.1 Integration accuracy

```

ACCURACY
:
options
:
END

```

The command determines the relative integration accuracy of the Molecular Dynamics integration corresponding to the total energy of the system and the internal coordinate vector respectively (Evans, 1977; Evans and Murad, 1977; Allen and Tildesley, 1987). The error estimator of the adaptive step size control takes both error limits into account independently. The options are:

ENERGY *energy*

Relative integration accuracy referring to the initial total energy of the system. The step size control tries to keep the relative standard deviation below the value of *energy*. This should be the main control variable of the integration precision. If the energy accuracy is too low, *i.e.* *energy* is too big, the integration might diverge. The default is 10^{-5} Hartree.

COORDINATE *coord*

Relative integration accuracy referring to the internal coordinate representation (Evans, 1977; Evans and Murad, 1977; Allen and Tildesley, 1987). If the coordinate accuracy is too low, *i.e.* *coord* is too big, the integration might diverge but on the other hand excessive precision reduces the integration speed significantly without calculating a better microcanonical trajectory. The default is 10^{-6} bohr.

15.2 Energy control

```
ENERGY
  ⋮
options
  ⋮
END
```

This command sets the initial total energy, the energy calculation interval and with it the updating and the file storage of the corresponding statistics. Furthermore it allows a stepwise increase of the kinetic energy during one integration for successive trajectories.

INITIAL *initial*

The absolute initial total energy of the system. This value is measured in the current energy units. The kinetic part is equally distributed to every vibrational degree of freedom. The initial overall linear and angular momentum always vanish. When restarting an old trajectory this allows the old configuration to be used with a new total energy. If no initial total energy value is given all velocities of the system are unchanged.

LOOPS *loops*

Number of integration loops with stepwise increased kinetic energy. Every loop is always integrated over the entire time range. After finishing a loop the kinetic energy is increased by *delta* and distributed to every vibrational degree of freedom. The overall linear and angular momentum are forced to zero afterwards. The configuration remains unchanged. The default number of loops is 1.

DELTA *delta*

The kinetic energy value which is added to the system after every loop. The default is 0.0. The change in temperature can be estimated by dividing by $\frac{1}{2}Nk_B$, where N is the number of degrees of freedom and k_B is the Boltzmann constant.

EVERY *t*

The minimal time-interval at which the different energy contributions are estimated and the statistical variables are updated. Because of the adaptive step size in the time-variable the calculations are not exactly equidistant in time. The use of this option does not automatically cause storage of the energies in a file. The default is 1.0 fs.

FILE *file*

The name of the file in which the energies are stored after every energy calculation. The output is in columns and depends upon whether the ALL option is set or not. The default file is `mddrun.eng`.

ALL

This option produces more detailed output. Usually the order of the columns in the energy file is *time-variable kinetic energy potential energy total energy*

With the ALL option set the additional columns are the energy contributions

... *electrostatic repulsion induction dispersion*

The file is an ordinary text file with columns separated by spaces.

The energy of the system can also be set using the command

TEMPERATURE *initial*

This command specifies the initial kinetic temperature of the system in Kelvin. The corresponding kinetic energy is equally distributed to every vibrational degree of freedom. The initial overall linear and angular momentum always vanish. When restarting an old trajectory this allows the old configuration to be used with a new kinetic energy.

15.3 Lennard-Jones systems

```
LJATOMS [EPSILON  $\epsilon$ ] [SIGMA  $\sigma$ ]
```

A simple Lennard-Jones pair interaction can be used for a system of atoms, and makes the simulation much faster. The Lennard-Jones potential is used only for the simulation; if the system is quenched the more complex interaction defined in the ordinary way is used. The options specify the well-depth ϵ and

the diameter σ in the Lennard-Jones potential

$$V_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right].$$

15.4 Integration method

The command

`METHOD method`

selects the numerical integration method which is used in the simulation.

`SYMPLECTIC`

An explicit symplectic splitting method (Dullweber *et al.*, 1997) with fixed time-step size is used. The scheme is stable and efficient, and can be used for all rigid body systems without constraints. This integration scheme is recommended and it is the default.

`BULIRSCH`

A Bulirsch-Stoer method (Stoer and Bulirsch, 1980; Press *et al.*, 1992) with adaptive step size control and polynomial approximation is used. The accuracy requirements for the total energy and the coordinates are treated independently in the error estimator. This method is suitable for almost all more-or-less smooth potential energy surfaces.

`RUNGE-KUTTA`

A fifth-order Cash-Karp Runge-Kutta method (Press *et al.*, 1992) with adaptive step size control is used. The accuracy requirements for the total energy and coordinates are treated independently in the error estimator.

It is possible to control a running simulation using a *Command File*. The command

`COMMANDFILE file`

specifies the name of the command file that is to be used. It is a simple text file containing one keyword per line. After the keywords have been processed the *Command File* is deleted automatically, unless the `KEEP` keyword is present. The default file is `mdcommand.txt`.

Keywords for the *Command File* are

`STOPLOOP`

The simulation will end as soon as the current energy loop is finished.

`SKILOOP`

As soon as the current energy loop is finished the simulation will leave out the next energy and jump to the next higher value directly. This keyword can be used more than once to jump over several energies.

`KEEP`

The *Command File* will not be deleted after it has been read, so all commands will be taken into account after every energy loop.

15.5 Plotting configurations

The trajectory can be stored in a file which can be used to visualise the time evolution of the molecular motion. The command is

`PLOT options`

and the options are:

`EVERY t`

The minimal time-interval (fs) at which the geometries are stored. The default is 1.0. Because of the adaptive step size in the time-variable the configurations are not exactly equidistant in time. The use of this option automatically causes storage of the geometries in a file.

`FILE file`

The name of the file in which to save the geometries. The output is in *xyz*-format and includes time information in the title. The default *file* is `mdrun.plot`.

TITLE *title*

The title of the geometry data. It should be enclosed in quotes if it contains spaces. The time corresponding to the configuration is always appended to the title. The default title is MD-Plot.

15.6 Quenching

A potential energy surface can be explored by performing a high-temperature molecular dynamics simulation, and quenching the system from time to time — that is, by optimizing the geometry, in effect reducing the temperature to zero so that the system falls into the nearest minimum. Afterwards the simulation continues from the configuration from which it was quenched, i.e., as if the quench had not happened.

The optimization procedure (Wales *et al.*, 1995) is controlled by a preceding OPTIONS section, as for the OPTIMIZE environment. The quenched energies and optionally the geometries are written to specified files.

```
The command is
QUENCHING
  ⋮
  options
  ⋮
END
```

and the options are:

EVERY *t*

The minimum time-interval at which the trajectory is quenched. Because of the adaptive step size in the time-variable the calculations are not exactly equidistant in time. The default of *t* is 0.0 (no quenching).

FILE *file*

The file where the optimized energies are stored. The output is in columns in the order *time-variable* *total energy*

The default *file* is `mdrun.quench`.

GEOMETRY *geofile*

If this file is specified the optimized geometries are stored in *geofile.xyz*. The output is in xyz format, with one frame for each quenched geometry, and includes energy information in the title. The default *geofile* is `mdrun.qgeo`.

TITLE *title*

The title used in the geometry data file. The corresponding energy is always appended to the title. The default title is MD-Quenching.

GLOBALMIN *gmfile*

If this file is specified the lowest-energy configuration of all quenched geometries is saved separately. This can be used to look for the global minimum during a simulation. The configuration is saved in xyz format (*gmfile*) and in a dump-file (*gmfile.dump*) which is suitable for the RESTART command.

15.7 Dumping and restarting a trajectory

DUMP *file*

Dump the final positions to the specified file at the end of the trajectory, and the velocities to a file *file.moldyn*.

MDRESTART [*file*]

This command causes an old Molecular Dynamics trajectory to be restarted. It is different from the RESTART command because it also expects the information about the initial velocities which are stored when the DUMP command is used in the MD environment. The specified file contains the old trajectory. To save all information about a trajectory the DUMP command should have been used in a previous integration. In this case the configuration will have been stored in *file* and is accessible to the RESTART

command. In addition the velocities will have been stored in *file.moldyn*. Both files are necessary for the RESTART command.

15.8 Statistics

```
STATISTICS
:
options
:
END
```

This command controls the calculation of the statistical observables in the Molecular Dynamics run. All observables are time averages and should converge during the integration.

EVERY *t*

The minimal time-interval at which the statistical observables are saved to file. The default is 1000 fs. The frequency of updating the statistics is not affected by this value but is determined by the ENERGY command. Because of the adaptive step size in the time-variable the output is not exactly equidistant in time. The use of this option automatically causes storage of the statistics in a file.

FILE *file*

The name of the file in which the statistical observables are stored. The output is in columns and depends upon whether the ALL option is set or not. The default file is *mdrun.stat*.

ALL

This option produces more detailed printing. Usually the order of the columns in the statistics file is

$$t \quad \langle E_{\text{tot}} \rangle \quad \sigma(E_{\text{tot}}) \quad \sigma_{\text{rel}}(E_{\text{tot}}) \quad T_{\text{kin}} \quad \sigma(T_{\text{kin}}) \quad \sigma_{\text{rel}}(T_{\text{kin}}) \quad \delta_{\text{Lind}} \quad \sigma(\delta_{\text{Lind}}) \quad \sigma_{\text{rel}}(\delta_{\text{Lind}})$$

With the ALL option the additional columns are the observables

$$T_{\text{vib}} \quad \sigma(T_{\text{vib}}) \quad \sigma_{\text{rel}}(T_{\text{vib}}) \quad T_{\text{rot}} \quad \sigma(T_{\text{rot}}) \quad \sigma_{\text{rel}}(T_{\text{rot}}) \quad P_{\text{lin}} \quad \sigma(P_{\text{lin}}) \quad \sigma_{\text{rel}}(P_{\text{lin}}) \quad P_{\text{ang}} \quad \sigma(P_{\text{ang}}) \quad \sigma_{\text{rel}}(P_{\text{ang}})$$

The file is an ordinary text file with columns separated by spaces. The notation is as follows:

- $\langle x \rangle$ abbreviates the time average of x , $\sigma(x)$ the standard deviation and $\sigma_{\text{rel}}(x)$ the relative standard deviation compared to the average of the time averages.
- t is the time, E_{tot} the total energy and T_{kin} the kinetic temperature. P_{lin} and P_{ang} are the overall linear and angular momentum of the system, respectively, and are given in atomic units.
- The Lindemann criterion ([Rose and Berry, 1992](#)),

$$\delta_{\text{Lind}} = \frac{2}{N(N-1)} \sum_{i < j}^N \frac{\left(\langle r_{ij}^2 \rangle - \langle r_{ij} \rangle^2 \right)^{\frac{1}{2}}}{\langle r_{ij} \rangle},$$

is calculated from the fluctuations of the site centre of mass distances r_{ij} .

- The *vibrational* and *rotational temperature*, T_{vib} and T_{rot} , respectively, are defined as the kinetic temperature calculated only from the rotational and vibrational degrees of freedom, respectively. Any difference between the two values results from the internal use of non-reduced coordinates.

15.9 Time step control

```
TIME
:
options
:
END
```

The command sets the integration time of the Molecular Dynamics simulation and defines the boundaries

for the adaptive step size control. Times are in femtoseconds (fs). MINDT, MAXDT and MAXITER are only required for integrations using an adaptive step-size scheme.

START *starttime*

Specifies the starting time-point of the integration. The default is 0.0.

STOP *endtime*

Specifies the finishing time-point of the integration. The default is 1000.0.

TIMESTEP *dt*

The length of a time step (in fs) for the symplectic integration scheme. The default is 0.0.

MINDT *min*

MAXDT *max*

Lower and upper limits for the adaptive time-step size in the integration routine. The default lower limit is 0.0001 fs. Usually this limit should be reached only if the potential energy surface is too rough or the integration errors are too big. The default upper limit is 1000.0 fs. This value can control the global integration accuracy but this should normally be done by the ACCURACY command so that this limit is never reached.

MAXITER *iter*

Upper limit of the internal integration step number to prevent the program running for ever. The default is 10^8 .

16 Calculation of the second virial coefficient

ORIENT can calculate pressure and dielectric virial coefficients. The second pressure virial coefficient $B(T)$ is calculated by integration of the standard formula (Gray and Gubbins, 1984, p. 210):

$$B(T) = -\frac{1}{2} \int \int [\exp(-U_{12}/kT) - 1] d\Omega d^3\mathbf{r} + \frac{\hbar^2}{24(kT)^3} \left[\frac{\langle \mathbf{F}^2 \rangle_0}{M} + \sum_{\alpha} \frac{\langle T_{\alpha}^2 \rangle_0}{I_{\alpha\alpha}} \right].$$

The first term of this expression is the classical expression; the second is the quantum correction to order \hbar^2 , involving the mean square force $\langle \mathbf{F}^2 \rangle_0$ and the mean square torques $\langle T_{\alpha}^2 \rangle_0$ on one molecule. M is the molecular mass and the $I_{\alpha\alpha}$ are the molecular principal moments of inertia.

Two options are available for the integration over orientations. The molecules are positioned with their respective origins at $(0,0,0)$ and $(0,0,R)$. The relative orientation is defined by three Euler angles (α, β, γ) for each molecule, but α_1 can be fixed at zero. This leaves five angles for non-linear molecules, and three for linear ones, where the integrand is independent of γ_1 and γ_2 . For the linear case, a Gauss-Legendre integration scheme over β_1 and β_2 , with uniform integration over α_2 , is most efficient, but for non-linear molecules a Sobol quasirandom sequence (Press *et al.*, 1992) of relative orientations of the two molecules is recommended, to provide a uniform coverage of the orientational space. For each relative orientation Ω , the Mayer function $f(R, \Omega)$ is evaluated for distances R from some maximum value R_{\max} by steps ΔR downwards. This process stops when R reaches a specified minimum value R_{\min} or when the Mayer function becomes close to -1 . (This procedure overcomes a problem that arises with many model potentials, where the potential becomes large and positive as R decreases but then decreases to spurious large negative values for very small values of R .) The values of the Mayer function at each R are accumulated to obtain the orientational integral, and these values are then integrated numerically over R to obtain the virial coefficient. The values of the Mayer function are also written to a file (the 'Mayer-function file', see below), and can be read in again by a subsequent job.

The data file must contain specifications of two molecules and the interactions between them. The molecular positions and orientations must be given in the form

MOLECULE *name* at x1 y1 z1 rotated alpha1 beta1 gamma1

The name of the molecule is arbitrary but the names of the variables must be as shown. Similarly the position and orientation of molecule 2 must be given using the variables x2, y2, z2, alpha2, beta2 and gamma2. There must be a Variables declaration of these variables at the beginning of the program, but the values provided are irrelevant since the program changes them in the course of the integration.

The integration is carried out by a VIRIAL command, which has the form

```
VIRIAL qqquad subcommands END
```

with the following sub-commands:

```
CLASSICAL
```

Calculate the pressure virial coefficient, omitting quantum corrections. This is the default.

```
QUANTUM
```

Calculate the pressure virial coefficient, including quantum corrections. (It takes considerably longer to do this.)

```
DIELECTRIC [SWITCH  $R_s$ ] [OMIT INDUCTION]
```

Calculate the dielectric virial coefficient. The formulae used are taken from [Joslin and Goldman \(1993\)](#); there are different formulae for the short-range and long-range regions, but their ranges of validity overlap. The SWITCH parameter R_s specifies where the switch from one formula to the other occurs. It should be large enough for the molecules not to overlap significantly at any relative orientation. The numerical values for both formulae are written to the Mayer-function file, so they can be inspected and if necessary the radial integration can be repeated with a different switch point. (See the Integrate option below.) The values from the short-range and long-range formulae should be approximately the same over a range of values of R . Differences are due in part to numerical errors in the angular integration, but for small R the long-range formula is invalid. At long range both formulae are formally valid, but the short-range formula is much more susceptible to numerical error. There should be a range of values of R where the formulae approximately agree, and the switch between the formula may be taken anywhere in this range without affecting the calculated result very much. OMIT INDUCTION specifies that the induction term, which is usually small, is to be omitted. (This option does not omit the effects of induced dipole moments, which are often large, but the effects arising from the difference between the dimer polarizability and the tensor sum of the polarizabilities of the individual molecules.)

```
STEPS BY  $\Delta_1$  TO  $R_1$  BY  $\Delta_2$  TO  $R_2$  ...
```

Specifies the radial integration range and stepsize (see above). The radial integration can be broken into a number of ranges, using a step-size of Δ_1 from $R = 0$ out to $R = R_1$, then a step-size of Δ_2 out to $R = R_2$, and so on. A smaller step-size is needed in the region where the interaction energy is varying rapidly, and a larger step can be used at greater distances.

```
SOBOL [[ORIENTATIONS]  $N$ ] [[INDUCTION]  $N_{\text{ind}}$ ] [DISPLAY  $n$ ]
```

Specifies a Sobol integration scheme. The default number of orientations is 2048, but this will probably not be enough for an accurate result. The number given is rounded up to a power of 2. If N_{ind} is specified, the induction term, which is usually small and is quite time-consuming to evaluate, is calculated for only N_{ind} orientations (rounded up to a power of 2). The results are displayed after n orientations (rounded up to a power of 2), and thereafter at each power of 2. This makes it possible to specify a large number of orientations and to monitor convergence while the calculation is running, stopping it when it has converged. If the display option is omitted, results are displayed starting at 512 orientations. Display 0 specifies that only the final results are to be displayed.

```
GAUSS [ALPHA  $n_\alpha$ ] [BETA  $n_\beta$ ] [GAMMA  $n_\gamma$ ]
```

Use a Gauss-Legendre integration scheme over the β coordinates, as described above, and uniform integration over α_2 , γ_1 and γ_2 . n_α , n_β , and n_γ are the number of integration points in each angle. The default values are 7, 7, and 1 respectively, which will normally give a reasonable result for linear molecules. For non-linear molecules, n_γ must be increased. If the molecule has a C_n symmetry axis, n and n_γ should have no common factor. For a molecule with a C_3 axis, for example, 3-point integration over γ will simply evaluate the same contribution 3 times. Here 2-point integration would give a better result (but 5-point or 7-point would be better still).

```
CHECK POTENTIAL [MAX  $E_{\text{max}}$ ] [QUIET]
```

Some potentials, notably the exp-6, rise to a maximum on the repulsive wall and then decrease, eventually approaching $-\infty$ as $R \rightarrow 0$. This leads to spurious large positive contributions to the Mayer function at short range, and will often cause a floating-point exception. If check potential is specified, the program checks for cases when, after reaching E_{max} or more, the potential decreases as R decreases. If this happens, the Mayer function is set to -1 (corresponding to a large positive energy) for all smaller values of R . The value of E_{max} may be specified (in the current energy units) if required, but the default

value of 0.01 a.u. should normally be satisfactory. The program reports any anomalies on standard output unless `quiet` is specified, and prints a summary of them in any case. If many anomalies are reported, it is an indication that the potential is unsatisfactory. Damping the induction and dispersion terms will usually overcome the problem.

```
TEMPERATURES  $T_1$   $T_2$   $T_3$  ...
```

Temperatures, in Kelvin, for which the virial coefficient is to be evaluated. This command may be repeated if many temperatures are required. It is most efficient to do all temperatures at once.

```
MAYER file
```

Specifies, optionally, the name of the file to which are written values of the Mayer function, averaged over orientations, together with the quantum correction terms if they are calculated. Default is `mayer.out`.

```
VIRIAL file
```

Specifies, if required, a file to which the values of the virial coefficient are written, with the quantum corrections if specified. The default is to write the virial coefficient results to standard output only.

```
SAVE file [n]
```

(For Sobol integrations only.) Save the state of the calculation to the specified file after every n orientations, so that the calculation can be restarted later. Default $n = 1024$.

```
RESTART file
```

Restart a calculation that was previously saved in the specified file.

```
INTEGRATE file
```

Read in Mayer function data from a file produced in a previous calculation, and carry out the radial integration. This requires little computation time, and is particularly useful in calculations of the dielectric virial coefficient.

17 Optimizing potential parameters

Most of the parameters defining an ORIENT potential are obtained from *ab initio* calculations on the individual molecules, and do not need to be fitted to data describing the interaction. The repulsion parameters, however, do need to be fitted in this way, and so may dispersion and damping function parameters. Procedures for doing this are now available in ORIENT, using *ab initio* results obtained using either the InterMolecular Perturbation Theory (IMPT) of [Hayes and Stone \(1984\)](#) or the Symmetry-Adapted Perturbation Theory (SAPT) of [Jeziorski et al. \(1994\)](#).

17.1 Repulsion parameters

The procedure for repulsion is as follows. First, define the two molecules concerned in the usual way. The distributed multipoles should be determined *ab initio* for each molecule, using the same basis set as in the perturbation theory calculation. In the case of IMPT, SCF multipoles should be used; for SAPT, they should be CCSD values; and for SAPT-DFT they should be DFT values calculated with the same functional as for the energies. The repulsion should be defined using the PAIRS section in the usual way; the parameters specified will be used as starting values for the optimization. Do not include any dispersion or induction. (See below for fitting dispersion damping parameters.)

A new data section is needed to specify the *ab initio* values to be used for fitting. It takes the form

```
FIT  $v_1$   $v_2$   $v_3$  ... ENERGY [WEIGHT]
 $x_1$   $x_2$   $x_3$  ...  $E$  [ $W$ ]
 $x_1$   $x_2$   $x_3$  ...  $E$  [ $W$ ]
:
END
```

Here v_1 , v_2 , *etc.*, are the names of variables used to define the geometry, while ENERGY and (optionally) WEIGHT appear literally as given. There follows a table of geometries, each specified by the values x_1 , x_2 , *etc.*, of the geometry variables, followed by the value E of the energy for that geometry, and, optionally,

a weight. The energy given should be the total first-order energy. The weight is the statistical weight to be assigned to the energy point in the fitting process; if it is omitted, or if the `Weight` keyword is omitted, a weight of 1 is assigned. (But see below for the `WEIGHTS` command in the `ADJUST` data section.) If a file is imported from elsewhere, the table may include one or more irrelevant columns. Rather than edit out each such column, it is sufficient to include the keyword `Ignore` in the header for each one.

Blank lines in the table are ignored, as are lines beginning `NOTE` or `'!'`. The subcommand

```
ECHO [ON|OFF]
```

may occur anywhere in the table to switch on or off the echoing of subsequent data lines to the output. Default is `ON`.

The fitting process is specified by an `ADJUST` data section. This starts with a `VARIABLES` subsection, which takes the same form as a `PAIRS` section, but instead of parameter values it takes string arguments, which are arbitrary names (maximum 12 characters) for those parameters that are to be optimized. A null string, or either of the strings `"0"` or `"-"` denotes that the corresponding parameter is to be kept constant during the optimization. Parameters listed in the `PAIRS` section need not appear in the `ADJUST` specification; any not mentioned will be kept constant with the value given. The `ADJUST` section may be repeated, to allow other parameters to be fitted, possibly while keeping previously-fitted parameters constant. However all parameters specified under `ADJUST` must have been listed in the `PAIRS` section, possibly with zero values.

The following may also occur in an `ADJUST` section:

```
PENALTIES
F n1 c1 n2 c2 ...
F n1 c1 n2 c2 ...
:
END
```

Each line of a `Penalties` declaration specifies a penalty function that is to be added to the sum of squares being minimized. This can be used to overcome problems of linear dependence or near-dependence of the parameters, or to impose conditions that they should satisfy. The penalty function takes the form $F(c_1 p(n_1) + c_2 p(n_2) + \dots)^2$, where $p(n)$ is the parameter whose name was given as n in the `ADJUST` section. There is a dummy parameter called `"0"` whose value is 1; a coefficient c_0 can be specified for it in order to incorporate a constant into the penalty function. F is a multiplying constant, and is an energy value in the current energy units. An example of a penalty function declaration is

```
Penalties
1.0 rhoA 1.0 rhoB -1.0
1.0 alphaA 1.0 alphaB -1.0
1.0 rhoA 1.0 rhoC -2.0 rhoB 1.0
end
```

This specifies that parameters `'rhoA'` and `'rhoB'` are to be kept approximately equal, as are parameters `'alphaA'` and `'alphaB'`, while parameter `'rhoC'` is to be kept approximately equal to the average of `'rhoA'` and `'rhoB'`. The value of F can be used to modify the strength of the penalty without the need to change all the coefficients.

An alternative, simpler, form of penalty may be defined using anchors:

```
ANCHORS
n1 a1 w1
n2 a2 w2
:
END
```

Here n_i is a parameter name, a_i is its anchor value, and w_i is the weight of this anchor. This defines a penalty $w_i(p(n_i) - a_i)^2$, where $p(n_i)$ is the value of parameter n_i . This form of penalty is useful where an approximate value of the parameter has been found by some means and the fitted value is to be prevented from departing too far from the approximate value. The weight w_i is an energy in the current energy units.

The total number of penalties may not exceed the number of parameters.

Other items that may occur in an ADJUST section include the following:

```
STEP s
ITERATIONS n
CONVERGE c
WEIGHTS [ EXP kT | RELATIVE | LOG | MIXED e ]
```

The STEP value is the maximum amount by which a parameter may change in one optimization step, and the ITERATIONS value is the maximum number of iterations. The optimization is considered to have converged when the r.m.s. derivative of the weighted sum of squares of the residuals with respect to the parameters falls below the convergence criterion *c*. WEIGHTS specifies the weighting of the data points. It over-rides any explicit weights specified in the FIT data section. The possible weighting schemes are:

- Exponential weighting specifies that points with energy E are to be given weight $\exp(-E/kT)$, where kT is an energy in whatever energy units are currently being used. A value of $2.5\text{mH} = 790\text{K}$ is usually satisfactory.
- Relative weighting specifies that points with energy E are to be given a weight proportional to $1/E^2$. This is equivalent to assuming that the uncertainty in an energy E is proportional to E .
- Logarithmic weight assigns a weight $|\log(|E|)|$ to a point with energy E . This gives a higher weighting to points with smaller energies, but in a much milder fashion than relative or exponential weighting.
- Mixed weighting gives a weight $1/E^2$ to points with $|E| > e$, and a uniform weight $1/e^2$ to points with energy $|E| < e$. The value of e should be specified in the current energy units.

17.2 Dispersion damping parameters

Dispersion is fitted in a similar way to repulsion. Here the energy values provided in the FIT data section are *ab initio* dispersion energies. Multipole moments must not be specified (use 'Default rank -1'), and the PAIRS section must specify only dispersion, not repulsion. The dispersion specification may include C_n coefficients and their anisotropies in the usual way, and damping functions may be specified as required. Again, the VARIABLES part of the ADJUST data section takes the same form as the PAIRS section, with arbitrary non-null names to denote the values that are to be optimized and null strings to denote any that are to be kept fixed.

An additional item that may be found necessary in fitting dispersion parameters is

```
SCALE
  n1 s1
  n2 s2
  ⋮
END
```

The parameter with name n_1 is to be scaled by the factor s_1 . That is, the variable used in the fitting procedure has a value s_1 times smaller than the physical parameter n_1 . Fitting procedures are much more efficient if all the variables and their derivatives have similar magnitudes, so a suitable value for the scale factor of any parameter is an estimate of its likely value. However this is a very crude prescription, and the scale factors may need to be adjusted in the light of experience. Note that the maximum step-size applies to the scaled values. A parameter such as a C_8 coefficient, which may have a value of 50,000 in atomic units, will be optimized very slowly indeed with a typical step-size of order 1 unless it is scaled, and scaling will allow larger steps to be taken.

18 Graphical displays

Versions 4.6 and later of the program can display some calculated properties graphically. If compiled from source, this part of the program needs a Fortran 2003 or later compiler and the OpenGL libraries. See the INSTALL file for details.

In principle, any scalar or vector field that can be calculated on a grid of points on the molecular surface can be displayed. At present the program can display the energy of interaction between a probe atom and a specified molecule at a grid of points on the molecular surface, or the difference between two such energies. Two special cases are predefined: in the first case the probe is a unit charge, and in the second case it is a set of unit dipoles in the x , y and z directions. In both of these cases, the only interaction considered is that with the charge distribution of the specified molecule, so the result is either the electrostatic potential or the electric field. Otherwise, the probe atom is defined in the usual way (§6.3) and the complete interaction of the probe with the molecule is calculated. In either case the result is an energy, in atomic units internally, but it can be displayed in whatever units are required.

This part of the program has been completely re-written in version 4.9, to make it more logical, more versatile and more convenient to use. Unfortunately this means that data files for earlier versions are not compatible with the new version.

The command usually takes the form

```
DISPLAY
  grid definition
  map definitions
  display options
  SHOW section
END
```

18.1 Setting up the grid

The grid definition specifies the grid on which the display is constructed. (It is also possible to import a previously-defined grid; see below.) This section takes the form

```
GRID
  NAME name
  TITLE "descriptive title"
  MOLECULE mol-name
  [SMOOTHED  $s$  | GOLDBERG]
  STEP  $t$  [A|B]
  RADIUS [SCALE  $s$ ] [ADD  $a$ ]
END
```

The name of the grid should distinguish this grid from others that might be defined elsewhere. The molecule should have been defined previously in the usual way, with atom types specified. The next item optionally specifies how the grid is to be constructed, but the default SMOOTHED procedure uses the marching cubes algorithm to construct a surface made up of modified van der Waals spheres with the joints between them smoothed over, and this is usually satisfactory. The optional parameter s specifies the degree of smoothing of the joints between van der Waals spheres, but the default of 0.5 is usually satisfactory. An alternative is the GOLDBERG method, which uses a Goldberg icosahedron, obtained by shrinking an icosahedron onto the molecular surface and subdividing each triangle until the required resolution is obtained. The default is much more robust, and is recommended.

The van der Waals radii are modified, if required, by the RADIUS command, in which the radii can be scaled or incremented by specified amounts. If both SCALE and ADD are specified, they can be given in either order; the operations are carried out in the order given. The STEP command specifies the nominal separation between points of the grid, in angstrom or bohr if A or B is specified, or otherwise in the current length unit. In practice most separations will be somewhat smaller than this and some may be

larger. For a large molecule or a fine grid it is possible to increase the maximum number of grid points from the default of 50,000 by using the command

```
ALLOCATE POINTS n
```

but such a large grid and the maps defined on it will take a long time to construct, and usually a better approach is to reduce the number of points by increasing the STEP.

Following the grid definition the grid can be exported to a file for later use, though this is less likely to be necessary with the revised DISPLAY module. The command is simply

```
EXPORT [GRID | POINTS] file
```

Here GRID specifies that the complete grid information is to be exported, starting with header lines that specify the name and other characteristics of the grid. POINTS specifies that only the list of grid point positions is to be output, with minimal header information. This can then be imported into another program, for example a CAMCASP energy-scan calculation.

A previously exported grid, or a grid generated by another program, such as a CAMCASP isodensity grid, can be imported from a file:

```
IMPORT GRID file [map-name]
```

Usually the grid information and the map data (see below) are held in separate files, but it is possible to read a file that contains map data along with the points list. In this case a *map-name* can be specified; the map data will then be available to the program under this name. The optional *mapname* specifies the name of the map that is to hold the map data in the file, if any. If no *map-name* is given but the file contains map data, it is read into a map with a name that is either taken from a MAP-NAME header in the grid file, if present, or is the same as the name of the grid.

The following command can be used to print out the principal characteristics of the grid, perhaps to check that they are as expected or simply to list the details of an imported grid:

```
DESCRIBE GRID
```

18.2 Defining maps

One or more map definitions follow the grid definition. Each specifies a scalar or vector function (or both) to be evaluated at each of the grid points. The form of the map definition is:

```
MAP name  
  MOLECULE name  
  TITLE "descriptive title"  
  function  
END
```

The name of the map should distinguish between this and other maps defined in this job. The *function* is one of

```
POTENTIAL  
FIELD [MAGNITUDE]  
PROBE molecule  
DIFFERENCE map1 [minus] map2
```

The POTENTIAL function is the electrostatic potential at the grid point arising from the specified molecule. Evidently this molecule must correspond in position and orientation to the molecule used to define the grid, but it need not be identical. For example, it could be a point-charge model of the same molecule. However it must have been defined in the usual way earlier in the Orient data file. The FIELD function is the electrostatic field vector at the grid point. In this case, a scalar function is calculated too, and is used to define the colour associated with the grid point. The default scalar function is the electrostatic potential; FIELD MAGNITUDE specifies that the scalar is the magnitude of the electric field.

If PROBE is specified, it is necessary to define and specify a probe molecule (usually an atom), and the function associated with each grid point is the energy of interaction between the probe and the target molecule when it is at that grid point. In this case all contributions to the energy that have been specified

in the data file are included. The probe can be a point charge; this just yields the electrostatic potential if the target molecule carries only multipoles, but it can be used to explore the induction energy if distributed polarizabilities have been defined, or the dispersion energy if the probe is a polarizable atom and dispersion coefficients have been defined.

A map can be exported to a file using the command

```
EXPORT MAP file
```

and a previously-defined map may be imported:

```
IMPORT MAP file [mapname]
```

If a mapname is given, it will replace any name recorded in the file.

Finally, a DIFFERENCE map is the difference between two previously-defined maps, and can be used to explore, for example, the difference between the potential arising from a high-order multipole description and a point-charge or simplified multipole description of a molecule.

As for the grid, the principal details of a map can be printed out:

```
DESCRIBE MAP map-name [map-name ...]
```

18.3 Displaying maps

Once the maps have been defined, one of them can be displayed. Unfortunately it is only possible to display one map in any one job, but if several of the maps are to be displayed the only part of the data file that needs to be changed is the SHOW section:

```
SHOW map-name
  options
END
```

Indeed, if several different maps are to be displayed for the same molecule and grid, a recommended procedure is to define all the required maps in one data file and just add the necessary SHOW section for each one in turn. The script `orient_display.py` in the Orient bin directory may be used to do this automatically. Normally, up to 12 maps may be defined, but this number may be increased:

```
ALLOCATE MAPS n
```

The following options can be used, either before the SHOW section (if they are to apply to any SHOW section) or within it. Note however that some of these features can be changed using the mouse and keyboard once the display is on the screen. See §18.4 for details.

```
TITLE "descriptive title"
```

Title for the display window. Default is to use the map title.

```
HEIGHT np
```

Display the picture in a window with height np pixels. Default is 500.

```
VIEWPORT d
```

Display the molecule in a cube extending from $-d$ to d in x , y and z . Default is $d = 10$ bohr.

```
VIEWPOINT x y z [UP x' y' z']
```

View the system from the direction (x,y,z) , with the direction (x',y',z') vertical on the screen. The default viewpoint is from the direction $(0,0,-1)$ with the vertical $(0,1,0)$. This can be changed using the mouse.

```
BACKGROUND { GREY | RGB | RGB255 | RGBX } colour
```

Specify the background colour for the display. Colours are specified as a single number between 0 and 1 (for grey-scale), as a set of 3 numbers between 0 and 1 (RGB), as a set of 3 numbers between 0 and 255 (RGB255) or as a single string of 6 hex digits (RGBX) the first pair of digits specifying the red value, the

second pair the green and the third pair the blue. The default background colour is white.

A ball-and-stick representation of the molecule can be displayed by

```
BALL-AND-STICK [molecule]
```

The bonding must be specified, either in the molecule definition (p. 9) or by using the EDIT command (p. 20). The colours used for atoms and bonds are defined in the TYPES command, either by reference to the atomic number (for $Z \leq 18$) or explicitly.

The following options apply only to scalar fields:

```
COLOUR-SCALE [ ON | OFF ] [MIN a] [MAX b] [BOTTOM b] [TOP t] [UNIT unit]
```

ON specifies that a colour scale is to be displayed at the side of the picture; this is the default. OFF turns the scale off. The colour scale is to run from a to b . Values are in the specified unit, if given, or in the current energy unit otherwise. (Specify UNIT eV to specify electrostatic potentials in volts.) Normally the colour scale bar runs from a to b , but if TOP is specified, the value t corresponds to the top of the colour scale bar; similarly b corresponds to the bottom if BOTTOM is specified. The colour bar is then grey except between the limits a and b of the colour scale. For example,

```
Colour-scale min -10 max -5 top 0
```

might be used for a display of dispersion energies with values between -5 and -10 , the top of the scale bar being set at 0. If the MIN and MAX values are not specified, the minimum and maximum of the displayed dataset are used.

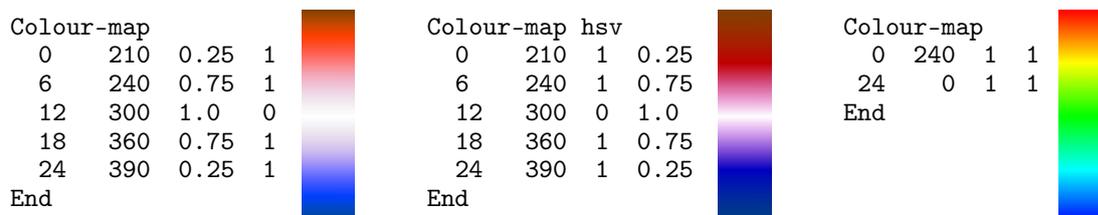
Lines can be drawn across the colour scale at suitable positions using the TICKS command:

```
TICKS [unit] t1 t2 ...
```

The tick positions are assumed to be in the current units unless some other unit is specified.

```
COLOUR-MAP [ HLS | HSV ]
:
n hue lum sat [or hue sat val]
:
END
```

Define a colour map for the colour scale. The colours can be defined by hue, luminance and saturation (HLS, the default) or by hue, saturation and value (HSV). Luminance and value are related but not the same; less obviously, saturation is different in the two schemes. See “HSL and HSV” in Wikipedia for an explanation. In either case, the colour map is defined by specifying the colours at several levels, denoted by n , between the minimum and maximum of the colour scale. The first level is numbered 0, and there may be up to 32 levels, but not all of them need to be specified — the program interpolates linearly to fill in missing values. For example, two of the scales below run from dark greenish blue (hue 210, low luminance) through blue (hue 240) and white (saturation 0, hue irrelevant) to red (hue 0 or 360) and dark brownish red (hue 30 or 390, low luminance). They illustrate the differences between HLS (left) and HSV (centre). The right-hand picture shows the conventional ‘rainbow’ scale, and the data for it illustrates that the interpolation can run from high values to low as well as low to high.



Hue values are interpolated between the numbers given and then taken modulo 360, to allow values to be interpolated past the ends of the scale, normally taken to be 0 and 360. The program doesn't provide a default colour map, because the map should be designed to suit the problem. However the left and centre maps are suitable for displaying potentials, or differences between two calculations of the same quantity. In these cases zero should be in the middle of the scale. The rainbow map is often used but is not usually a good choice. See [Rogowitz and Treinish \(1996\)](#) for a very useful guide to the design of colour maps,

and for reasons for not using the rainbow map.

```
COLOURS { GREY | RGB | RGB255 | RGBX }  
:  
  colours  
:  
END
```

This is another way to define the colours used in the colour scale. It is available for backwards compatibility, but is deprecated. Colours are specified one per line, as a single number between 0 and 1 (for grey-scale), as a set of 3 numbers between 0 and 1 (RGB), as a set of 3 numbers between 0 and 255 (RGB255) or as a single string of 6 hex digits (RGBX) the first pair of digits specifying the red value, the second pair the green and the third pair the blue.

```
CONTOURS level level ... [unit]
```

Add contours to the surface at the levels given, assumed to be in the current unit unless a different unit is specified. It will usually be necessary to specify eV for potential plots.

The following option applies to vector displays only:

```
SCALE s
```

Apply the scale factor s to the length of the vectors. For electric field displays, the values are in atomic units (i.e. E_h/ea_0), and a scale factor in the range 100–1000 or thereabouts is likely to be suitable. The default value is 200.

18.4 Interacting with the display

It may take a while for the display to appear, as the field to be displayed has to be computed at each grid point. Once the display has appeared, the mouse can be used to rotate the displayed molecule about axes in the plane of the display, by dragging with the left button, to increase or decrease its displayed size (actually by decreasing or increasing the viewport) by dragging with the right button, and to rotate about the z axis (perpendicular to the plane of the display) by dragging with the middle button.

There are also several keys that affect the display. Letters may be in upper or lower case.

? prints this summary on stdout (so not to the screen unless output is to the screen).

F hides or restores the vector display, if present.

H hides or restores the surface and vector display. This allows a clearer view of the ball-and-stick model of the molecule.

M prints the rotation matrix and angle-axis rotation for the current view.

Q causes the program to quit.

R resets the display to its original size and orientation.

S starts or stops the rotation of the display around the vertical axis.

T step-by-step rotation.

U deals with an oddity of the display. The program attempts to apply the colours to the outside of the surface, but for some reason it sometimes applies them to the inside, leaving the outside surface transparent. While this gives a clearer view of the ball-and-stick model, it gives a confusing display of the surface. The U key switches between these states.

– This has two functions. If the display is rotating, it reverses the direction of rotation. For a vector display its function is more important. The vectors are drawn with their tails attached to the grid

point, but the inward-pointing vectors would then be hard to see, so they are not shown. The `-` key changes this, so that the head of each vector is attached to the grid point, and the outward-pointing vectors are suppressed and the inward-pointing ones are shown.

The following keys affect both the speed of rotation, if the display is rotating, and the length of the vectors in a vector display.

digit A single digit 1–9 multiplies the rotation speed and the vector length by the value of the digit.

/ Changes this so that a single digit divides by the value of the digit.

* Restores the effect of digits to multiply speed or vector length.

0 (Zero) suppresses the vector display altogether. Repeating this key restores it.

Note that the display can be copied to a file using the `import` command from `imagemagick`. Enter e.g.

```
import <filename>.png
```

on the command line. The cursor will change to cross-hairs. Click the mouse in the display window to copy the content of the window to the file in `png` format. Use other file extensions for other formats.

19 Using ORIENT as a subroutine for other programs

ORIENT has been designed so that you can easily use it as a subroutine to provide values of interaction energies for use in other calculations. It has been used in this way, for example, to carry out Diffusion Monte Carlo calculations of intermolecular vibrational wavefunctions ([Meredith and Stone, 1998](#)).

The procedure for doing this is as follows. The main program ‘`runorient`’ provided with the package needs to be replaced by your own program. This must begin with a set of `USE` statements of the form

```
USE <module>, ONLY : <names>
```

The purpose of these statements is twofold. First, some modules need to be referenced in this way so that subroutines and variables defined in them are available to your program. These subroutines and variables are listed by name following the colon. Secondly, some modules need to be referenced in your main program because they contain variables whose values need to be preserved between calls to ORIENT subroutines. If you do not `USE` the module in your main program, their values may be lost between calls.

The first statement of your program, apart from any initialization that you wish to do, should then be

```
CALL ORIENT
```

This will cause ORIENT to read its data from standard input in the usual way, in order to define your molecules and their properties. It is also possible to carry out any of the normal ORIENT procedures. One such procedure that you might want to use is the `show variables` command, which will list the variables (as defined in the data file) that specify the positions and orientations of the molecules. This part of the data file should be terminated with `Finish`. ORIENT will then exit, returning to your program. You can now carry out whatever further calculations you wish, calling ORIENT subroutines as required. An example is shown in Fig. 11 (p. 59). It probably isn’t necessary to `USE` all of the modules shown in Fig. 11, but it will do no harm to do so. The subroutine ‘`energy`’ is provided specifically for this application — it isn’t used in the ORIENT program itself. The variables `es`, `er`, `eind`, `edisp` and `etotal` return the electrostatic, exchange–repulsion, induction, dispersion and total energies respectively.

```

PROGRAM my_prog
USE parameters, only :      IMPLICIT NONE
USE indparams, only :      DOUBLE PRECISION :: es,er,eind,edisp,etotal
USE alphas, only :         DOUBLE PRECISION :: values(40)
USE binomials, only :
USE consts, only :         ! Read data for Orient
USE derivs, only :         call orient
USE geom, only :
USE global, only :         ! Initialize your calculation
USE induction, only :     ...
USE interact, only :
USE io, only :             ! Program loop
USE labels, only :        do while ...
USE lattice, only :       ! Set molecular positions and orientations
USE mc, only :            ! in the values array
USE md, only :            values(1)=...
USE minimize, only :      values(2)=...
USE molecules, only :     ...
USE monolayer, only :     ! Calculate energy for this geometry
USE optimize, only :      call energy(values,es,er,eind,edisp,etotal)
USE optimize_layer, only : ! Do with it what you will
USE plot, only :          call private(etotal)
USE switches, only :      end do
USE tabulate, only :
USE types, only :         END PROGRAM my_prog
USE variables, only :
USE sites, only :
USE sizes, only :
USE sdata, only :

```

Figure 11: Organization of a program to use Orient as a subroutine.

References

- Allen, M. P. and Tildesley, D. J. (1987) *Computer Simulation of Liquids*, Clarendon Press, Oxford.
- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D. (1999) *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Andreev, E. A. (1973) 'On asymptotic calculation of the exchange interaction,' *Theor. Chim. Acta* **28**, 235–239.
- Bondi, A. (1964) 'Van der Waals volumes and radii,' *J. Phys. Chem.* **68**, 441–451.
- Buckingham, A. D. and Fowler, P. W. (1985) 'A model for the geometries of Van der Waals complexes,' *Can. J. Chem.* **63**, 2018–2025.
- Buckingham, A. D., Fowler, P. W. and Stone, A. J. (1986) 'Electrostatic predictions of shapes and properties of Van der Waals molecules,' *Internat. Rev. Phys. Chem.* **5**, 107–114.
- Cerjan, C. J. and Miller, W. H. (1981) 'On finding transition states,' *J. Chem. Phys.* **75**, 2800–2806.
- Dullweber, A., Leimkuhler, B. and McLachlan, R. (1997) 'Symplectic splitting methods for rigid body molecular dynamics,' *J. Chem. Phys.* **107**, 5840–5851.
- Engkvist, O., Price, S. L. and Stone, A. J. (2000) 'Developments in computational studies of crystallization and morphology applied to urea,' *Phys. Chem. Chem. Phys.* **2**, 3017–3027.
- Engkvist, O. and Stone, A. J. (1999a) 'Adsorption of water on the MgO(001) surface,' *Surface Sci.* **437**, 239–248.
- Engkvist, O. and Stone, A. J. (1999b) 'Adsorption of water on the NaCl(100) surface. I. Intermolecular potentials and low temperature structures,' *J. Chem. Phys.* **110**, 12089–12096.
- Evans, D. J. (1977) 'On the representation of orientation space,' *Molec. Phys.* **34**, 317–325.

- Evans, D. J. and Murad, S. (1977) 'Singularity-free algorithm for molecular dynamics simulation of rigid polyatomics,' *Molec. Phys.* **34**, 327–331.
- Ferrenberg, A. M. and Swendsen, R. H. (1988) *Phys. Rev. Lett.* **61**, 2635.
- Ferrenberg, A. M. and Swendsen, R. H. (1989) *Phys. Rev. Lett.* **63**, 1195.
- Fowler, P. W. and Stone, A. J. (1987) 'Induced dipole moments of Van der Waals complexes,' *J. Phys. Chem.* **91**, 509–511.
- Frantz, D. D., Freeman, D. L. and Doll, J. D. (1990) *J. Chem. Phys.* **93**, 2769.
- Fraschini, E. and Stone, A. J. (1998) 'An H···H model potential for the exchange–repulsion energy of the methane dimer,' *J. Comput. Chem.* **19**, 847–857.
- Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Zakrzewski, V. G., Montgomery, Jr, J. A., Stratmann, R. E., Burant, J. C., Dapprich, S., Millam, J. M., Daniels, A. D., Kudin, K. N., Strain, M. C., Farkas, O., Tomasi, J., Barone, V., Cossi, M., Cammi, R., Mennucci, B., Pomelli, C., Adamo, C., Clifford, S., Ochterski, J., Petersson, G. A., Ayala, P. Y., Cui, Q., Morokuma, K., Malick, D. K., Rabuck, A. D., Raghavachari, K., Foresman, J. B., Cioslowski, J., Ortiz, J. V., Stefanov, B. B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Gomperts, R., Martin, R. L., Fox, D. J., Keith, T., Al-Laham, M. A., Peng, C. Y., Nanayakkara, A., Gonzalez, C., Challacombe, M., Gill, P. M. W., Johnson, B. G., Chen, W., Wong, M. W., Andres, J. L., Head-Gordon, M., Replogle, E. S. and Pople, J. A. (1998) 'Gaussian98,' Tech. rep., Gaussian, Inc., Pittsburgh PA, U.S.A.
- Gray, C. G. and Gubbins, K. E. (1984) *Theory of Molecular Fluids*, vol. 1: Fundamentals, Oxford University Press, Oxford.
- Hayes, I. C. and Stone, A. J. (1984) 'An intermolecular perturbation theory for the region of moderate overlap,' *Molec. Phys.* **53**, 83–105.
- Hodges, M. P. (1998) 'Xmakemol: A program for viewing and manipulating atomic and molecular systems,' .
- Hodges, M. P. and Stone, A. J. (2000) 'A new representation of the dispersion interaction,' *Molec. Phys.* **98**, 275–286.
- Jeziorski, B., Moszynski, R. and Szalewicz, K. (1994) 'Perturbation theory approach to intermolecular potential energy surfaces of Van der Waals complexes,' *Chem. Rev.* **94**, 1887–1930.
- Joslin, C. and Goldman, S. (1993) '2nd dielectric and refractivity virial-coefficients of the polarizable dipolar hard-sphere fluid,' *Molec. Phys.* **79**, 1351–1364.
- Kumar, A. and Meath, W. J. (1994) 'Reliable isotropic and anisotropic dipole properties, and dipolar dispersion energy coefficients, for CO evaluated using constrained dipole oscillator strength techniques,' *Chem. Phys.* **189**, 467–477.
- Le Sueur, C. R. and Stone, A. J. (1993) 'Practical schemes for distributed polarizabilities,' *Molec. Phys.* **78**, 1267–1291.
- Le Sueur, C. R. and Stone, A. J. (1994) 'Localization methods for distributed polarizabilities,' *Molec. Phys.* **83**, 293–308.
- Le Sueur, C. R., Stone, A. J. and Fowler, P. W. (1991) 'Induced dipole moments in acetylene complexes,' *J. Phys. Chem.* **95**, 3519–3522.
- McQuarrie, D. A. (1976) *Statistical Mechanics*, Harper & Row, New York.

- Meredith, A. W. and Stone, A. J. (1996) 'A perturbation theory study of adlayer CO on NaCl (100),' *J. Chem. Phys.* **104**, 3058–3070.
- Meredith, A. W. and Stone, A. J. (1998) 'An *ab initio* and diffusion Monte Carlo study of the potential energy surface of the CO dimer,' *J. Phys. Chem. A* **102**, 434–445.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) 'Equation of state calculations by fast computing machines,' *J. Chem. Phys.* **21**, 1087–1092.
- Mirsky, K. (1978) 'The determination of the intermolecular interaction energy by empirical methods,' in *Computing in Crystallography*, eds. R. Schenk, R. Olthof-Hazenkamp, H. van Koningsveld and G. C. Bassi, 169–182, Delft University Press.
- Misquitta, A. J. and Stone, A. J. (2011) in preparation. See also www-stone.ch.cam.ac.uk/programs.html.
- Popelier, P. L. A. and Stone, A. J. (1994) 'Formulae for the first and second derivatives of anisotropic potentials with respect to geometrical parameters,' *Molec. Phys.* **82**, 411–425, Erratum, *Molec. Phys.* (1995) **84**, 811.
- Popelier, P. L. A., Stone, A. J. and Wales, D. J. (1994) 'Topography of potential energy surfaces for Van der Waals complexes,' *Faraday Disc.* **97**, 243–264.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992) *Numerical Recipes in Fortran*, Cambridge University Press, Cambridge, 2nd edn.
- Price, S. L. and Stone, A. J. (1987) 'The electrostatic interactions in Van der Waals complexes involving aromatic molecules,' *J. Chem. Phys.* **86**, 2859–2868.
- Rogowitz, B. E. and Treinish, L. A. (1996) 'Why should engineers and scientists be worried about color?' Tech. rep., Yorktown Heights, NY, <http://www.research.ibm.com/people/l/lloyd/color/color.HTM>.
- Rose, J. P. and Berry, R. S. (1992) *J. Chem. Phys.* **96**, 517.
- Smirnov, B. M. and Chibisov, M. I. (1965) 'Electron exchange and changes in the hyperfine state on collision of atoms of alkali metals,' *Sov. Phys.-JETP* **21**, 624.
- Stoer, J. and Bulirsch, R. (1980) *Introduction to Numerical Analysis*, Springer Verlag, New York.
- Stone, A. J. (1981) 'Distributed multipole analysis; or how to describe a molecular charge distribution,' *Chem. Phys. Lett.* **83**, 233–239.
- Stone, A. J. (1985) 'Distributed polarizabilities,' *Molec. Phys.* **56**, 1065–1082.
- Stone, A. J. (1996) *The Theory of Intermolecular Forces*, International Series of Monographs in Chemistry, Clarendon Press, Oxford.
- Stone, A. J. (1998) 'GDMA: distributed multipoles from Gaussian98 wavefunctions,' Tech. rep., University of Cambridge.
- Stone, A. J. (2005) 'Distributed multipole analysis: Stability for large basis sets,' *J. Chem. Theory Comput.* **1**, 1128–1132.
- Stone, A. J. and Alderton, M. (1985) 'Distributed multipole analysis—methods and applications,' *Molec. Phys.* **56**, 1047–1064.
- Stone, A. J. and Price, S. L. (1988) 'Some new ideas in the theory of intermolecular forces: anisotropic atom-atom potentials,' *J. Phys. Chem.* **92**, 3325–3335.

- Stone, A. J. and Tong, C.-S. (1994) 'Anisotropy of atom–atom repulsions,' *J. Comput. Chem.* **15**, 1377–1392.
- Szczeńśniak, M. M., Brenstein, R. J., Cybulski, S. M. and Scheiner, S. (1990) 'Potential energy surface for dispersion interaction in (H₂O)₂ and (HF)₂,' *J. Phys. Chem.* **94**, 1781–1788.
- Tang, K. T. and Toennies, J. P. (1992) 'The damping function of the Van der Waals attraction in the potential between rare gas atoms and metal surfaces,' *Surface Science* **279**, L203–L206.
- Tsai, C. J. and Jordan, K. D. (1993) 'Use of the histogram and jump-walking methods for overcoming slow barrier crossing behavior in monte carlo simulations: Applications to the phase transitions in the (Ar)₁₃ and (H₂O)₈ clusters,' *J. Chem. Phys.* **99**, 6957.
- Wales, D. J. (1991) 'Theoretical study of some small Van der Waals complexes containing inert gas atoms,' *Molec. Phys.* **74**, 1–25.
- Wales, D. J. (2003) *Energy Landscapes*, Cambridge University Press, Cambridge.
- Wales, D. J., Popelier, P. L. A. and Stone, A. J. (1995) 'Potential energy surfaces of Van der Waals complexes of water and hydrogen halides modelled using distributed multipoles,' *J. Chem. Phys.* **102**, 5551–5565.
- Werner, H.-J. and Knowles, P. J. (1998) 'MOLPRO98,' Tech. rep., MOLPRO is a package of *ab initio* programs written by H.-J. Werner and P. J. Knowles, with contributions from J. Almlöf, R. D. Amos, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, S. T. Elbert, C. Hampel, R. Lindh, A. W. Lloyd, W. Meyer, A. Nicklass, K. Peterson, R. Pitzer, A. J. Stone, P. R. Taylor, M. E. Mura, P. Pulay, M. Schütz, H. Stoll, and T. Thorsteinsson.
- Wheatley, R. J. and Price, S. L. (1990) 'An overlap model for estimating the anisotropy of repulsion,' *Molec. Phys.* **69**, 507–533.
- Williams, D. E. (1965) 'Repulsion center of a bonded hydrogen atom,' *J. Chem. Phys.* **43**, 4424–4426.
- Williams, D. E. (1967) 'Non-bonded potential parameters derived from crystalline hydrocarbons,' *J. Chem. Phys.* **47**, 4680–4684.

Change Log

Version 4.6

1. New commands provided for graphical displays of electrostatic potential, electric field, and other properties.
2. Extensive minor changes to improve the logical organisation of the program. Some code rewritten, some bugs fixed.

Version 4.5

1. Several periodic layers can be defined, and optimized independently.

Version 4.4

1. The code for lattice summations has been rewritten, and analytical derivatives can now be calculated, making structure optimizations more efficient for clusters and adlayers on surfaces.
2. The options for optimizations, including quenching optimizations during Monte Carlo or molecular dynamics simulations, can be set in a separate OPTIONS data block and are not reset at the beginning of an optimization or simulation.

Version 4.3

1. The direct-space part of lattice sums is carried out over cells within a specified distance of the global origin, rather than over a range of cell indices. This change makes little difference for cubic lattices, but gives better convergence in lower symmetries. The data syntax has changed as a result.
2. The syntax for specifying sites in a lattice has changed.
3. Various bug fixes.

Version 4.2

1. The dispersion and induction damping factors can now be specified independently.
2. A better optimizer has been installed to handle cases where the energy derivatives are not available.
3. Various bugs have been fixed.

Version 4.1

1. The program has been converted to Fortran 90. This allows run-time allocation of arrays, so that large arrays can be allocated only when they are needed.
2. Parameters affecting array sizes (number of sites, number of molecules, etc.) can be changed at run-time. In particular, the number of polarizable sites is zero by default.
3. Improved virial coefficient routines. The dielectric virial coefficient can also be calculated.
4. The potential fitting procedures have been extended to allow fitting of dispersion energy parameters as well as repulsion.
5. The induction energy calculation should be more efficient.
6. Systems involving molecular clusters, periodic adlayers and nanocrystals adsorbed on crystal surfaces can be studied.

Index of commands and declarations

		Page
ADD	Add specified increments to one or more position or orientation variables.	5
ADJUST	Fit repulsion or dispersion parameters to <i>ab initio</i> data points.	51
ADLAYER	Define a periodic two-dimensional layer of molecules.	14
ANALYSE	Analyse many-body contributions	26
ATOM	Specify the position and characteristics of an atom.	11
CONSTANTS	Define a list of named constants.	5
DEBUG	Set switches to control diagnostic output from certain routines.	7
DEFAULT	Specify the default rank of multipole moments.	6
DISPLAY	Produce a graphical display of electrostatic potential, electric field, or probe atom energy.	53
EDIT	Modify a molecule description.	20
ENERGY	Calculate the energy of the current configuration.	25
ESLIMIT	Ignore high-rank electrostatic interactions.	6
FINISH	Stop.	8
FIT	List <i>ab initio</i> data points to be fitted using ADJUST.	50
IGNORE	Ignore a section of the data file.	7
LATTICE	Define a lattice.	11
LAYER	Define a periodic two-dimensional layer of molecules.	14
MC	Carry out a Monte Carlo simulation	32
MD	Carry out a molecular dynamics simulation.	42
MINIMIZE	Find a minimum on the potential-energy surface of the current system.	28
MOLECULE	Specify the position and structure of a molecule.	8
NANOCRYSTAL	Define a nanocrystal.	15
NOTE	Comment; ignored.	7
OPTIMIZE	Find a stationary point on the potential-energy surface of the current system.	28
PAIRS	Specify the parameters for the interaction between pairs of sites.	17
PARAMETERS	Modify general program parameters.	50
POLARIZABILITIES	Specify or manipulate polarizabilities.	22
PLOT	Plot the current system configuration.	8
PRINTLEVEL	Specify the quantity of printed output.	7
RESTORE	Restore a molecule previously suppressed, or a geometry previously dumped.	7
SADDLE	Find a saddle-point on the potential-energy surface of the current system.	28
SET	Set one or more of the position and orientation variables.	5
SHOW	Display various things.	25
SLICE	Sample a potential energy surface in one or two dimensions.	27
STOP	Stop.	8
SUPPRESS	Mark a molecule as inactive.	6
SWITCH	Set various switches to control the calculation.	7
TABULATE	Tabulate the electrostatic potential, and the field and field gradient if required, at a series of specified points.	27
TIME	Print the cpu time used.	7
TITLE	Specify a title, or print a previously specified title.	6
TRIPLE-DIPOLE	Estimate the triple-dipole dispersion energy.	26

		Page
TYPES	Specify the properties of site types.	16
UNITS	Specify units of distance and energy.	6
VARIABLES	Define a list of named variables.	5
VIRIAL	Calculate the pressure or dielectric second virial coefficient for a pair of molecules.	48