

# THE DL\_MULTI USER MANUAL

M. Leslie,  
CCLRC, Daresbury Laboratory,  
Daresbury,  
Warrington WA4 4AD,  
England

Version 2.00 Jan 2003

## ABOUT DL\_MULTI

DL\_MULTI is an extension to the standard DL\_POLY package developed at Daresbury Laboratory by M. Leslie for the EPSRC's Collaborative Computational Project for the Computer Simulation of Condensed Phases (CCP5). The package is the property of The Council for the Central Laboratory of the Research Councils (CCLRC).

DL\_MULTI is issued free **under licence** to academic institutions pursuing scientific research of a non-commercial nature. Commercial organisations may be permitted a licence to use the package after negotiation with the owners. Daresbury Laboratory is the sole centre for distribution of the package. Under no account is it to be redistributed to third parties without consent of the owners.

The purpose of the DL\_MULTI package is to provide software for academic research that is inexpensive, accessible and free of commercial considerations. Users have direct access to source code for modification and inspection. In the spirit of the enterprise, contributions in the form of working code are welcome, provided the code is compatible with DL\_MULTI in regard to its interfaces and programming style and it is adequately documented.

## **DISCLAIMER**

Neither the CCLRC, EPSRC, CCP5 nor any of the authors of the DL\_MULTI package or its derivatives guarantee that the package is free from error. Neither do they accept responsibility for any loss or damage that results from its use. Use of the DL\_MULTI package without charge is confined to academic research only. Commercial use is only permissible following negotiation with Daresbury Laboratory. Users are not entitled to redistribute the program to third parties.

## ACKNOWLEDGEMENTS

DL\_MULTI was developed under the auspices of the Council for the Central Laboratory of the Research Councils and the Engineering and Physical Sciences Research Council. The package is the property of the Council for the Central Laboratory of the Research Councils of the United Kingdom.

Advice, assistance and encouragement in the development of DL\_MULTI has been given by many people. We gratefully acknowledge the following:

W. Smith, D. Willock, S.L. Price, A. Stone

## Manual Notation

In the DL\_MULTI manual the same notation is used as in the DL\_POLY Manual. Specific fonts are used to convey specific meanings:

1. *directories* - indicates unix file directories
2. ROUTINES - indicates subroutines, functions and programs.
3. *macros* - indicates a macro (file of unix commands)
4. **directive** - indicates directives or keywords
5. **variables** - indicates named variables and parameters
6. FILE - indicates filenames.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	The DL_MULTI Package . . . . .	7
1.2	Extensions in DL_MULTI compared with DL_POLY . . . . .	7
1.2.1	Molecular Systems . . . . .	7
1.2.2	Force Field . . . . .	7
1.2.3	Boundary Conditions . . . . .	8
1.2.4	Target Computers . . . . .	8
1.3	Obtaining the Source Code . . . . .	8
<b>2</b>	<b>DL_MULTI Data Files</b>	<b>9</b>
2.1	The INPUT files . . . . .	10
2.1.1	The CONTROL File . . . . .	10
2.1.2	The CONFIG File . . . . .	10
2.1.3	The FIELD File . . . . .	11
2.2	The OUTPUT Files . . . . .	19
2.2.1	The OUTPUT File . . . . .	19

# Chapter 1

## Introduction

### Scope of Chapter

This chapter describes the concept, design and directory structure of DL\_POLY and how to obtain a copy of the source code.

## 1.1 The DL\_MULTI Package

DL\_MULTI is an extension of DL\_POLY . It is designed to carry out simulations on rigid molecules whose electrostatics are described using the distributed multipole analysis of Stone et al. [1]

In the next section we outline the differences of DL\_MULTI compared with DL\_POLY as briefly as possible.

## 1.2 Extensions in DL\_MULTI compared with DL\_POLY

The current version of DL\_MULTI is based on DL\_POLY 2.13. We describe below the extensions in DL\_MULTI compared with DL\_POLY and also the limitations.

### 1.2.1 Molecular Systems

DL\_MULTI will simulate the following molecular species:

1. Simple rigid molecules e.g.  $\text{CCl}_4$ ,  $\text{SF}_6$ , Benzene, etc.
2. Rigid molecular ions with point charges e.g.  $\text{KNO}_3$ ,  $(\text{NH}_4)_2\text{SO}_4$ , etc.
3. Systems which consist of mixtures of different types of molecule.

However, the following systems cannot be treated.

1. Molecules with flexible bonds

### 1.2.2 Force Field

The DL\_POLY force field is used and includes the following features:

1. All common forms of non-bonded atom-atom potential;
2. Atom-atom (site-site) Coulombic potentials;
3. Atom-atom (site-site) distributed multipole potentials;

The following DL\_POLY force field features should not be used in DL\_MULTI :

1. Valence angle potentials;
2. Dihedral angle potentials;
3. Inversion potentials;
4. Improper dihedral angle potentials;
5. 3-body valence angle and hydrogen bond potentials;
6. 4-body inversion potentials;
7. Sutton-Chen density dependent potentials (for metals)



### 1.2.3 Boundary Conditions

DL\_MULTI has only been tested in parallelepiped periodic boundary conditions. The author of DL\_MULTI cannot guarantee that the other boundary conditions in DL\_POLY will work.

#### 1.2.3.1 Parallel Algorithms

DL\_MULTI exclusively employs the **Replicated Data** parallelisation strategy [2, 3]

### 1.2.4 Target Computers

DL\_MULTI is targeted towards distributed memory parallel computers. However, versions of the program for serial computers are easily produced. To facilitate this all machine specific calls are located in dedicated FORTRAN routines, to permit substitution by appropriate alternatives, or even deletion. Note that some of the communication routines in DL\_MULTI, which have routines in DL\_POLY with the same name, have been modified and have only been tested using MPI. If you are planning to use a different parallelisation method you should test these routines yourself.

DL\_MULTI has been tested on on the the following computers:

1. IBM SP/2
2. SUN SPARC and ULTRA SPARC.
3. Beowulf systems

Porting of DL\_MULTI to these and other machines requires MPI message passing tools.

## 1.3 Obtaining the Source Code

Holders of a licence for DL\_POLY can obtain a copy of DL\_MULTI as a separate stand alone program.

## Chapter 2

# DL\_MULTI Data Files

### Scope of Chapter

This chapter describes all the input and output files for DL\_POLY , examples of which are to be found in the *data* sub-directory.

## 2.1 The INPUT files

DL\_MULTI requires five input files named CONTROL, CONFIG, FIELD, TABLE and REVOLD. The first three files are mandatory, while TABLE is used only to input certain kinds of pair potential, and is not always required. REVOLD is required only if the job represents a continuation of a previous job. TABLE and REVOLD are the same in DL\_MULTI as in DL\_POLY and are not described further here. In the following sections we describe the changes to the standard DL\_POLY files.

### 2.1.1 The CONTROL File

The CONTROL file is read by the subroutine SIMDEF and defines the control variables for running a DL\_POLY or DL\_MULTI job.

The only difference in the CONTROL file between DL\_MULTI and DL\_POLY is in the definition of the Ewald sum precision. DL\_POLY uses the record

```
ewald precision  f
whereas DL_MULTI uses the following records
mulp precision
fd0
fd1
fd2
fd3
fd4
fr0
fr1
fr2
fr3
fr4
```

*fd0* is the Ewald precision in direct space for pole order 0, *fd1* for pole order 1 and so on. *fr0*, *fr1* are the corresponding terms for reciprocal space.

It is recommended that the user sets a high precision (smaller value of the *f* parameter) for pole orders greater than 0 in direct space. If this is not done there will be an energy drift due to energy changes when the higher order poles cross the cutoff boundary. Values of 0.001 can be used for pole order 0 and for all the reciprocal space terms, although smaller values will give better energy stability. Values not less than 0.000001 should be used for the higher order terms in direct space.

### 2.1.2 The CONFIG File

The format of the CONFIG file for DL\_MULTI is exactly the same as for DL\_POLY . Remember that all molecules of a particular molecule type must come together in the file and all atoms of a molecule must come together in the same order for all molecules. If the molecule has enantiomers, these are treated as two different molecules and all configurations of the first enantiomer must come before any of the second.

### 2.1.3 The FIELD File

The FIELD file contains the force field information defining the nature of the molecular forces. It is read by the subroutine SYSDEF. Excerpts from a force field file are shown below. The example is 5-azauracil.

```
DL_POLY TEST CASE azauracil
units ev
molecular types 2
azauracil+
multipole
nummols 96
atoms 11
NI          14.007          0.00    1
      4
      -0.495104
      -0.001568  -0.158410   0.013091
      -0.459413   0.018431  -0.035157   0.425186  -0.002068
      0.032228   0.664233  -0.130477  -0.013206   0.245208   2.059576   0.109874
      -0.287987  -0.054286  -0.290048  -0.887905  -0.346797   0.070680  -0.105214
      0.418582  -2.343179
NI          14.007          0.00    1
      4
      -0.608897
      0.018043   0.051003  -0.098955
      -0.204276  -0.038276   0.039801   0.041331   0.096621
      -0.035699  -0.202148   0.109761  -0.163215  -0.050993   2.800004  -0.016288
      -1.130707  -0.159565   0.055705   0.353922  -0.056397   0.058125  -0.131937
      -0.876265   2.423670
..... 9 more atoms
rigid units 1
  11   1   2   3   4   5   6   7   8   9  10  11
mulaxes molaxes
  1   3   6   2   3   6   2   3   1
finish
azauracil-
multipole
nummols 96
atoms 11
NI          14.007          0.00    1
      4
      -0.495104
      -0.001568  -0.158410   0.013091
      -0.459413   0.018431  -0.035157   0.425186  -0.002068
      0.032228   0.664233  -0.130477  -0.013206   0.245208   2.059576   0.109874
```

```

-0.287987 -0.054286 -0.290048 -0.887905 -0.346797 0.070680 -0.105214
0.418582 -2.343179
..... 10 more atoms
rigid units 1
  11  1  2  3  4  5  6  7  8  9  10  11
mulaxes molaxes
  1  3  6  2  3  6  2  3 -1
finish
vdw 15
CA    CA    buck  3832.1    0.277778    25.287
CA    HY    buck   689.5    0.272480     5.979
CA    HP    buck   446.9    0.242131     2.374
CA    NI    buck  3179.5    0.271003    19.007
CA    OX    buck  3022.8    0.264550    17.160
HY    HY    buck   124.1    0.267380     1.414
HY    HP    buck    80.4    0.238095     0.561
HY    NI    buck   572.1    0.265957     4.494
HY    OX    buck   543.9    0.259740     4.057
HP    HP    buck    52.1    0.214592     0.223
HP    NI    buck   370.8    0.236967     1.784
OX    HP    buck   352.6    0.232019     1.611
NI    NI    buck  2638.0    0.264550    14.286
NI    OX    buck  2508.0    0.258398    12.898
OX    OX    buck  2384.5    0.252525    11.645
close

```

### 2.1.3.1 Format

The FIELD file is fixed-formatted. Integers are formatted as “i5”, reals are normally “f12.0”, although the multipole moments are “f11.0”, and characters are “a4”, “a8”, “a40” or “a80”, depending on context.

### 2.1.3.2 Definitions of Variables

The file divides into three sections: general information, molecular descriptions, and non-bonded interaction descriptions, appearing in that order in the file.

#### 2.1.3.2.1 General information

There are no changes from DL\_POLY .

record 4 (optional)

**mpdist**            a40        Distance units used to define the multipoles

The distance units on the **mpdist** directive are described by additional keywords:

**angstrom** , for Å

**au** , for atomic units

The default if the mpdist directive is omitted is **au**.

### 2.1.3.2.2 Molecular details

As is the case for standard DL\_POLY , the FIELD file must give the molecular details for the molecule types in the same order as they appear in the CONFIG file.

The entry of the molecular details begins with the mandatory directive:

**molecules** *n* or

**molecular types** *n*

where *n* is an integer specifying the number of different *types* of molecule appearing in the FIELD file. Note that enantiomers are counted as two different types of molecule. Once this directive has been encountered, DL\_MULTI enters the *molecular description* environment in which only molecular description keywords and data are valid.

Immediately following the **molecular types** directive, are the records defining individual molecules. The first directive is the same as in DL\_POLY :

1. *name-of-molecule*

which can be any character string up to 80 characters in length. (Note: this is not a directive, just a simple character string.)

The next directive tells DL\_MULTI that there are multipoles in the FIELD file.

2. **multipole**

The next directive is the same as DL\_POLY

3. **nummols** *n*

where *n* is the number of times a molecule of this type appears in the simulated system. The molecular data then follow in subsequent records:

4. **atoms** *n*

where *n* indicates the number of atoms in this type of molecule. A number of records follow, each giving details of the atoms in the molecule i.e. site names, masses, multipole order and multipole components. Unlike the standard DL\_POLY , there are several records for each atom. The first record carries the entries:

<b>sitnam</b>	a8	atomic site name
<b>weight</b>	real	atomic site mass
<b>chge</b>	real	atomic site charge
<b>nrept</b>	integer	repeat counter

<code>ifrz</code>	integer	'frozen' atom (if <code>ifrz</code> > 0)
<code>igrp</code>	integer	neutral/charge group number

This is the same as the standard DL\_POLY . Note however that the atomic site charge read here is ignored as it will be read below in the multipole input. Also the repeat counter will almost always be 1 as atoms in the molecule with the same atomic number will have different multipoles and will be treated by DL\_MULTI as different atomic types. `ifrz` and `igrp` are not relevant for DL\_MULTI use and should be omitted.

The next record carries the entry (without a keyword)

<code>jpoleo</code>	i8	multipole pole order for this atom
---------------------	----	------------------------------------

$0 \leq \text{jpoleo} < 4$

`jpoleo` is the pole order for the multipoles for this atom.

There are then a number of records giving the magnitude of the multipoles depending on `jpoleo`. The real numbers giving the magnitude of the multipoles are "f11.0" fixed format.

In all cases there is a record format "f11.0" giving the multipole charge term (order 0).

If `jpoleo`  $\geq 1$  there is a record format "3f11.0" giving the dipole components.

If `jpoleo`  $\geq 2$  there is a record format "5f11.0" giving the quadrupole components.

If `jpoleo`  $\geq 3$  there is a record format "7f11.0" giving the octopole components.

If `jpoleo`  $\geq 4$  there are 2 records format "7f11.0/2f11.0" giving the hexadecapole components.

The order of the multipole components follow the usual order for spherical harmonics. [4] Table 1. These are given below.

$$\begin{aligned}
 Q_{00} &= \sum_i e_i \\
 Q_{10} &= \sum_i e_i z_i \\
 Q_{11c} &= \sum_i e_i x_i \\
 Q_{11s} &= \sum_i e_i y_i \\
 Q_{20} &= \sum_i e_i \frac{1}{2} (3z_i^2 - r_i^2)
 \end{aligned}$$

$$\begin{aligned}
Q_{21c} &= \sum_i e_i \sqrt{3} x_i z_i \\
Q_{21s} &= \sum_i e_i \sqrt{3} y_i z_i \\
Q_{22c} &= \sum_i e_i \frac{1}{2} \sqrt{3} (x_i^2 - y_i^2) \\
Q_{22s} &= \sum_i e_i \sqrt{3} x_i y_i \\
Q_{30} &= \sum_i e_i \frac{1}{2} (5z_i^3 - 3z_i r_i^2) \\
Q_{31c} &= \sum_i e_i \sqrt{\frac{3}{8}} x_i (5z_i^2 - r_i^2) \\
Q_{31s} &= \sum_i e_i \sqrt{\frac{3}{8}} y_i (5z_i^2 - r_i^2) \\
Q_{32c} &= \sum_i e_i \frac{1}{2} \sqrt{15} z_i (x_i^2 - y_i^2) \\
Q_{32s} &= \sum_i e_i \sqrt{15} x_i y_i z_i \\
Q_{33c} &= \sum_i e_i \sqrt{\frac{5}{8}} (x_i^3 - 3x_i y_i^2) \\
Q_{33s} &= \sum_i e_i \sqrt{\frac{5}{8}} (3x_i^2 y_i - y_i^3) \\
Q_{40} &= \sum_i e_i \frac{1}{8} (35z_i^4 - 30z_i^2 r_i^2 + 3r_i^4) \\
Q_{41c} &= \sum_i e_i \sqrt{\frac{5}{8}} (7x_i z_i^3 - 3x_i z_i r_i^2) \\
Q_{41s} &= \sum_i e_i \sqrt{\frac{5}{8}} (7y_i z_i^3 - 3y_i z_i r_i^2) \\
Q_{42c} &= \sum_i e_i \frac{1}{4} \sqrt{5} (x_i^2 - y_i^2) (7z_i^2 - r_i^2) \\
Q_{42s} &= \sum_i e_i \frac{1}{2} \sqrt{5} x_i y_i (7z_i^2 - r_i^2) \\
Q_{43c} &= \sum_i e_i \sqrt{\frac{35}{8}} z_i (x_i^3 - 3x_i y_i^2) \\
Q_{43s} &= \sum_i e_i \sqrt{\frac{35}{8}} z_i (3x_i^2 y_i - y_i^3)
\end{aligned}$$



$$Q_{44c} = \sum_i e_i \frac{1}{8} \sqrt{35} (x_i^4 - 6x_i^2 y_i^2 + y_i^4)$$

$$Q_{44s} = \sum_i e_i \frac{1}{2} \sqrt{35} (x_i^3 y_i - x_i y_i^3)$$

DLMULTI does not use the following directives from DLPOLY . **bonds, constraints, pmf, angles, dihedrals, inversions teth.** For DLMULTI the records **rigid units** and **mulaxes** are described below. **Note** that the atomic site indices referred to below are indices arising from numbering each atom in the molecule from 1 to the number specified in the **atoms** directive for this molecule. This same numbering scheme should be used for all descriptions of this molecule. DLMULTI will itself construct the global indices for all atoms in the systems, as for DLPOLY .

#### 5. rigid *n* or rigid units *n*

where *n* is the number of rigid units in the molecule. For DLMULTI *n* must be 1. It is followed by one record, specifying the sites in a rigid unit:

m	integer	number of sites in rigid unit
site 1	integer	first site atomic index
site 2	integer	second site atomic index
site 3	integer	third site atomic index
..	..	<i>etc.</i>
site m	integer	m'th site atomic index

Up to 15 sites can be specified on the first record. Additional records are used if necessary. Up to 16 sites are specified per record thereafter. The format is 16i5.

#### 6. mulaxes

**mulaxes**      a40      Define the axis system used to define the multipoles

There is a second keyword on the **mulaxes** directive.

**rotint** , the multipole components are described with respect to the principle moments of inertia axis system. As for DLPOLY , the principal axes are assigned to molecular types with the components of the rotational inertia tensor **I** obeying:  $I_{xx} \geq I_{yy} \geq I_{zz}$  (to insure all processors converge on the same axes system). There is an addition in DLMULTI from the standard DLPOLY which ensures that the principal axes are always in the same direction with respect to multiplication by -1. This ensures that the multipole components can be defined with respect to this axis system unambiguously.

**molaxes** , the multipole components are described with respect to an axis system defined by atoms in the molecule. This will be the normal input method for DL\_MULTI , although internally the program will carry out a conversion using Wigner matrices to the **rotint** convention before carrying out any calculation. The **mulaxes** record must be followed by one record with 9 integer values format 9i5.

```
direction 1
atom 1a
atom 1b
direction 2
atom 2a
atom 2b
atom 2c
direction 3
enantiomer flag
```

direction 1, 2 and 3 refer to the x, y and z directions in the local axis system. (Use the integer 1 for x, 2 for y and 3 for z, each of the values must be used once). Usually direction 1 will be 1, direction 2 will be 2 and direction 3 will be 3, and this must be the order if the enantiomer flag is  $-1$ . The axis in direction 1 (normally x axis) is defined along the bond from atom 1a to atom 1b. The axis in direction 2 (normally y axis) is defined in the plane containing atoms 2a, 2b and 2c, and normal to the axis in direction 1. The axis in direction 3 ( normally z axis) makes a right handed set with 1 and 2. The enantiomer flag can have values 1 or -1.

In many crystalline systems the crystal structure contains a mixture of two enantiomers (at least experimentally). Often this will be the result of a slight departure from planarity in the experimental results which may or may not be physically real. To permit the user the option of inputting a pair of enantiomers, they are treated by DL\_MULTI as two separate molecules. However, the setting up of the molecular axes using **molaxes** will set up two right handed axes systems which are not mirror images of each other. To allow for this the signs of the appropriate multipole components need to be changed. There are two ways the user can do this. First, the user can change the sign of all the odd-z components in the multipole description of the second molecule. Secondly, the user can use the enantiomer flag -1 and leave the components as they are. DL\_MULTI will then change the sign of the odd-z components. Note that if a non-standard order of the axes in **molaxes** is used, then the first method must be used. The user will need to change the sign of the odd-x or odd-y components for the second molecule (according to which axis is defined third in the description).

## 7. finish

This directive is entered to signal to DL\_POLY that the entry of the details of a

molecule has been completed.

The entries for a second molecule may now be entered, beginning with the *name-of-molecule* record and ending with the **finish** directive.

The cycle is repeated until all the types of molecules indicated by the **molecules** directive have been entered.

The user is recommended to look at the example FIELD files in the *data* directory to see how typical FIELD files are constructed.

### 2.1.3.3 Non-bonded Interactions

Non-bonded interactions are identified by atom types as opposed to specific atomic indices. The description in DL\_MULTI is the same as in DL\_POLY .

## 2.2 The OUTPUT Files

DLMULTI writes the same output files as DL\_POLY . The only differences are in the OUTPUT file.

### 2.2.1 The OUTPUT File

The job output consists of 7 sections: Header; Simulation control specifications; Force field specification; Summary of the initial configuration; Simulation progress; Summary of statistical data; Sample of the final configuration; and Radial distribution functions. These sections are written by different subroutines at various stages of a job. Creation of the OUTPUT file *always* results from running DLMULTI . It is meant to be a human readable file, destined for hardcopy output.

#### 2.2.1.1 Header

DLMULTI output is the same as DL\_POLY

#### 2.2.1.2 Simulation Control Specifications

DLMULTI writes out the Ewald convergence parameter that DL\_POLY would use and the reciprocal space cutoffs. These values are not used by DLMULTI . The correct values are printed when the force field and initial configuration have been read in.

#### 2.2.1.3 Force Field Specification

In addition to the normal DL\_POLY output, DLMULTI writes out the input distributed multipoles

#### 2.2.1.4 Summary of the Initial Configuration

In addition to the normal DL\_POLY output, the cutoffs for the Ewald sum for each pole order are written out. These are also recalculated every 100 time steps and written to the OUTPUT file, since for simulations in which the cell volume changes the cutoffs may need to be changed to obtain the same accuracy.

#### 2.2.1.5 Simulation Progress

This is the same as DL\_POLY , apart from the cutoff recalculation.

#### 2.2.1.6 Remaining subsections

These are all the same as for DL\_POLY .

# Bibliography

- [1] Price, S. L., Stone, A. J., and Alderton, M., 1984, *Mol. Phys.*, **52**, 987–1001.
- [2] Smith, W., and Forester, T. R., 1994, *Comput. Phys. Commun.*, **79**, 52.
- [3] Smith, W., and Forester, T. R., 1994, *Comput. Phys. Commun.*, **79**, 63.
- [4] Stone, A. In Maksić, Z. B., editor, *Theoretical Models of Chemical Bonding, Part 4*. Springer-Verlag, 1991.