

# A GENERAL-PURPOSE MD CODE

Keith Refson

Moldy is a general-purpose molecular dynamics simulation program which I wrote initially for my own research. It is sufficiently flexible that it ought to be useful for a wide range of simulation calculations of atomic, ionic and molecular systems and I am therefore offering it to the CCP5 community.

The design objectives were for a single program which is able to simulate a wide range of systems without arbitrary restrictions on the number or form of constituents. The system is specified at run time with a description file so there is no need to recompile when changing systems. Another design objective was that the program should handle much more of the bookkeeping than is traditionally done, especially with regard to keeping track of consistency of parameters, restart files, output trajectories and so forth. You don't have to worry about array sizes and limits because this is all handled automatically. It ought to be easier to concentrate on the science by making starting and keeping track of a new simulation a simpler process and helping to eliminate many of the frustrating and time-wasting mistakes that every simulator is so familiar with.

The program can handle any mixture of atoms or polyatomic molecules (linear or otherwise) of any size within the rigid-molecule approximation. There are no limits on the number of atoms in a molecule, the number of molecular species or number of molecules. The system can be in the liquid or solid state, with MD cells of arbitrary dimensions and angles and the simulation may be conducted at either constant volume or constant stress using the Parrinello-Rahman algorithm. Interactions are by pair-potentials (based at atomic sites in the case of molecules) with or without coulombic interactions. Most common forms of potential functions are supported (Lennard-Jones, Buckingham, Born-Meyer, MCY) and the program is designed to make it very easy to add others. Short-ranged forces are handled using the link-cell method and the long-ranged coulombic forces by the Ewald sum. Therefore the program ought to be suitable for simulating very large systems.

There are several features which are slightly novel. First, Moldy does not use the usual "minimum-image" convention, but instead includes interactions between a molecule and ALL of its periodic images that lie inside the cut-off radius. This is more strictly correct and just as easy to implement as minimum-image because of the link-cell algorithm. Second, Moldy incorporates a method of generating initial configurations for liquid systems called a "skew start". This can reliably generate a configuration which is partially ordered but avoids molecular overlap. Finally, there is a capability for defining a "framework" which is a rigid super-molecule permeating all of space. This may be used to model rigid surfaces or zeolite-like cages, for example.

One other aspect of moldy which might be unfamiliar is that it is written in C rather than FORTRAN. Fortran does not have the flexibility of dynamic memory allocation to allow the automatic sizing of the arrays which Moldy needs. This ought to present no problems as C compilers

are just as or more common than FORTRAN ones. The program is highly portable and has been optimised for both vector supercomputers (cray and convex), but also runs fast on modern unix workstations and even PCs. There is a parallel version for shared-memory parallel-processors including explicitly Cray and Convex machines. A port to distributed memory parallel architectures ought to be straightforward, though the current parallelization strategy will not scale well to very large numbers of processors.

The program incorporates radial distribution function calculations and running accumulation of many of the usual thermodynamic averages. Any more sophisticated analysis can be performed by storing configurational data throughout the run for later analysis. There are flexible facilities for doing this. In addition to the main program there are utilities for manipulating dump datasets and an interface to a molecular graphics module for AVS.

Given the above claims for generality and flexibility it is, only fair to mention the current limitations of the program. Only pair potentials are supported at the moment. New forms of potential function are easily added, but bond-bending or 3-body forces or shell models will take rather more work. The program treats molecules as rigid bodies using the quaternion algorithms, and no flexibility or other constraints are allowed. There is also no support at present for a thermostat - temperature is controlled by velocity scaling techniques. Since the source code is freely available I hope that others with a need for these facilities will be able to add modules and extend the capabilities.

Moldy differs in functionality and strategy from the CCP5 project, DL\_POLY. While DL\_POLY is implemented as a series of modules to be bolted together, Moldy is a single program which is configured at run-time by specification files. This does lead to less choice, for example in matters of boundary conditions and treatment of long-range forces, but makes starting a new simulation substantially easier and less prone to error. Moldy is aimed at systems of small molecules for which the rigid-molecule approximation is useful. It therefore supports massless sites and implements interaction cutoffs using a molecular rather than a site criterion. The "feature-lists" differ - Moldy implements the constant-stress ensemble for the study of solid-state phase transitions, whereas DL\_POLY offers thermostats - but we will no doubt see the holes filled as both programs develop. Moldy shares with DL\_POLY the principle of giving the user control over the source code, and is designed in a modular fashion using the principles of structured programming to encourage extensions to be added as needed. Finally, moldy is available now to anyone in CCP5 or otherwise who wishes to use it.

The source code may be obtained from the CCP5 program library in the usual way, and also directly by anonymous file transfer from Oxford. Connect to "earth.ox.ac.uk" using "ftp", with an account name of "anonymous" and your email address as password. The relevant files are all in the "/pub" directory and are

- moldy-2.6.tar.Z — The Unix distribution (also for MSDOS)
- moldy-2.6.com — The VMS distribution
- moldy-manual.ps.Z — The Manual in PostScript form. Note that the distribution files already contain the LaTeX source.

(The current release is 2.6; I intend to keep to this naming scheme and just keep the highest release on the ftp server.)

Please note that moldy is copyrighted and distributed under the GNU public license which is designed to encourage its distribution and modification. This is to ensure that the source code of moldy and any improvements made to it remain freely available. I would like to encourage anyone who improves the program to return the changes to me so they can be made incorporated into future releases for the benefit of all.

I am also keeping a list of email addresses of anyone who uses the program for notification of updates, bugs and so forth. Please notify me if you would like to be added to this list, preferably by email to [Keith.Refson@earth.ox.ac.uk](mailto:Keith.Refson@earth.ox.ac.uk).