

# DL\_POLY: A macromolecular simulation package

T. Forester and W. Smith,  
S.E.R.C. Daresbury Laboratory,  
Daresbury,  
Warrington WA4 4AD,  
England

The following is a summary of the DL\_POLY presentation given at the recent CCP5 AGM at Keele University, 17 September 1993 . . .

The DL\_POLY package is funded by the SERC through the SBCC (Science Board Computing Committee - which has recently changed it's name to the SMCC (Science Materials Computing Committee)). The motivation of the package is to provide a parallel macromolecular simulation package free to the academic community. The reasons for this are many. For example, while there are a number of commercially available (parallel) macromolecular simulation packages, these often have the drawback that they are not free. Moreover, while these packages tend to have very good "front ends" getting to the source code that drives the simulation dynamics can be difficult. Thus in these "black boxes" it can be difficult to know what the package is actually doing to the simulation and modification or extension of the code is not easily achieved. The DL\_POLY package is designed to avoid these problems. The source code is freely accessible, extensively documented and highly modular in nature. We wish to avoid the "black box" approach to simulation by providing code that is readily accessible to both verification and modification by users. The package is thus aimed at users who desire intelligent and informed control over their simulations. The package comes as a series of modules that the user "bolts" together for their own particular application. A series of "makefiles" is provided to facilitate this. The package is able to handle a broad range of macro systems - from material science applications (e.g. crystalline and amorphous solids) to simulation of molecules of biological interest (e.g. DNA, enzymes etc) plus a good deal in between. The target machines are Multiple Instruction Multiple Data (MIMD) parallel machines (eg. the Intel iPSC/860 at Daresbury) and single processor workstations. The idea being that users have a code that both works efficiently on a workstation and that will be quite painless to transfer onto parallel supercomputers should they gain access to such facilities. In developing this package beyond a basic capability we are dependent upon the support of the CCP5 community, for example, to supply additional modules for analysis or system generation.

We have adhered to a fairly stringent programming style. All modules are in FORTRAN 77 (modules written in C are acceptable provided they are FORTRAN callable). Variables and arrays are passed as arguments through subroutine calls to facilitate the interfacing of modules. Consequently there are no common blocks present in the source code. The modules are also extensively documented, the release I documentation totals about 220 pages.

Features currently available:

Release I is built around two basic strategies. The first is that of “Replicated Data” meaning that all processors have a complete copy of the atomic co-ordinates, velocities and forces but that evaluation of pair-wise interactions, integration of the equations of motion, application of constraints etc are split equally among the available processors. This approach works well for systems of up to at least 20000 particles and on machines with up to 100 or so processors. It has the advantage that the algorithm will run just as readily on a single processor as on a parallel machine, it is relatively straightforward to program, and results in excellent parallelisation efficiencies. When the number of processors become very large the algorithm suffers from global communication costs (the requirement that arrays etc are summed globally across all processors), but communication overheads also affect other parallelisation strategies. Replicated data is also more memory intensive than other strategies (e.g. domain decomposition) but we do not consider this to be a serious restriction at present. The second intrinsic feature of the code is that it is atomistic in nature - each site is assigned a mass, charge, coordinated etc and evaluation of the Verlet neighbourhood list is done on the basis of site-site separations. At present we do not have the capacity to handle rigid body equations of motion, massless sites and so forth nor to base real space cutoffs on a “molecular group” strategy.

Other features in the package are given below. In each case the required feature is selected by setting an integer variable in the job CONTROL file. As the package is modular in nature additional features are readily added simply by defining new properties for additional values of the integer keys. For example, all evaluations of periodic images take place within one subroutine (“images”) and additional periodic boundary conditions can be introduced simply by modifying this single subroutine. Note that all the features that follow are couched in the Replicated data / Atomistic code framework.

- A selection of periodic boundaries (free space, cubic, orthorhombic, parallelepiped, truncated octahedral, rhomboidal dodecahedral, two dimension periodicity)
- A selection of electrostatic methods (three dimensional Ewald sum, Coulombic, truncated and shifted Coulombic, distant dependant dielectric constant)
- All common atom-atom potentials.
- Flexible bonds and rigid bonds (SHAKE).
- Valence angle and dihedral angle potentials
- A selection of canonical thermostats (Nosé-Hoover and Gaussian constraints) and the non-canonical Berendsen thermostat.
- Single and multiple timestep algorithms
- On-line radial distribution functions and mean squared displacements. The general policy is that analysis routines are intended to be used off-line. This is because the analysis required is usually quite problem specific and so not incorporated into the release package. RDFs and MSDs are however almost universally required and

it makes sense to calculate them “on-line”. They can be turned “on” or “off” by keys in the job CONTROL file.

- Utility modules to aid in generation of initial configurations. These include setting up lattice structures, creating structures from database files (such as PDB files), and adding thermalised water to a structure.
- A series of utilities to aid in generation of force field files - particularly for polymer and bio-simulations. Utilities for helping generate GROMOS and AMBER force fields are included.
- Analysis routines: Correlation functions and statistical analysis. These are designed for off-line analysis.

The figure shows the performance of DL\_POLY as a function of the number of processors in use. The test case is an antifreeze protein emmersed in 1234 SPC waters. Cubic periodic boundary conditions are in use and electrostatic interactions are calculated using the Ewald summation. The plot shows  $\log(\text{average time per step})$  vs the dimension,  $d$ , of the hypercube. The number of processors in use is  $2^d$  and ranges from 2 through to 64 processors. On this plot perfect parallelization would correspond to lines with slope -1. This is what is seen for all subroutines except the reciprocal space summation in the Ewald sum (Ewald1) and for SHAKE. The “perfect parallelisation” is a consequence of the Replicated data strategy as no (or very little) inter-processor communication is required during these parts of the calculation and the work is equally shared among the processors. This is true for tasks such as construction of the Verlet neighbourhood list (“parlst”), and evaluation of real space pair interactions (“Ewald2” and “Sfrfce”). However for the subroutines “Ewald1” and “SHAKE” some inter-processor communication is unavoidable and the timing of the program reflects this especially as the number of processors increases. Eventually these algorithms become communication bound, that is, there is no reduction in execution time by using a larger number of processors.

Features in development:

Display modules for use with AVS.

Features for future development:

We hope to begin working soon on algorithms that will facilitate the simulation of “mega-systems” - ie. multi-million particle simulations. In particular this will mean exploiting the “domain decomposition” parallelization strategy and exploring the use of “heirachical multipole” methods for efficient handling of long range (e.g. electrostatic) potentials.

In addition there are several developments that can be added within the Replicated Data strategy, these include:

- Algorithms for rigid bodies (and massless sites).

- Algorithms for constant pressure simulations
- Inclusion of anisotropic site potentials (e.g Distributed multipoles).

The first two on this list are straightforward to implement and their implementation is dependent on us finding the time to include them in the package. The last item on the list includes some unresolved problems with handling torques on sites contained in a flexible molecular unit. Note that none of these features will be included in DL\_POLY release I.

Availability of the code:

Ultimately the package will be available by anonymous ftp from CCP5. However in the intervening 18 months or so from now the package will be available on restricted release within the U.K. only. Those interested in this limited release should initially contact us at Daresbury for more information.