Science & Technology Facilities Council

Technical Report
RAL-TR-2018-004

# Using mixed precision within DL_POLY's force and energy evaluations: short-range two-body interactions

HS Thorne

May 2018

Enquiries concerning this report should be addressed to:

RAL Library
STFC Rutherford Appleton Laboratory
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

**ISSN 1358-6254**

# Using mixed precision within DL_POLY's force and energy evaluations: short-range two-body interactions

H. Sue Thorne[1,2]

ABSTRACT

Huge developments were made in molecular dynamics simulation methods in the last two decades, particularly in the ability to use large scale parallel platforms. In this report, we consider the use of mixed precision within DL_POLY 4, a molecular dynamics simulation code developed by CCP5. In particular, we investigate using mixed precision within the calculations relating to the short-range force and energy contributions between an atom/ion and a near-by neighbour. We investigate the Ewald real-space kernel, the van der Waals potential calculations and the metal potential calculations.

This work forms part of the CoSeC-funded Software Outlook project.

**Keywords:** Molecular Dynamics, DL_POLY, Ewald Method, Van der Waals Potential, Metal Potential, Mixed Precision, Intel Xeon (IvyBridge), CoSeC, Software Outlook, CCP, CCP5

May 16, 2018

[1]The Hartree Centre, Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK.

[2]Correspondence to: sue.thorne@stfc.ac.uk

# 1 Background and motivation

Huge developments were made in molecular dynamics simulation methods in the last two decades, particularly in the ability to use large-scale parallel platforms. In this report, we consider the use of a mixed-precision methodology within the force calculations. In particular, we consider the calculations relating to the short-range force and energy contributions between an atom/ion and a near-by neighbour. In DL_POLY 4 (versions 4.09 and earlier), real values are stored in double precision and operated on using double precision arithmetic (IEEE Standard 754 is assumed). On modern high performance computing architectures, there is normally the provision to use single precision arithmetic and it is usually significantly faster than the corresponding double precision operation [4]. Additionally, the movement of real data will normally be significantly faster when passing single precision data compared to double precision data. However, the desire to speed-up the code must be balanced with the desire to get an accurate solution. Indeed, if all real values are stored in single precision and operated on using single precision arithmetic, then DL_POLY exhibits a significant loss in accuracy. Thus, it will be necessary to use single precision in a restricted manner, i.e., use mixed precision. In this report, we focus on using a mixed precision approach within the real-space sum from the Ewald sum method and also the van der Waals and metal potentials for short-range forces.

In Section 2, we introduce the algorithmic structure of DL_POLY 4 and the test problems used within this report. The Ewald Method is outlined in Section 3 along with two different mixed precision versions for the short-range interactions: these demonstrate that the structure of the code can be a determining factor as to whether the mixed precision approach is suitable. Mixed precision can only be a viable method if the underlying calculations are such that the use of single precision does not have a detrimental effect on the overall accuracy of the algorithm. In Section 4, mathematical considerations are used to show that the mixed precision approach is not suitable for use within the van der Waals potential and metal potential calculations. Conclusions are drawn in Section 5.

# 2 Algorithmic structure of DL_POLY

In Algorithm 1, we outline the algorithmic structure of DL_POLY. In this report, we are focusing on the use of mixed precision within the calculations for the short-range force contributions. It is important to note that, within each time step, we loop through each atom and its neighbours. As we will see later on, in a mixed precision approach, it is important that this update is done carefully to limit the overheads from precision conversions.

**Algorithm 1** `DL_POLY` outline

Initialise problem
Calculate forces $F(t = 0)$
**for** $it = 1, \ldots, nTimeSteps$ **do**
  Calculate velocity $v(t + \frac{\nabla t}{2})$
  Update positions $r(t + \nabla t)$
  Calculate long-range contributions to system energy and forces $F(t + \nabla t)$
  **for** $i = 1, \ldots, nAtoms$ **do**
    **for** $j$ in neighbours of $i$ **do**
      Update system energy
      Update $F(t + \nabla t)$ with short-range contributions to forces on $j$ due to $i$
    **end for**
  **end for**
  Calculate velocity $v(t + \nabla t)$ and statistics
**end for**
Calculate final statistics

In this report, we only consider the use of mixed precision in the calculation of the short-range contributions to the forces. In particular, the short-range contributions from the Ewald method, Section 3, and the van der Waals and metal potentials, Section 4. In [3], we consider the use of mixed precision within the calculation of the long-range contributions from the Ewald method and found that this approach could be used and maintain accuracy whilst giving up to a 14% reduction in execution time.

## 2.1 Numerical results

All of our tests were run on the Hartree Centre's Napier Compute System [1], which consists of 360 nodes containing 2 x12 core Intel Xeon processors (IvyBridge E5-2697v2 2.7GHz) and 64GB RAM. The interconnect is Infiniband from Mellanox (FDR Connect-IB 56 GB/s). In all our runs, 24 processes per node were requested. DL_POLY 4.09 was used as our base code and all double precision runs use this version. DL_POLY was compiled using Intel MPI (version 5.1.1) and its `mpif90` wrapper, which points to the GNU compiler `gfortran`. The flag `-pg` was used to enable profiling and we always set the environment variable `GMON_OUT_PREFIX` to `gmon.out-‘/bin/uname -n‘` to save the separate profiles for each MPI process.

For each test problem, number of MPI processes ($n_p$), and choice of whether double precision or mixed precision is used within the 3D FFT method, five separate runs were performed. The `gprof` profiler was used to analyse the timing profiles across all of the processes.

As part of DL_POLY, 28 small test problems are provided. Of these, 14 use the Ewald method. In Table 1, we list the test problems used and their attributes: the problems are numbered as in the DL_POLY Manual [5]. When comparing the use of the mixed precision and double precision versions of DL_POLY, we found that the mixed precision version that there was very little or no loss in accuracy in the output from DL_POLY. However, the problems sizes are too small to give meaningful mixed-precision and double precision comparisons.

| Problem | Total system size | Temp(K) | $r_{cut}$ | $r_{vdw}$ | Description |
|---|---|---|---|---|---|
| 01 | 27000 ions | 500 | 12.0 | 12.0 | Unit electric charges on sodium and chloride |
| 02 | 51737 atoms | 300 | 10.0 | 10.0 | 200 DMPC molecules in 9379 water molecules |
| 03 | 69120 atoms | 1000 | 12.03 | 7.6 | Potassium sodium disilicate glass |
| 04 | 99120 atoms | 300 | 8.0 | 8.0 | 8 Gramacidin A molecules in 32096 water molecules |
| 07 | 12428 atoms | 300 | 8.0 | 8.0 | Lipid bilayer in water |
| 08 and 09 | 8000 charged points and 4000 shells | 3000 | 6.0 | 6.0 | MgO with adiabatic and with relaxed shell model |
| 10 | 500 ions and 39000 atoms | 300 | 8.0 | 8.0 | Potential of mean force on K+ in water |
| 18 | 34992 atoms | 25 | 7.0 | 7.0 | SPC IceVII Water with constraint bonds |
| 19 | 34992 atoms | 25 | 7.0 | 7.0 | SPC IceVII Water with rigid bodies |
| 20 | 28816 atoms | 295 | 9.0 | 9.0 | 64 NaCl ion pairs with 4480 water molecules represented by constraint bonds and 4416 water molecules represented by rigid bodies |
| 21 | 29052 particles | 295 | 9.0 | 9.0 | 7263 TIP4P rigid body water molecules |
| 22 | 44352 ions | 400 | 9.0 | 8.0 | Ionic liquid dimethylimidazolium |
| 23 | 23712 ions | 310 | 10.1 | 10.1 | 600 molecules of calcite in 6904 water molecules |

Table 1: Standard DL_POLY test problems using the Ewald Method and their attributes. The total system size, system temperature, the short-range cut-off ($r_{cut}$), and the cut-off value used by the van der Waals method ($r_{vdw}$) are provided.

To obtain more realistic results, we enlarge some of the test problems to enable us to examine the effect of communication between processes and give a better idea of the mixed precision behaviour. To enlarge the test problems, we use DL_POLY's `nfold` facility. For test problem 02, we had to additionally alter the CONTROL file by appropriately scaling the maximum number of $k$-vector indices in the Ewald specification line. We append "b" to the test problem number for problems that were expanded by doubling the $x$, $y$ and $z$ dimensions; "c" is

appended to the problem number for problems where the $x$, $y$ and $z$ dimensions are increased by a factor of 5. The original problem is replicated 8 (125) times and shifted appropriately to fill the enlarged space. With the enlarged systems, it is necessary to increase the number of MPI processes employed to solve the problems.

# 3 The Ewald Method

In molecular dynamics, the conditionally convergent Coulomb sum is used to describe the system energy contributions from ionic interactions in periodic charged systems. The Ewald Sum replaces the Coulomb sum by three distinct sums with guaranteed convergence:

$$E = E^S + E^L + E^{SELF},$$

where $E^S$, the *real space* sum, is cast in normal physical space and represents the short-range interactions; $E^L$, the *reciprocal space* sum, is cast in the reciprocal space of the unit cell and represents the long-range terms; $E^{SELF}$ is the self interaction term. Suppose we have $N$ ions in a vacuum at locations $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N$ with point charges $q_1, q_2, \ldots, q_N$, respectively. We define $\epsilon_0 = 8.854 \times 10^{-12}\mathrm{C}^2\mathrm{N}^{-1}\mathrm{m}^{-2}$ to be the electric constant (or vacuum permittivity). Let the ions be subjected to periodic boundary conditions, which we describe using three repeat vectors $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, which form the *supercell*. Thus, if there is an ion with charge $q_i$ at location $\mathbf{r}_i$, then there are also ions with charge $q_i$ at $\mathbf{r}_i + n_1\mathbf{c}_1 + n_2\mathbf{c}_2 + n_3\mathbf{c}_3$, where $n_1$, $n_2$, and $n_3$ are arbitrary integers. We simplify the notation by writing an arbitrary repeat vector as $\mathbf{n}L$, where $L$ represents the supercell. If we further assume that the charge distribution is of Guassian form with standard deviation $\sigma$ and the supercell has volume $V$, then $E^S$, $E^L$ and $E^{SELF}$ are defined as

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}\,\mathbf{erfc}\left(\frac{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}{\sqrt{2}\sigma}\right), \tag{1}$$

$$E^L = \frac{1}{2V\epsilon_0}\sum_{\mathbf{k}\neq 0}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{k^2}e^{i\mathbf{k}\cdot(\mathbf{r}_i - \mathbf{r}_j)e^{-\sigma^2 k^2/2}}, \tag{2}$$

$$E^{SELF} = \frac{1}{4\pi\epsilon_0}\frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{N} q_i^2. \tag{3}$$

In the above, $\sum_{\mathbf{k}\neq 0}$ is the summation over the reciprocal lattice of $L$, which is derived using the Fourier transform, and $k = |\mathbf{k}|$.

The real space sum, $E^S$, quickly converges and, hence, contributions to the forces from this sum are only calculated for the nearby neighbours of an atom and not all of the neighbours [2]. Thus, we obtain

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{i=1}^{N}\sum_{j\ \text{near}\ i} q_i q_j u\left(|\mathbf{r}_i - \mathbf{r}_j|\right), \quad \text{where} \quad u(r) = \frac{1}{r}\,\mathbf{erfc}\left(\frac{r}{\sqrt{2}\sigma}\right). \tag{4}$$

Ions at $\mathbf{r}_i$ and $\mathbf{r}_j$ are considered to be nearby neighbours if $|\mathbf{r}_i - \mathbf{r}_j| < r_{cut}$, where $r_{cut}$ is calculated within DL_POLY or provided by the user. In DL_POLY's implementation, during the initialisation phase, the value of $u(r)$ is calculated at uniformly distributed points $r$ in $\left(0, \frac{\tau}{\sqrt{2}\sigma}\right)$ and stored. When computing $E^S$, these stored values are interpolated using a 3-point B-spline scheme to give a good approximation to the required value.

The forces on ion $i$ due to the Coloumbic interactions are defined as

$$\mathbf{f}_i = -\frac{\nabla E}{\nabla \mathbf{r}_i} = -\frac{\nabla E^S}{\nabla \mathbf{r}_i} - \frac{\nabla E^L}{\nabla \mathbf{r}_i}.$$

For the short-range interactions, we have

$$-\frac{\nabla E^S}{\nabla \mathbf{r}_i} = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{j=1}^{N} q_i q_j\left(\mathbf{r}_i - \mathbf{r}_j\right)\hat{u}\left(|\mathbf{r}_i - \mathbf{r}_j|\right), \quad \text{where} \quad \hat{u}(r) = \left(u(r) - \frac{1}{\sqrt{2}\sigma}\frac{2}{\sqrt{\pi}}e^{-\frac{r^2}{2\sigma^2}}\right)/r^2. \tag{5}$$

In the same manner as for the energy calculations, $\hat{u}(r)$ is calculated at uniformly distributed points $r$ in $\left(0, \frac{\tau}{\sqrt{2}\sigma}\right)$ and stored. A 3-point interpolation scheme is then used to produce a good approximation to the individual terms needed in the calculation of $\frac{\nabla E^S}{\nabla \mathbf{r}_i}$.

### 3.1 Mixed precision: version 1

Our first mixed precision variant for the Ewald real space kernel is summarised as follows:

- Initialisation:
  - Compute $u(r)$ and $\hat{u}(r)$ at uniformly distributed points in $\left(0, \frac{\tau}{\sqrt{2}\sigma}\right)$ using double precision and store these values in single precision;
- During the inner iteration of the short-range computation:
  - Using single precision, compute $u_{ij}$ and $\hat{u}_{ij}$, approximations to $u(|\mathbf{r}_i - \mathbf{r}_j|)$ and $\hat{u}(|\mathbf{r}_i - \mathbf{r}_j|)$, respectively, via 3-point interpolation;
  - Convert $q_i$, $q_j$ and $\mathbf{r}_i - \mathbf{r}_j$ to single precision;
  - Perform the single precision calculations $s_{ij} = q_i q_j u_{ij}$ and $\hat{s}_{ij} = q_i q_j \hat{u}_{ij} (\mathbf{r}_i - \mathbf{r}_j)$;
  - Convert $s_{ij}$ and $\hat{s}_{ij}$ to double precision before including in the energy and force calculations;

It was a fairly simple process to convert the Ewald real space kernely to use the above mixed precision approach.

In our numerical results, we report times that are averaged (mean) across all five test runs and all MPI processes. We are particularly interested in

- *ewald_real*, the total time spent performing the Ewald real space kernel by an MPI process;
- $force\_energy$, the total time spent performing the force and energy calculations by an MPI process;
- *total*, the total DL_POLY execution time for an MPI process.

In Table 2, we compare the values of *ewald_real*, $force\_energy$ and *total* for the double precision version of DL_POLY (version 4.09) and our mixed precision (version 1) code. For the majority of codes, the mixed precision version has an overhead of between 11 and 14% for the Ewald real space kernel calculations. This then results in a 1-2% increase in the total execution times. Thus, although the floating point calculation times will be smaller, the conversions between single and double precision are resulting in an overall increase in execution time on this architecture. At time step $t$, let $neighs(i,t)$ be the nearby neighbours of atom/ion $i$ that are used in the short-range force calculations. Suppose that the system has *natoms* atoms/ions, then, ignoring conversions in the initialisation process, for each time step, $t$, version 1 of our mixed precision Ewald real space kernel requires:

- $11 \times natoms$ double precision floating point operations;
- $53 \sum_{i=1}^{natoms} neighs(i,t)$ single precision floating point operations;
- $4 \times natoms + 5 \sum_{i=1}^{natoms} neighs(i,t)$ conversions from double precision to single precision;
- $14 \times natoms + 3 \sum_{i=1}^{natoms} neighs(i,t)$ conversions from single precision to double precision.

The large number of precision conversions are the major reason for the overheads and, additionally, some values are being converted more than once in this naive mixed precision implementation of the Ewald real space kernel.

| | | *ewald_real* | | | *force_energy* | | | *total* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Problem | $n_p$ | mixed | double | ratio | mixed | double | ratio | mixed | double | ratio |
| 02b | 2 | 32.81 | 28.76 | 1.14 | 163.85 | 160.77 | 1.02 | 200.97 | 197.99 | 1.02 |
| | 4 | 16.56 | 14.61 | 1.13 | 82.80 | 81.16 | 1.02 | 102.45 | 100.84 | 1.02 |
| | 8 | 8.31 | 7.44 | 1.12 | 41.86 | 41.06 | 1.02 | 52.44 | 51.65 | 1.02 |
| | 16 | 4.19 | 3.74 | 1.12 | 22.36 | 21.92 | 1.02 | 28.22 | 27.77 | 1.02 |
| | 32 | 2.12 | 1.90 | 1.12 | 11.69 | 11.50 | 1.02 | 14.97 | 14.78 | 1.01 |
| 04b | 8 | 8.37 | 7.37 | 1.14 | 37.49 | 36.67 | 1.02 | 61.46 | 60.67 | 1.01 |
| | 16 | 4.26 | 3.74 | 1.14 | 19.49 | 19.04 | 1.02 | 33.21 | 32.87 | 1.01 |
| | 24 | 2.83 | 2.51 | 1.13 | 13.82 | 13.55 | 1.02 | 24.76 | 24.62 | 1.01 |
| | 32 | 2.15 | 1.90 | 1.13 | 10.07 | 9.89 | 1.02 | 17.22 | 17.05 | 1.01 |
| 07b | 8 | 2.27 | 1.89 | 1.20 | 20.92 | 20.62 | 1.01 | 41.00 | 40.80 | 1.01 |
| | 16 | 1.14 | 0.96 | 1.19 | 12.88 | 12.75 | 1.01 | 24.66 | 24.51 | 1.01 |
| | 24 | 0.75 | 0.64 | 1.17 | 11.62 | 11.46 | 1.01 | 21.58 | 21.10 | 1.02 |
| | 32 | 0.58 | 0.49 | 1.19 | 7.34 | 7.19 | 1.02 | 13.85 | 13.70 | 1.01 |
| 10b | 8 | 3.14 | 2.77 | 1.13 | 14.55 | 14.18 | 1.03 | 24.72 | 24.23 | 1.02 |
| | 16 | 1.59 | 1.40 | 1.13 | 7.59 | 7.38 | 1.03 | 13.09 | 12.88 | 1.02 |
| | 24 | 1.05 | 0.95 | 1.11 | 5.25 | 5.14 | 1.02 | 9.09 | 8.96 | 1.01 |
| | 32 | 0.80 | 0.72 | 1.12 | 3.79 | 3.72 | 1.02 | 6.77 | 6.67 | 1.02 |
| 18b | 8 | 11.64 | 10.35 | 1.12 | 33.41 | 32.19 | 1.04 | 40.09 | 38.89 | 1.03 |
| | 16 | 5.91 | 5.27 | 1.12 | 17.52 | 16.93 | 1.03 | 21.34 | 20.74 | 1.03 |
| | 24 | 4.18 | 3.54 | 1.18 | 12.78 | 11.62 | 1.10 | 15.64 | 14.38 | 1.09 |
| | 32 | 2.99 | 2.67 | 1.12 | 9.31 | 8.99 | 1.04 | 11.45 | 11.14 | 1.03 |
| 19b | 8 | 3.32 | 2.99 | 1.11 | 10.45 | 10.07 | 1.04 | 11.95 | 11.57 | 1.03 |
| | 16 | 1.69 | 1.52 | 1.11 | 5.41 | 5.22 | 1.04 | 6.34 | 6.15 | 1.03 |
| | 24 | 1.13 | 1.00 | 1.13 | 3.72 | 3.60 | 1.03 | 4.44 | 4.31 | 1.03 |
| | 32 | 0.85 | 0.76 | 1.12 | 2.77 | 2.68 | 1.03 | 3.27 | 3.19 | 1.02 |
| 02c | 144 | 7.27 | 6.49 | 1.12 | 49.44 | 48.81 | 1.01 | 70.93 | 70.31 | 1.01 |
| | 192 | 5.47 | 4.90 | 1.12 | 34.36 | 33.84 | 1.02 | 52.01 | 51.55 | 1.01 |
| 04c | 144 | 7.32 | 6.44 | 1.14 | 36.84 | 36.03 | 1.02 | 74.94 | 74.16 | 1.01 |
| | 192 | 5.49 | 4.84 | 1.13 | 27.16 | 26.56 | 1.02 | 55.38 | 54.81 | 1.01 |
| 18c | 96 | 4.96 | 4.40 | 1.13 | 17.66 | 17.14 | 1.03 | 28.92 | 28.46 | 1.02 |
| | 144 | 3.31 | 2.94 | 1.12 | 11.78 | 11.64 | 1.01 | 18.29 | 18.27 | 1.00 |

Table 2: The values of *ewald_real*, *force_energy* and *total* for the double precision version of DL_POLY (version 4.09) and our mixed precision (version 1) code.

## 3.2 Mixed precision: version 2

Version 2 of our mixed precision approach aims to significantly reduce the overheads from the precision conversion by defining force arrays $F_{x,t}$, $F_{y,t}$ and $F_{z,t}$ at time time $t$ and directions $x$, $y$, and $z$, respectively, as

$$F_{l,t} = F^d_{l,t} + double(F^s_{l,t}), \quad l \in \{x, y, z\},$$

where $F^d_{l,t}$ holds double precision data, $F^s_{l,t}$ holds single precision data and $double(\cdot)$ represents the conversion from single to double precision. The amendments to the standard DL_POLY algorithm are as follows:

- Initialisation:
  - Compute $u(r)$ and $\hat{u}(r)$ at uniformly distributed points in $\left(0, \frac{\tau}{\sqrt{2}\sigma}\right)$ and store these values in single precision;
- For each time step:
  - Initialise $F^s_{x,t}(:) = 0$, $F^s_{y,t}(:) = 0$ and $F^s_{x,t}(:) = 0$

– During the inner iteration of the short-range computation:
  * Using single precision, compute $u_{ij}$ and $\hat{u}_{ij}$, approximations to $u(|\mathbf{r}_i - \mathbf{r}_j|)$ and $\hat{u}(|\mathbf{r}_i - \mathbf{r}_j|)$, respectively, via 3-point interpolation;
  * Convert $q_i$, $q_j$ and $\mathbf{r}_i - \mathbf{r}_j$ to single precision and perform the single precision calculations $q_i q_j u_{ij}$ and $q_i q_j \hat{u}_{ij} (\mathbf{r}_i - \mathbf{r}_j)$; convert energy contribution to double precision before including in the overall energy; add force contributions to single precision force arrays
– Convert entries in single precision force arrays $F^s_{x,t}$, $F^s_{y,t}$ and $F^s_{x,t}$ to double precision and add to $F_{x,t}$, $F_{y,t}$ and $F_{x,t}$, respectively.

In DL_POLY, linked lists are used to speed-up the identification of the short-range neighbours of an atom/ion. Without the use of linked lists (or equivalent method), each atom in the system would need to be tested to see if it is a short-range neighbour of the current atom $i$, resulting in $O(natoms^2)$ tests: linked lists reduces this to $O(natoms)$ tests. The linked lists are stored in an array of length $mxlist$, the maximum length of the link list. Ignoring conversions in the initialisation process, for each time step, $t$, version 1 of our mixed precision Ewald real space kernel requires:

- 4 array allocations and deallocations of length at least $mxlist$ (this could be removed from the time step loop with major restructuring of the DL_POLY code);
- 3 array allocations, initialisations and deallocations of length equal to the force arrays, i.e., at least $natoms$ (this could be removed from the time step loop with further restructuring of the DL_POLY code);
- $14 \times natoms$ double precision floating point operations;
- $3 \times natoms + 53 \sum_{i=1}^{natoms} neighs(i,t)$ single precision floating point operations;
- $4 \times natoms + 5 \sum_{i=1}^{natoms} neighs(i,t)$ conversions from double precision to single precision;
- $15 \times natoms$ conversions from single precision to double precision.

Thus, there is a significant saving in the number of conversions from single to double precision but there is an increase in the number of single and double precision floating point operations, and additional array allocations, initialisations and deallocations.

The values of $ewald\_real$, $force\_energy$, $force\_energy - ewald\_real$ and $total$ for version 2 of the mixed precision code and the original double precision code are provided in Table 3. For each test problem and each time measurement, the double precision version is faster than version 2. Comparing the two mixed precision versions, the Ewald real space kernel time is significantly improved when version 2 is used instead of version 1. However, the overall time calculating force and energy updates is worse for version 2. The main reason for this is the additional array allocations, initialisations and deallocations. A complete restructure of DL_POLY would reduce these overheads but the work being performed in single precision within the Ewald real space kernel is not large enough relative to the number of necessary conversions to produce an overall reduction in computation time. In molecular dynamics simulations, the short-range force and energy updates are formed from a number of components and the Ewald summation is just one of these possible components. As will be seen in the following section, some of these components must be performed in double precision and, hence, it is not possible to further reduce the number of conversions from single to double precision.

# 4 The van der Waals method

In Section 3, we considered the use of the Ewald summation in the calculation of short-range energy and force updates due to electrostatics. As well as these contributions, there are also two-body short-range contributions from the van der Waals potentials and metal potentials. DL_POLY contains numerous options for further short ranged (van der Waals) potentials, see Appendix. As in Section 3, the individual components for the energy and

| | | *ewald_real* | | | *force_energy* | | | *force_energy − ewald_real* | | | *total* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem | $n_p$ | mixed2 | double | ratio | mixed2 | double | ratio | mixed2 | double | ratio | mixed2 | double | ratio |
| 02b | 2 | 29.36 | 28.76 | 1.02 | 165.21 | 160.77 | 1.03 | 135.85 | 132.01 | 1.03 | 202.36 | 197.99 | 1.02 |
| 02b | 4 | 14.88 | 14.61 | 1.02 | 83.66 | 81.16 | 1.03 | 68.79 | 66.55 | 1.03 | 103.34 | 100.84 | 1.02 |
| 02b | 8 | 7.55 | 7.44 | 1.01 | 42.36 | 41.06 | 1.03 | 34.81 | 33.61 | 1.04 | 52.94 | 51.65 | 1.03 |
| 02b | 16 | 3.80 | 3.74 | 1.02 | 22.59 | 21.92 | 1.03 | 18.79 | 18.18 | 1.03 | 28.44 | 27.77 | 1.02 |
| 02b | 32 | 1.92 | 1.90 | 1.01 | 11.83 | 11.50 | 1.03 | 9.91 | 9.60 | 1.03 | 15.12 | 14.78 | 1.02 |
| 04b | 8 | 7.57 | 7.37 | 1.03 | 38.01 | 36.67 | 1.04 | 30.44 | 29.30 | 1.04 | 61.96 | 60.67 | 1.02 |
| 04b | 16 | 3.86 | 3.74 | 1.03 | 19.77 | 19.04 | 1.04 | 15.92 | 15.30 | 1.04 | 33.61 | 32.87 | 1.02 |
| 04b | 24 | 2.59 | 2.51 | 1.03 | 14.05 | 13.55 | 1.04 | 11.45 | 11.04 | 1.04 | 25.09 | 24.62 | 1.02 |
| 04b | 32 | 1.96 | 1.90 | 1.03 | 10.23 | 9.89 | 1.03 | 8.26 | 7.99 | 1.03 | 17.32 | 17.05 | 1.02 |
| 07b | 8 | 2.10 | 1.89 | 1.11 | 21.41 | 20.62 | 1.04 | 19.31 | 18.73 | 1.03 | 41.40 | 40.80 | 1.01 |
| 07b | 16 | 1.06 | 0.96 | 1.11 | 13.24 | 12.75 | 1.04 | 12.17 | 11.79 | 1.03 | 25.04 | 24.51 | 1.02 |
| 07b | 24 | 0.72 | 0.64 | 1.11 | 11.98 | 11.46 | 1.04 | 11.26 | 10.82 | 1.04 | 22.00 | 21.10 | 1.04 |
| 07b | 32 | 0.53 | 0.49 | 1.09 | 7.51 | 7.19 | 1.05 | 6.98 | 6.70 | 1.04 | 13.98 | 13.70 | 1.02 |
| 10b | 8 | 2.83 | 2.77 | 1.02 | 14.73 | 14.18 | 1.04 | 11.90 | 11.41 | 1.04 | 24.91 | 24.23 | 1.03 |
| 10b | 16 | 1.46 | 1.40 | 1.04 | 7.68 | 7.38 | 1.04 | 6.22 | 5.98 | 1.04 | 13.23 | 12.88 | 1.03 |
| 10b | 24 | 0.98 | 0.95 | 1.03 | 5.31 | 5.14 | 1.03 | 4.34 | 4.20 | 1.03 | 9.13 | 8.96 | 1.02 |
| 10b | 32 | 0.73 | 0.72 | 1.02 | 3.86 | 3.72 | 1.04 | 3.12 | 3.00 | 1.04 | 6.81 | 6.67 | 1.02 |
| 18b | 8 | 10.65 | 10.35 | 1.03 | 33.72 | 32.19 | 1.05 | 23.06 | 21.84 | 1.06 | 40.40 | 38.89 | 1.04 |
| 18b | 16 | 5.42 | 5.27 | 1.03 | 17.75 | 16.93 | 1.05 | 12.34 | 11.66 | 1.06 | 21.56 | 20.74 | 1.04 |
| 18b | 24 | 3.81 | 3.54 | 1.08 | 12.96 | 11.62 | 1.12 | 9.16 | 8.08 | 1.13 | 15.85 | 14.38 | 1.10 |
| 18b | 32 | 2.75 | 2.67 | 1.03 | 9.45 | 8.99 | 1.05 | 6.71 | 6.31 | 1.06 | 11.60 | 11.14 | 1.04 |
| 19b | 8 | 3.02 | 2.99 | 1.01 | 10.57 | 10.07 | 1.05 | 7.55 | 7.08 | 1.07 | 12.08 | 11.57 | 1.04 |
| 19b | 16 | 1.56 | 1.52 | 1.03 | 5.49 | 5.22 | 1.05 | 3.93 | 3.70 | 1.06 | 6.43 | 6.15 | 1.05 |
| 19b | 24 | 1.05 | 1.00 | 1.05 | 3.78 | 3.60 | 1.05 | 2.73 | 2.60 | 1.05 | 4.50 | 4.31 | 1.04 |
| 19b | 32 | 0.79 | 0.76 | 1.05 | 2.81 | 2.68 | 1.05 | 2.02 | 1.93 | 1.05 | 3.32 | 3.19 | 1.04 |
| 02c | 144 | 6.59 | 6.49 | 1.02 | 49.98 | 48.81 | 1.02 | 43.39 | 42.32 | 1.03 | 71.57 | 70.31 | 1.02 |
| 02c | 192 | 4.97 | 4.90 | 1.02 | 34.88 | 33.84 | 1.03 | 29.91 | 28.94 | 1.03 | 52.60 | 51.55 | 1.02 |
| 04c | 144 | 6.64 | 6.44 | 1.03 | 37.33 | 36.03 | 1.04 | 30.69 | 29.59 | 1.04 | 75.59 | 74.16 | 1.02 |
| 04c | 192 | 5.01 | 4.84 | 1.04 | 27.54 | 26.56 | 1.04 | 22.53 | 21.72 | 1.04 | 55.76 | 54.81 | 1.02 |
| 18c | 96 | 4.50 | 4.40 | 1.02 | 17.82 | 17.14 | 1.04 | 13.32 | 12.74 | 1.05 | 29.06 | 28.46 | 1.02 |
| 18c | 144 | 3.00 | 2.94 | 1.02 | 11.94 | 11.64 | 1.03 | 8.93 | 8.69 | 1.03 | 18.46 | 18.27 | 1.01 |

Table 3: The values of *ewald_real*, *force_energy*, *force_energy* − *ewald_real* and *total* for version 2 of the mixed precision code and the original double precision code.

force updates can be calculated using mixed precision in a number of ways. The first proposal, as in the short-range Ewald summation, is to calculate each individual short-range component using single precision, convert this value to double precision and add to the overall energy/force value. As in Section 3.1, the conversion of each individual component from single to double precision forms a large overhead and the reduced floating point execution times do not compensate enough for this overhead. For TEST02, there was a 10% increase in the total van der Waals calculation time when 8 MPI processes were used, resulting in the overall simulation time increasing by 4%; for 1 MPI process, there was an 18% increase in van der Waals calculation time and a 4% increase in the total simulation time.

In Section 3.2, the short-range energy and force contributions were calculated in single precision and the force updates accumulated in single precision arrays throughout the time step before converting to double precision at the end of the time step and adding to the double precision force arrays. The range of values for $|\mathbf{r}_i - \mathbf{r}_j|$ lie in the range $[0, r_{vdw}]$, where $r_{vdw}$ is measured in angstroms ($10^{-10}$m). The various forms of van der Waals potentials available all involve high orders or exponentials involving $|\mathbf{r}_i - \mathbf{r}_j|$. Thus, each short-range component can vary by many orders of magnitude. In Figure 1, we provide the absolute value of the short-range van der Waals potential energy contribution for different values of $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| \in [1, r_{vdw}]$ for potassium-potassium

interactions in TEST10. Test 10 uses the Lennard-Jones potential with $\epsilon = 1.356523$, $\sigma = 3.13$ and $r_{vdw} = 8$. We also provide the absolute value of the force multiplier $\alpha_f$, where the force on atom $j$ due to the potential energy between atoms $i$ and $j$ is $\underline{f}_j = \alpha_f (\mathbf{r}_i - \mathbf{r}_j)$ (the range of values for the actual force components will be even wider). In single precision, between 7 and 8 digits are stored. Let $a$ and $b$ be stored using single precision values, $a$ and $b$, and assume that $|a| > 10^8 |b|$, then adding $a$ and $b$ together (using IEEE standard 754) results in the value $a$. In DL_POLY, the number of "small" components can be very large and, hence, the cumulative effect of these can be significant but they may be missed if single precision arithmetic is used. When this approach was tested within DL_POLY, we immediately saw this phenomenon and, hence, accuracy of the simulation was lost. Therefore, we do not recommend using a mixed-precision approach for the van der Waals calculations. Similarly, the different options for metal potentials use equations contain exponentials of $|r_i - r_j|$ or high degree polynomials involving $|r_i - r_j|$. Hence, this approach is not recommended for the metal potentials.

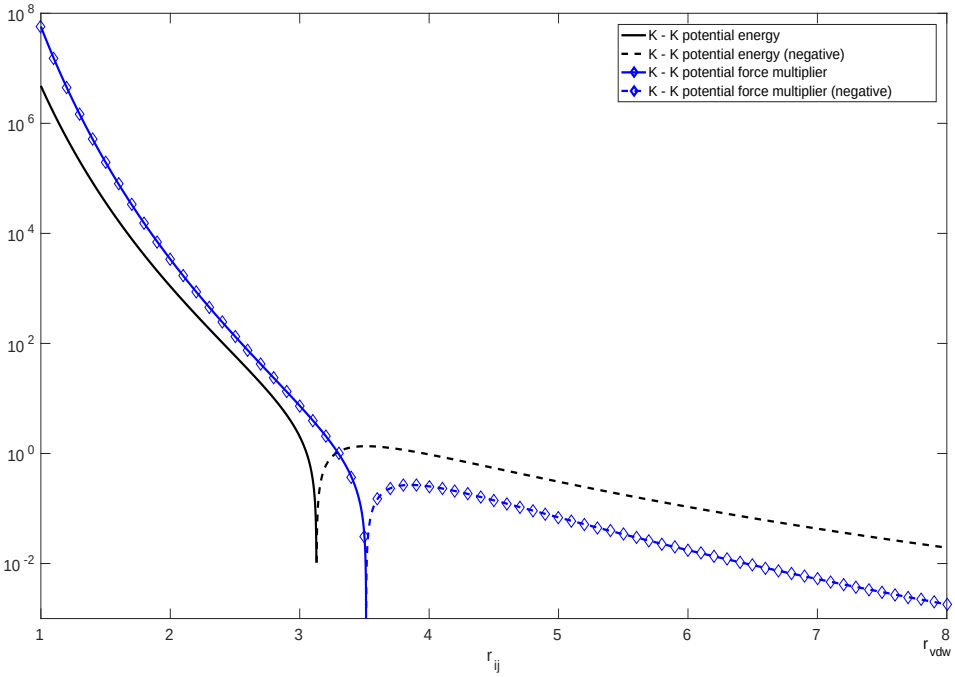


Figure 1: Interactions between two potassium ions in TEST10: the absolute value of the short-range van der Waals (Lennard-Jones) potential energy contributions for different values of $r_{ij} = |r_i - r_j|$ and the absolute value of the force multiplier $\alpha_f$, where the force on atom $j$ due to the potential energy between atoms $i$ and $j$ is $\underline{f}_j = \alpha_f (r_i - r_j)$. Solid lines indicate that the energy (force multiplier) is positive in value; dashed lines indicate that the value has been negated.

# 5 Conclusions

We conclude that, whilst mixed-precision can theoretically be used within some of the short-range energy and force calculations with very minor losses in accuracy, our tests show that it is not advisable when the code is being run on Intel Xeon-based processors because the overheads from data conversions (single to double precision and vice versa) outweigh the gains from using single precision within the calculations. In comparison, in [3] we saw that mixed-precision can be successfully used within the Ewald summation's reciprocal space sum for long-rage interactions to give accurate results and time/energy savings.

**Acknowledgements** The author would like to thank Alin Marin Elena and Michael Seaton for the discussions we had with regards using mixed precision within DL_POLY, and for proposing this work.

# References

[1] See http://community.hartree.stfc.ac.uk/wiki/site/admin/resources.html.

[2] J. Kolafa and J. W. Perram, *Cutoff errors in the ewald summation formulae for point charge systems*, Molecular Simulation, 9 (1992), pp. 351–368.

[3] H. S. Thorne, *Using mixed precision within DL_POLY's force and energy evaluations: long-range interactions and fast Fourier transforms*, Tech. Rep. RAL-TR-2018-003, 2018.

[4] H. S. Thorne, L. Mason, and A. D. Taylor, *Mixed precision: Is it the holy grail for software efficiency?*, in 2nd Conference of Research Software Engineers (RSE17), Manchester, U.K., September 7-8 2017. http://purl.org/net/epubs/work/34840371.

[5] I. Todorov and W. Smith, *The DL_POLY_4 User Manual (version 4.08)*. March 2016.

# Appendix: DL_POLY's van der Waals Potential Energy/Forces options

DL_POLY provides 12 different short-range pair force options, as listed below. In the following, we define $\underline{\mathrm{r}}_{ij} = r_i - r_j$ and $r_{ij} = |r_{ij}|$. $U(r_{ij})$ represents the potential energy component from the pair of atoms $i$ and $j$; the force on atom $j$ due to this potential is formally defined as $\underline{\mathrm{f}}_j = -\frac{1}{r_{ij}}\left[\frac{\nabla}{\nabla r_{ij}} U(r_{ij})\right]\underline{\mathrm{r}}_{ij}$.

- 12-6 potential:

$$\begin{aligned} U(r_{ij}) &= \left(\frac{A}{r_{ij}^{12}}\right) - \left(\frac{B}{r_{ij}^{6}}\right) \\ \underline{\mathrm{f}}_j &= 6\left(\frac{2A}{r_{ij}^{14}} -\right) \end{aligned}$$

- Lennard-Jones potential:

$$\begin{aligned} U(r_{ij}) &= 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \\ \underline{\mathrm{f}}_j &= 24\epsilon\left[2\frac{\sigma^1 2}{r_{ij}^1 4} - \frac{\sigma^6}{r_{ij}^8}\right]\underline{\mathrm{r}}_{ij} \end{aligned}$$

- $n$-$m$ potential:

$$\begin{aligned} U(r_{ij}) &= \frac{E_0}{n-m}\left[m\left(\frac{r_0}{r_{ij}}\right)^n - n\left(\frac{r_0}{r_{ij}}{}^m\right)\right] \\ \underline{\mathrm{f}}_j &= \frac{nmE_0}{n-m}\left[\left(\frac{r_0}{r_{ij}}\right)^n - \left(\frac{r_0}{r_{ij}}\right)^m\right]\frac{\underline{\mathrm{r}}_{ij}}{r_{ij}^2} \end{aligned}$$

- Buckingham potential:

$$\begin{aligned} U(r_{ij}) &= A\exp\left(-\frac{r_{ij}}{\rho}\right) - \frac{C}{r_{ij}^6} \\ \underline{\mathrm{f}}_j &= \left[\frac{A}{\rho r_{ij}}\exp\left(-\frac{r_{ij}}{\rho}\right) - \frac{6C}{r_{ij}^8}\right]\underline{\mathrm{r}}_{ij} \end{aligned}$$

- Born-Huggins-Meye potential:

$$\begin{aligned} U(r_{ij}) &= A\exp\left[B\left(\sigma - r_{ij}\right)\right] - \frac{C}{r_{ij}^6} - \frac{D}{r_{ij}^8} \\ \underline{\mathrm{f}}_j &= \left[\frac{AB}{r_{ij}}\exp\left[B\left(\sigma - r_{ij}\right)\right] - \frac{6C}{r_{ij}^8} - \frac{8D}{r_{ij}^{10}}\right]\underline{\mathrm{r}}_{ij} \end{aligned}$$

- Hydrogen-bond (12-10) potential:

$$\begin{aligned} U(r_{ij}) &= \left(\frac{A}{r_{ij}^{12}}\right) - \left(\frac{B}{r_{ij}^{10}}\right) \\ \underline{\mathrm{f}}_j &= \left[\left(\frac{12A}{r_{ij}^{14}}\right) - \left(\frac{10B}{r_{ij}^{12}}\right)\right]\underline{\mathrm{r}}_{ij} \end{aligned}$$

- Shifted force n-m protential (aka Mie):

$$\begin{aligned} U(r_{ij}) &= \frac{\alpha E_0}{(n-m)}\left[m\beta^n\left\{\left(\frac{r_0}{r_{ij}}\right)^n - \left(\frac{1}{\gamma}\right)^n\right\} - n\beta^m\left\{\left(\frac{r_0}{r_{ij}}\right)^m - \left(\frac{1}{\gamma}\right)^m\right\}\right] + \\ &\quad \frac{nm\alpha E_0}{(n-m)}\left(\frac{r_{ij}-\gamma r_0}{\gamma r_0}\right)\left\{\left(\frac{\beta}{\gamma}\right)^n - \left(\frac{\beta}{\gamma}\right)^m\right\}, \\ \underline{\mathrm{f}}_j &= \frac{\alpha n m E_0}{(n-m)}\left[\left(\frac{\beta r_0}{r_{ij}}\right)^n - \left(\frac{\beta r_0}{r_{ij}}\right)^m + \left(\frac{r_{ij}}{\gamma r_0}\right)\left\{\left(\frac{\beta}{\gamma}\right)^n - \left(\frac{\beta}{\gamma}\right)^m\right\}\right]\frac{\underline{\mathrm{r}}_{ij}}{r_{ij}^2}, \end{aligned}$$

  where

$$\begin{aligned} \alpha &= \frac{(n-m)}{\left[n\beta^m\left(1+\left(m\gamma^{-1}-m-1\right)\gamma^{-m}\right) + m\beta^n\left(1+\left(n\gamma^{-1}-n-1\right)\gamma^{-n}\right)\right]}, \\ \beta &= \gamma\left(\frac{\gamma^{m+1}-1}{\gamma^{n+1}-1}\right)^{\frac{1}{n-m}} \\ \gamma &= \frac{r_{\mathrm{cut}}}{r_0}. \end{aligned}$$

- Morse potential:

$$\begin{aligned} U(r_{ij}) &= E_0\left[\left\{1-\exp\left(-k\left(r_{ij}-r_0\right)\right)\right\}^2 - 1\right] \\ \underline{\mathrm{f}}_j &= -\frac{2kE_0}{r_{ij}}\exp\left(-k\left(r_{ij}-r_0\right)\right)\left\{1-\exp\left(-k\left(r_{ij}-r_0\right)\right)\right\}\underline{\mathrm{r}}_{ij} \end{aligned}$$

- Shifted Weeks-Chandler-Anderson potential:

$$\begin{aligned} U(r_{ij}) &= \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r_{ij}-\Delta}\right)^{12} - \left(\frac{\sigma}{r_{ij}-\Delta}\right)^6\right] + \epsilon & : \quad r_{ij} < 2^{\frac{1}{6}}\sigma + \Delta \\ 0 & : \quad r_{ij} \geq 2^{\frac{1}{6}}\sigma + \Delta \end{cases} \\ \underline{\mathrm{f}}_j &= \begin{cases} 4\frac{\epsilon}{r_{ij}}\left[\left(12\frac{\sigma}{r_{ij}-\Delta}\right)^{12} - 6\left(\frac{\sigma}{r_{ij}-\Delta}\right)^6\right]\frac{\underline{\mathrm{r}}_{ij}}{r_{ij}-\Delta} & : \quad r_{ij} < 2^{\frac{1}{6}}\sigma + \Delta \\ 0 & : \quad r_{ij} \geq 2^{\frac{1}{6}}\sigma + \Delta \end{cases} \end{aligned}$$

- Standard DPD potential:

$$\begin{aligned} U(r_{ij}) &= \begin{cases} \frac{A}{2}r_{ij}\left(1-\frac{r_{ij}}{r_c}\right)^2 & : \quad r_{ij} < r_c \\ 0 & : \quad r_{ij} \geq r_c \end{cases} \\ \underline{\mathrm{f}}_j &= \begin{cases} \frac{A}{2r_{ij}}\left(1-\frac{r_{ij}}{r_c}\right)\left(3\frac{r_{ij}}{r_c}-1\right) & : \quad r_{ij} < r_c \\ 0 & : \quad r_{ij} \geq r_c \end{cases} \end{aligned}$$

- 14-7 pair potential:

$$
\begin{aligned}
U(r_{ij}) &= \epsilon \left( \frac{1.07}{\left(\frac{r_{ij}}{r_0}\right) + 0.07} \right)^7 \left( \frac{1.12}{\left(\frac{r_{ij}}{r_0}\right)^7 + 0.12} - 2 \right) \\
\underline{\mathrm{f}}_j &= 7\epsilon \left( \frac{1.07}{\left(\frac{r_{ij}}{r_0}\right) + 0.07} \right)^7 \left[ \left( \frac{1.07}{\left(\frac{r_{ij}}{r_0}\right) + 0.07} \right) \left( \frac{1.12}{\left(\frac{r_{ij}}{r_0}\right)^7 + 0.12} - 2 \right) + \frac{1.12}{\left( \left(\frac{r_{ij}}{r_0}\right)^7 + 0.12 \right)^2} \left( \frac{r_{ij}}{r_0} \right)^6 \right] \left( \frac{\underline{\mathrm{r}}_{ij}}{r_{ij} r_0} \right)
\end{aligned}
$$

- : Tabulation: The potential is only defined numerically.

The user defines the potential to be used and the required values of $A$, $B$, $C$, $D$, $E_0$, $k$, $n$, $m$, $r_0$, $r_c$, $\epsilon$, $\sigma$, $\rho$ or $\Delta$ as used within the potential definition.