A Short History of DL_POLY

Bill Smith

Daresbury Laboratory,
Cheshire, England, UK
Parallel Computing at Daresbury

• Late 1980's - Daresbury became centre for parallel computing
• Installed Intel iPSC/2 and iPSC/860 computers
• CSE staff were encouraged to develop parallel programs
• WS started to investigate parallel MD algorithms
Important Collaborations

1989-1992

• Systolic Loop MD Algorithms with D. Fincham and A. Raine (Keele)
• Domain Decomposition MD with M. Pinches and D. Tildesley (Soton)
• Domain Decomposition MD with D. Rapaport (Bar-Ilan)
WS and Parallel MD

• Explored Replicated Data algorithms
  – Methods for parallel force calculations
  – Methods for staging data on/off nodes
  – Methods for stopping/restarting simulations
  – Global sum methods
• Models for performance estimation
• Developed Parallel Ewald sum
• Developed parallel SHAKE algorithm
1991
- Opportunity existed for a new general purpose parallel program for CCP5
- Build on existing parallel know-how, software and algorithms
- Program would have most of the functionality of the program library and open new areas
- A 'flagship' code for CCP5!
- User base: maybe 35 UK groups?
Design Considerations

- General purpose **package**: support a wide range of applications:
  
  Simple atomistic systems, ionic systems, molecular systems, solutions, polymers, biopolymers, liquid crystals...

- Run on serial and parallel computers.

- Insulate user where possible from issues concerning parallel execution.

- Comprehensive error reporting.

- Fully documented and amenable to user modification.
Choice of Algorithm

- Replicated Data*:
  • Fully general purpose, easy to implement, good performance for ~100 processors and ~30,000 atoms, memory hungry.

- Systolic loops:
  • Fully distributed, good for ~100 procs, ~30,000 atoms, difficult to implement for general purpose.

- Domain Decomposition:
  • Fully distributed, excellent for large scale simulations $10^3$ procs, $10^6$ atoms, difficult to implement for general purpose.
- Did not meet all objectives in capability.
- Wide range of potential applications (cf. program library)
- Performance and scaling acceptable.
- Viable platform for extension.
- To accelerate further development, funding sought from SBCC/SMCC.
The DL_POLY Project

1993-1997

- SBCC/SMCC grants to **CCP5** for DL_POLY project. PDRA: Tim Forester (2+2 yrs)
- DL_POLY brought to high specification:
  - Fully documented and methods published
  - Website and distribution procedure established
  - Code released as DL_POLY 1.1 (1993)
  - Scheme developed to process input data including dynamic array allocation
  - Extensive range of electrostatic models
  - 3-,4-body and metal potentials
The DL_POLY Project

• Assorted algorithms for NVT and NPT ensembles
• Multiple time step algorithm
• Rigid body algorithms (+linked rigid bodies)
• Code re-released as DL_POLY 2.x (1994)
• Tools for bio-simulations (force field builder etc.)
• Library of tools for system set-up and results analysis
• TF and WS involved in many scientific collaborations.
The DL_POLY Project

1998-2011

- Smoothed Particle Mesh Ewald (SPME)
- Hautman-Klein (2D) Ewald sum
- DL_POLY Java GUI
- Tersoff Potential
- Hyperdynamics (D. Harris & J. Harding)
- Metadynamics (D. Quigley & P.M. Rodger)
- Solvation tools (PA Cazade, P. Bordat & R. Brown)
The DL_POLY Project

2011
• Re-release as DL_POLY Classic

2016
• Path integral molecular dynamics (PIMD)

DL_POLY Spin-offs
• Multipole electrostatics – DL_MULTI (M. Leslie)
• Force Field Builder – DL_FIELD (Chin Yong)
• Biosimulation – DLPROTEIN (S. Melchionna)
By 1997 it was clear that DL_POLY (RD version) was not a leading code any more, at least in terms of HPC.

Work began on a domain decomposition version: DL_POLY 3

Based on parallel link cells methods of M. Pinches & D. Tildesley and D. Rapaport.
- Must share file structures with DL_POLY.
- Should retain same molecular models (i.e. retain generality of purpose).
- Should scale to large computers \((10^3 \text{ procs})\) and systems \((10^6 \text{ atoms})\).
- Share same philosophy with original DL_POLY regarding user friendliness, documentation, coding standards, ease of use etc.
Algorithm Issues

- How to map the simulated system on a set of processors in a general way?
- Distributed data model I/O implications. (Trivial for small cases but needs advanced solution for large simulations).
- Data must be transferred between processors for system contiguity (solved problem!)
Molecular structure data must be transported when atoms move naturally between domains.

Need to quickly find specific atoms in each domain to evaluate intramolecular forces. (Efficient search or use global data?)

How to calculate Coulombic forces? (Abandon Ewald and use FMM?).
During 1998-1999 workable solutions to these issues were found and put together in a new code: DL_POLY 3.

Some things had to go (some PBCs, HKE and rigid bodies) but the final code could still potentially simulate a wide variety of systems, if not as many as DL_POLY Classic.

Limited testing showed that it worked, but more manpower was needed to go further. Enter eMinerals....
eMinerals

- **Eminerals 1/2** was an NERC eScience project started by Martin Dove at Cambridge. A component of this project was concerned with large scale MD simulations....

- The project provided funding for a PDRA at Daresbury to continue the development of DL_POLY 3.

- Ilian Todorov was recruited into that position in January 2003....
- IT's first task was to get familiar with the existing code and understand its non-trivial parallel algorithms. (Much Q&A ensued!)
- The next stage was to test and debug the code. A number of bugs were found but the basic code design was sound.
- DL_POLY_3.1 was released in March 2003.
- Throughout the life of the eMinerals projects (2002-2008) IT continued to add new features to the program and improve its efficiency.
DL_POLY_4

- DL_POLY 4 first appeared in 2010.
- It is written in syntactically strict Fortran 90 to improve its portability and to provide for future enhanced functionality.
- It is highly optimised and capable of extremely large scale simulations (>10^7 atoms) on massively parallel systems.
- Particular features worth mentioning:
  • State of the art parallel SPME method
  • The parallel I/O staging algorithm
  • Point multipole Ewald sum
Thermostats and the Equipartition Principle

Bill Smith and Alin Elena

Daresbury Laboratory,
Cheshire, England, UK
Background

- Daniel Dervin contact Alin Elena ~ Jan 2017.
- Simulations of a system composed of a silicate framework and encapsulated aromatic molecules behaved anomalously.
- The silicate framework froze as simulation proceeded, despite Nose'-Hoover thermostat.
- Studies showed that equipartition principle did not always reliably occur in other silicate systems.
- Why does this happen? How can it be easily identified? What to do about it when it is?
User Query (D. Dervin 2017)
Why Equipartition Fails

- It has long been known that systems based on harmonic potentials have difficulty achieving equipartition.
- Exchange of energy between degrees of freedom is inefficient.
- The silicate systems we looked at were rich in harmonic components, including harmonic bonds, angles and dihedrals.
- In harmonic systems anomalous behaviour is not necessarily readily apparent - but it may be there nevertheless!
Checking Equipartition

\[ M_u^n = \left( \frac{1}{2 \pi \sigma^2} \right)^{1/2} \int_{-\infty}^{\infty} u^n \exp \left( -\frac{u^2}{2 \sigma^2} \right) \, du. \]

\[ \sigma^2 = k_B T/m \]

\[ M_u^n \equiv \langle u^n \rangle \]

\[ M_u^{2m} = \sigma^{2m} (2m - 1)!! \]

\[ \langle u^2 \rangle = \sigma^2 \]

\[ \langle u^4 \rangle = 3 \sigma^4 \]

\[ \langle u^6 \rangle = 15 \sigma^6 \]

\[ \langle u^8 \rangle = 105 \sigma^8 \]

\[ \langle u^{10} \rangle = 945 \sigma^{10} \]

\[ f(u) = \left( \frac{m}{2 \pi k_B T} \right)^{1/2} \exp \left( -\frac{mu^2}{2 k_B T} \right) \]
Ensuring Equipartition

- Nose'-Hoover thermostat:

\[ \dot{r}_i = \frac{\dot{p}_i}{m_i} \quad \text{i=1,...,N} \]

\[ \dot{p}_i = f_i - \dot{p}_i \xi \]

\[ \dot{\xi} = \frac{1}{Q} \left( \sum_{i=1}^{N} \frac{p_i^2}{m_i} - f k_B T \right) \]
Ensuring Equipartition

- Nose'-Hoover chains:

\[
\dot{r}_i = \frac{\dot{p}_i}{m_i} \\
\dot{p}_i = f_i - \dot{p}_i \xi_1 \\
\xi_1 = \frac{1}{Q_1} \left( \sum_{i=1}^{N} \frac{p_i^2}{m_i} - f k_B T \right) - \xi_1 \xi_2 \\
\xi_m = \frac{1}{Q_m} \left( Q_{m-1} \xi_{m-1}^2 - k_B T \right) - \xi_m \xi_{m+1} \\
\xi_M = \frac{1}{Q_M} \left( Q_{M-1} \xi_{M-1}^2 - k_B T \right) \quad \text{for} \quad i=1, \ldots, N \quad \text{and} \quad m=2, \ldots, M-1
Ensuring Equipartition

- Multiple Nose'-Hoover chains:

\[
\begin{align*}
\dot{r}^\alpha_i &= \frac{p^\alpha_i}{m_i} \\
\dot{p}^\alpha_i &= f^\alpha_i - p^\alpha_i \xi^\alpha_{i,1} \\
\dot{\xi}^\alpha_{i,1} &= \frac{1}{Q_{i,1}} \left( \frac{(p^\alpha_i)^2}{m_i} - k_B T \right) - \xi^\alpha_{i,1} \xi^\alpha_{i,2} \\
\dot{\xi}^\alpha_{i,m} &= \frac{1}{Q_{i,m}} \left( Q_{i,m-1} (\xi^\alpha_{i,m-1})^2 - k_B T \right) - \xi^\alpha_{i,m} \xi^\alpha_{i,m+1} \\
\dot{\xi}^\alpha_{i,M} &= \frac{1}{Q_{i,M}} \left( Q_{i,M-1} (\xi^\alpha_{i,M-1})^2 - k_B T \right)
\end{align*}
\]

\(i = 1, \ldots, N \quad \alpha = x, y, z\)
• Alternative – Gentle Thermostat (Leimkuhler):

\[
\begin{align*}
\dot{r}_i^\alpha &= \frac{p_i^\alpha}{m_i} \\
\dot{p}_i^\alpha &= f_i^\alpha - p_i^\alpha \xi_i^\alpha \\
\dot{\xi}_i^\alpha &= \frac{1}{Q_{i,1}} \left( \left( \frac{p_i^\alpha}{m_i} \right)^2 - k_B T \right) + w \sigma - \lambda \xi_i^\alpha
\end{align*}
\]

\(\sigma\) - is a Gaussian random number
\(w\) - is a (small) weighting factor
\(\lambda\) - is drag coefficient
The Dragon MD Code

• Experimental code for investigating equipartition issues.
• Based on DL_POLY Classic (stripped-down – no rigid bodies, no constraints, NVT only).
• Incorporates $2n$ moment Gaussian parameters for particle velocity distribution.
• Nose'-Hoover, N-H chain, Leimkuhler, single and multiple thermostats.
• Available from Daresbury.
The End