Science and
Technology
Facilities Council

Scientific Computing

Welcome

Science and Technology Facilities Council

Scientific Computing

DL SOFTWARE

Enabling Discovery

DL_Software: Atomistic to Mesoscopic Scale Modelling & Simulation: DL_POLY, DL_MESO_DPD, DL_FIELD, D_ATA, SHAPESPYER, DL_MESO_LBE

14-16 May @ University of Manchester

Ilian Todorov, Michael Seaton, Benjamine Speake, Joseph Thacker, Harvey Devereux, Chin Yong, Andrey Brukhno & Valeria Losasso

HEC MCC UK

UK COMES

Consortium on Mesoscale Engineering Sciences

CCP5

Collaborative Computational Project

Computer Simulation of Condensed Phases

40th Anniversary

CoSeC

Computational Science Centre for Research Communities

Flexible Agenda over three days

1 Introduction to Molecular Dynamics

What does it do and what it does not?

2 Introduction to DL_POLY

Purpose, history, capability, use, etc.

3 Introduction to DPD and DL_MESO

Why and how is DPD different from MD. Purpose of Coarse Graining.

4 Calculations on the Fly

Why do we need calculations on the fly.

5 What's up doc

Demonstration, labs and helping your research projects by using DL_POLY and the rest of the packages

Science and
Technology
Facilities Council

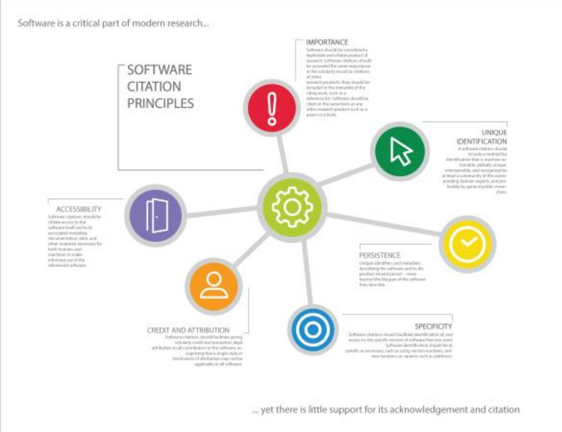
Scientific Computing

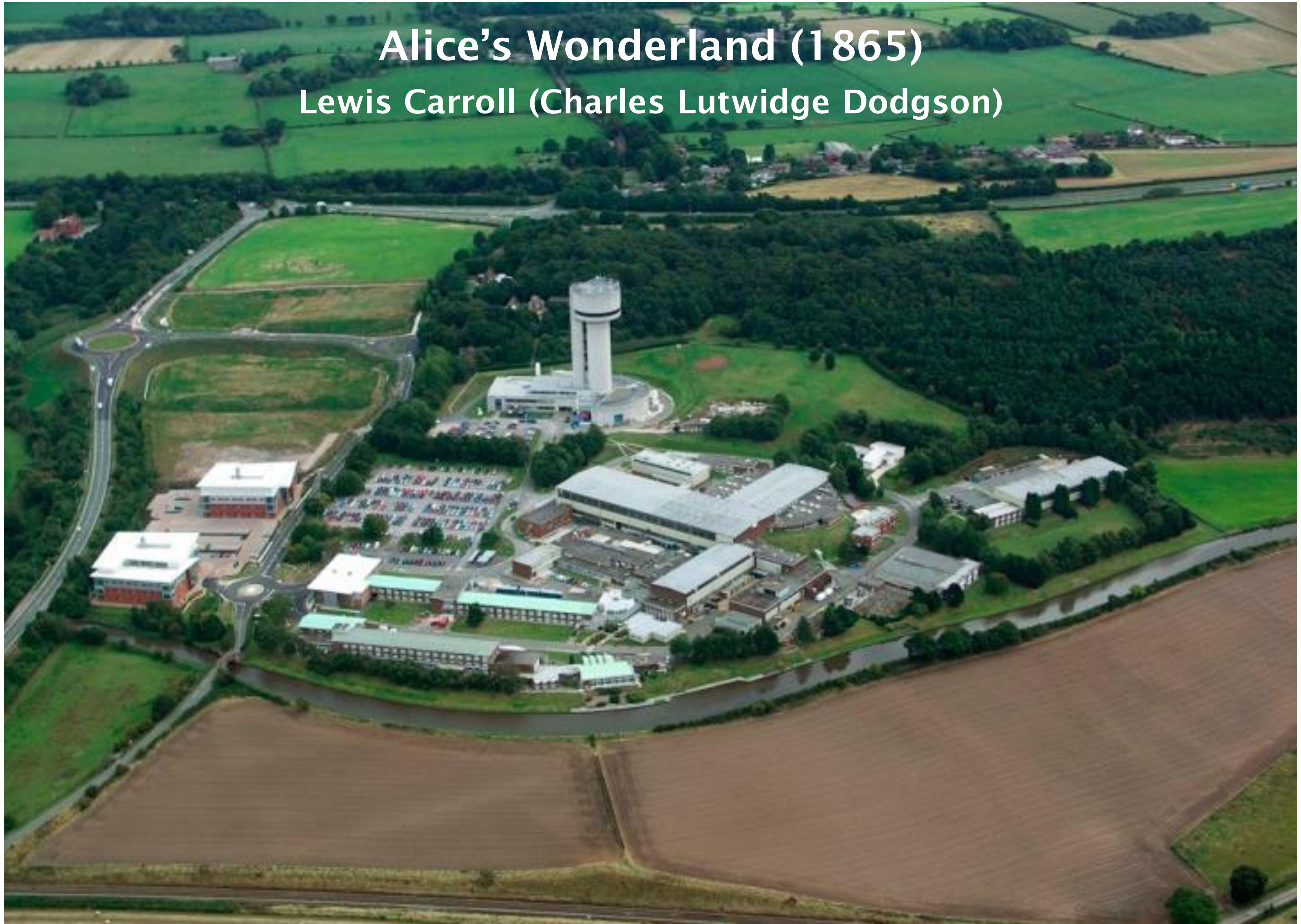
Daresbury Laboratory

RSE
SOCIETY OF RESEARCH SOFTWARE ENGINEERING

FORCE11
The Future of Research Communications and e-Scholarship

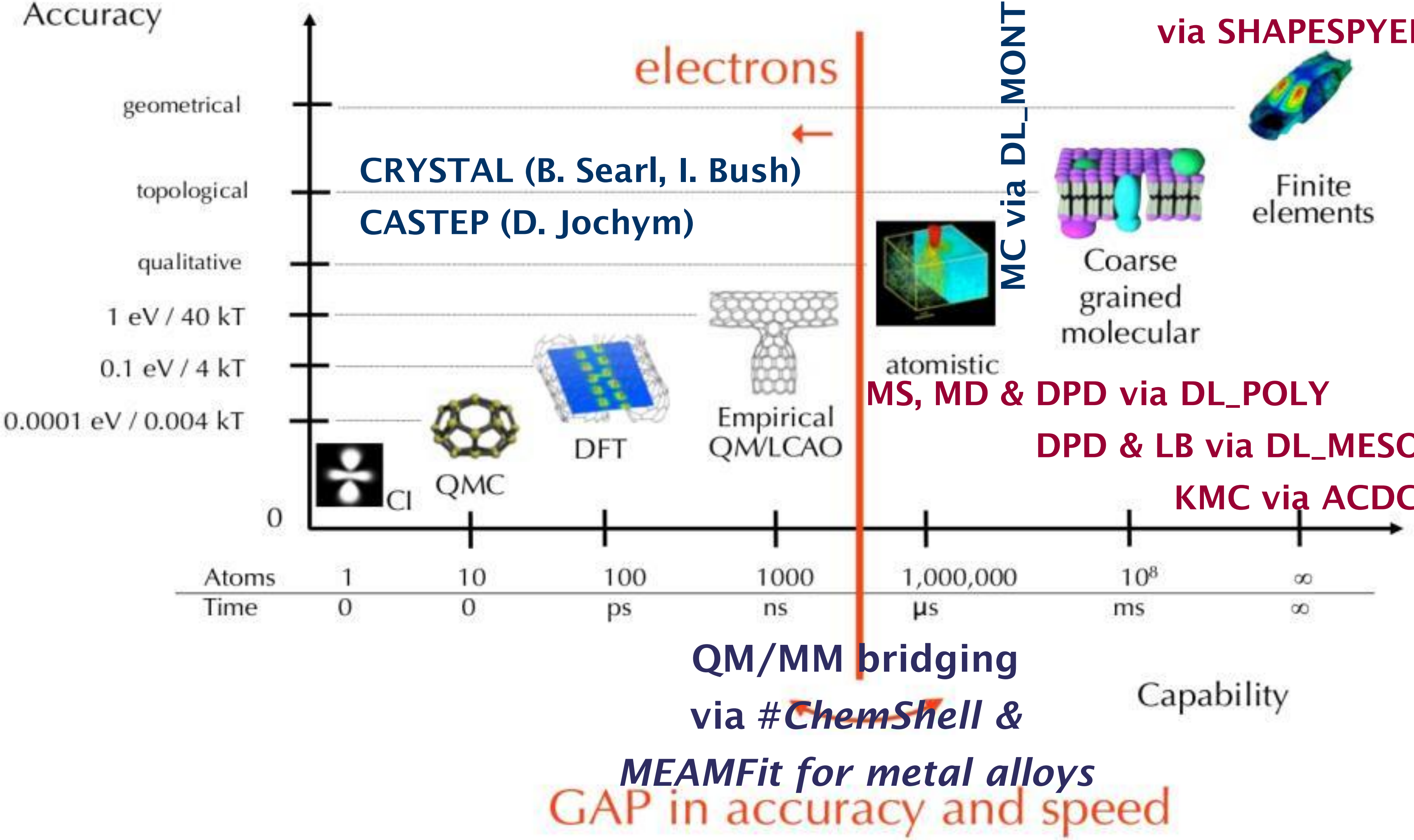
Software Sustainability Institute




Alice’s Wonderland (1865)
Lewis Carroll (Charles Lutwidge Dodgson)

UKRI
Science and Technology Facilities Council
Scientific Computing

Scales of Materials Modelling



FF mapping
via DL_FIELD

Coarse graining and soft-condensed matter workflows
via SHAPESPYER

MC via DL_MONTE

CRYSTAL (B. Searl, I. Bush)
CASTEP (D. Jochym)

MS, MD & DPD via DL_POLY
DPD & LB via DL_MESO
KMC via ACDC

QM/MM bridging
via *#ChemShell &*
MEAMFit for metal alloys

Molecular Simulation Ecosystem

- DL_POLY – particle Molecular Dynamics
- dlpoly-py framework – Alin Elena
- DL_MONTE – atomistic Monte Carlo including GCMC – Andrey Brukhno
- ACDC – Adaptive kinetic Monte Carlo – John Purton
- pyChemShell – collaborative framework & QM/MM coupling environment
  - DL-FIND – configuration minimisation library - Thomas Keal, You Lu
- DL_MESO – mesoscale dynamics via Dissipative Particle Dynamics and Lattice Boltzmann Equation – Michael Seaton
- DL_FIELD – advanced force-field matching, conversion & set-up for soft-matter – Chin Yong – DL_ANALYSER/D_ATA – advanced analytics for soft-matter simulations
- SHAPESPYER – a workflow and simulation builder for condensed soft-mater systems (atomistic & coarse-grained) – Andrey Brukhno
- MEAMfit – potential optimisation & high temperature *ab initio* MD – Andrew Duff

Science and
Technology
Facilities Council

Scientific Computing

Introduction to particle dynamics

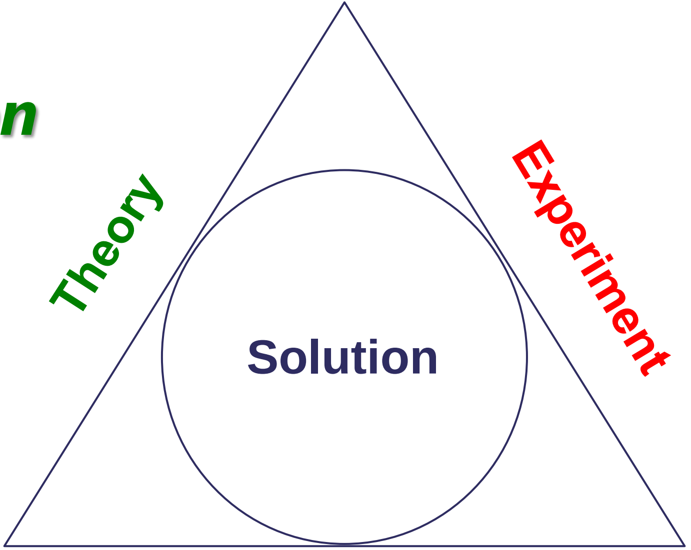
What is particle dynamics?

Newton's Laws of Motion

force = **m**ass * **a**cceleration

t = **t**ime
v = **v**elocity
r = position

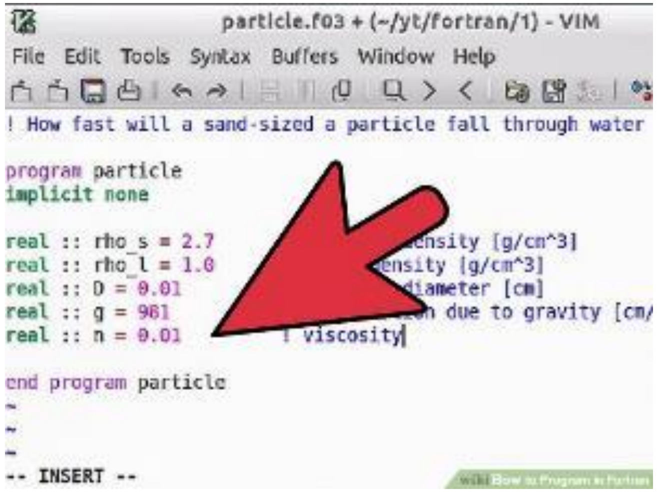
$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v} * \mathbf{t} + \mathbf{f} * \mathbf{t}^2 / \mathbf{m}$$


Theory
Experiment
Solution

Computation

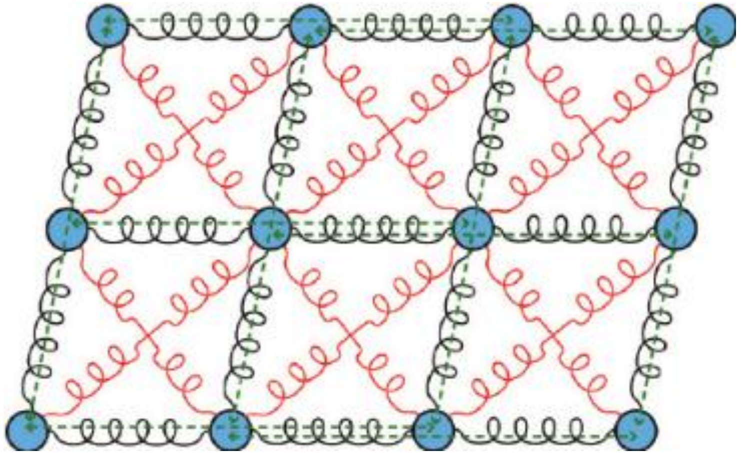
Instruments

Software (maths)



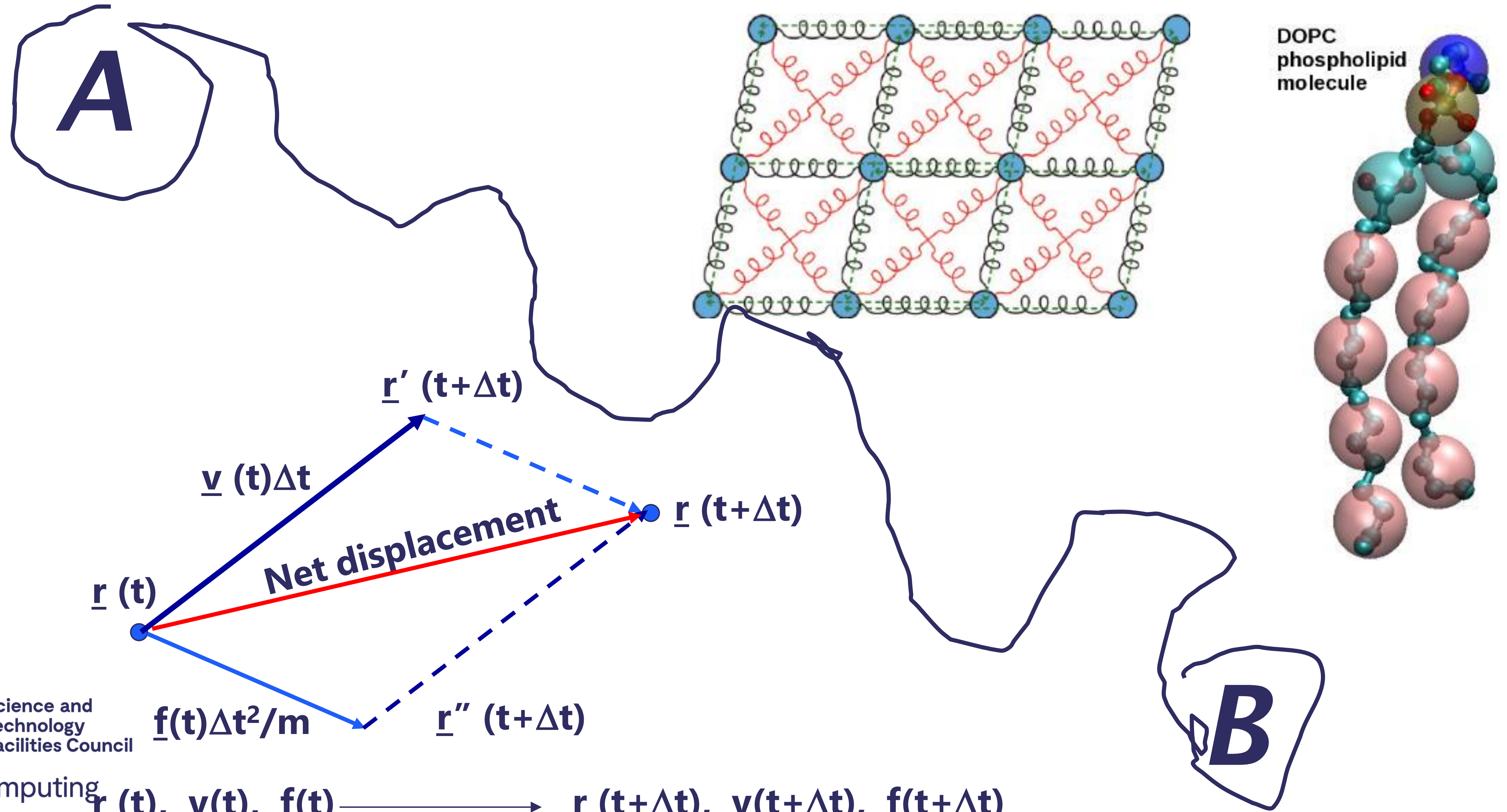
```
! How fast will a sand-sized a particle fall through water

program particle
implicit none

real :: rho_s = 2.7            ensity [g/cm^3]
real :: rho_l = 1.0           ensity [g/cm^3]
real :: D = 0.01               diameter [cm]
real :: g = 981                 n due to gravity [cm/
real :: n = 0.01            ! viscosity

end program particle
~
~
~
-- INSERT --
```

Model (forces)



Computer (crunch)

Particle dynamics is a journey

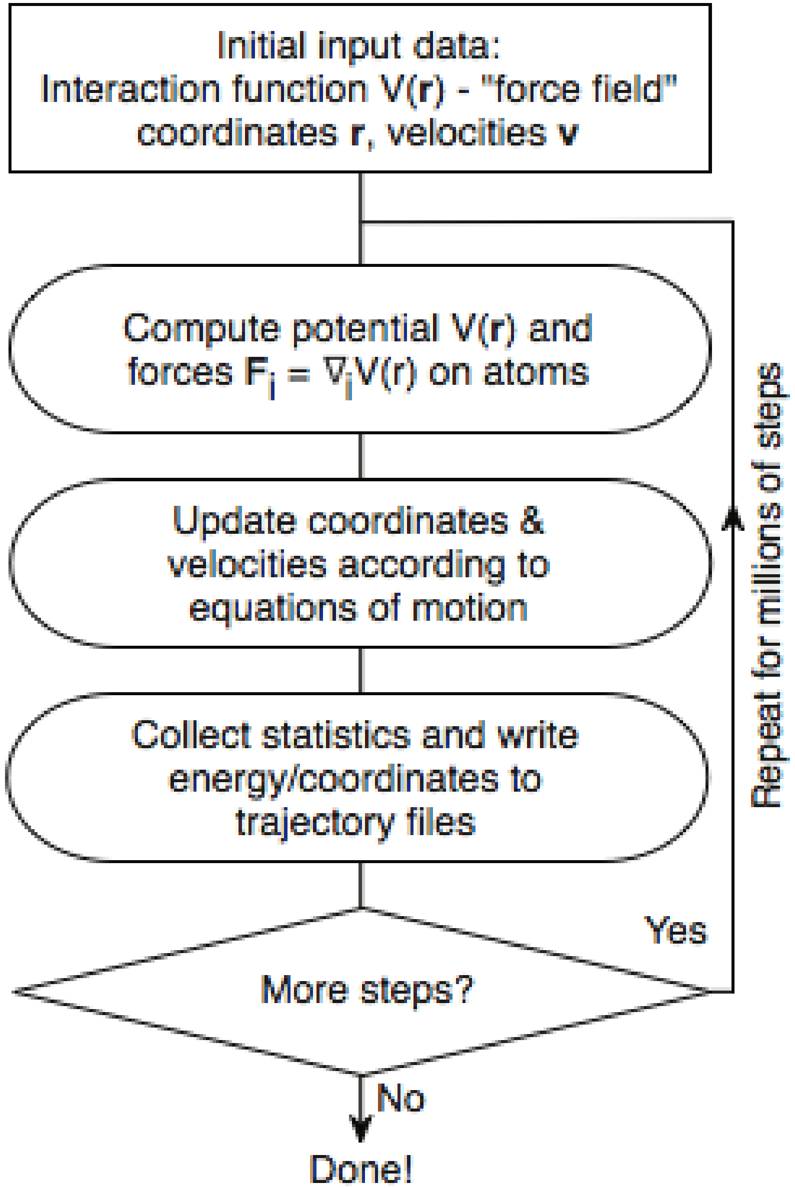


$\underline{r}\ (t),\ \underline{v}(t),\ \underline{f}(t) \longrightarrow \underline{r}\ (t+\Delta t),\ \underline{v}(t+\Delta t),\ \underline{f}(t+\Delta t)$

Why use MD?

- Theoretical tool for modelling the detailed microscopic behaviour of many different types of systems, including; gases, liquids, solids, polymers, surfaces and clusters.
- In an MD simulation, the classical equations of motion governing the microscopic time evolution of a many body system are solved numerically, subject to the boundary conditions appropriate for the geometry or symmetry of the system.
- Can be used to monitor the microscopic mechanisms of energy and mass transfer in chemical processes, and dynamical properties such as absorption spectra, rate constants and transport properties can be calculated.
  - Can be employed as a means of sampling from a statistical mechanical ensemble and determining equilibrium properties. These properties include average thermodynamic quantities (pressure, volume, temperature, etc.), structure, and free energies along reaction paths.

What is MD in a nutshell



- MD is the solution of the classical equations of motion for atoms and molecules to obtain the time evolution of a system.

- It is applied to many-particle systems since a general analytical solution is not possible. Thus one must resort to numerical methods and computers.

- It does classical mechanics only since a fully fledged many-particle time-dependent quantum method is not yet available.

- It uses a Maxwell-Boltzmann averaging process for thermodynamic properties (i.e. time averaging).

Use of MD

Ensemble average:

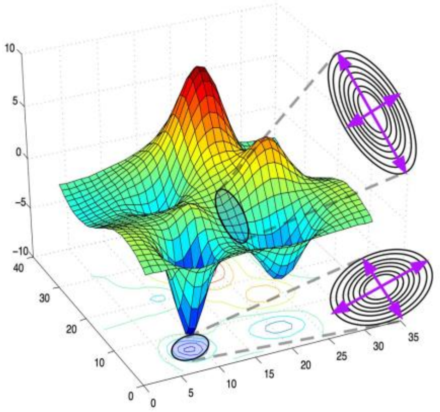
- Free energy (of binding, solvation, interaction) differences
- Diffusion coefficients, viscosity, elastic constants
- Reaction rates, phase transition properties
- Protein folding times
- Structure refinement

Non-equilibrium processes:

- Energy Dissipation/Radiation damage
- Sound Propagation
- Surface coating
- Nucleation (meta-dynamics)

Some properties can be obtained directly from neutron scattering

Limitations of MD

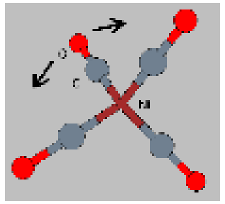


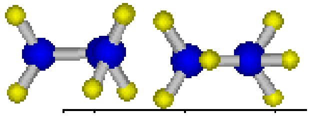
- Parameters are imperfect and fit to particular P, T, pH, etc.
- Phase space is not sampled exhaustively
- Example: Free energies of solvation for amino acids often have errors ~1kJ/mol
- Likely impossible to calculate binding free energies more accurately than this
- Chemical bonds breaking and creation is not allowed
- Limited polarization effects; waters can reorient, but partial charges are fixed

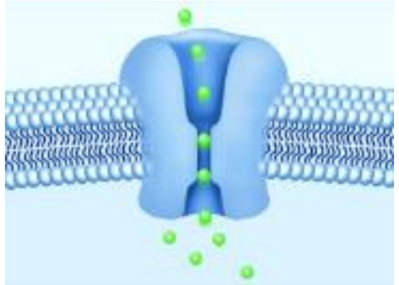
However, MD simulations are cheaper than experiments as more easy to set up, repeat with changes to the model systems, their force-field, the initial conditions and simulation control; and thus serve as an invaluable testing tool for scientists! It may be used cleverly to answer cheaply hypothetical and comparative questions!

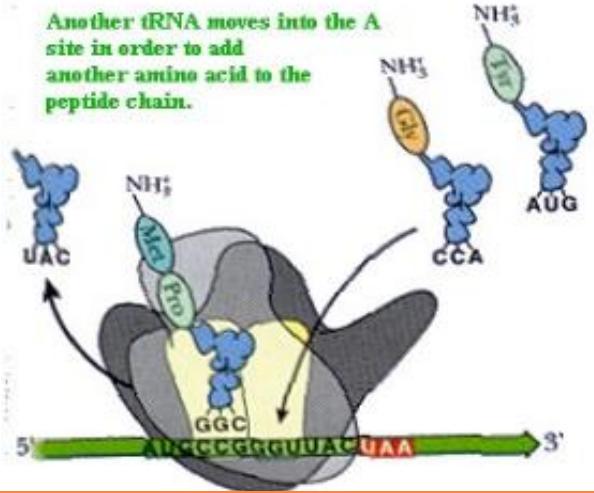
It can be made DFT level exact and embed reactivity with the help of MLIPs. However, it comes at a cost!

Time and Length Scales







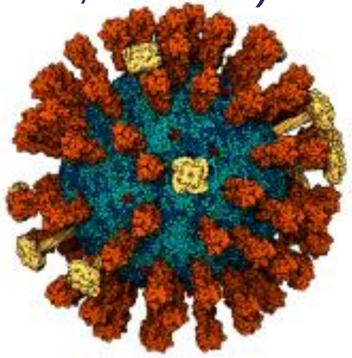


Biological Experiments

10^{-15}s 10^{-12}s 10^{-9}s 10^{-6}s 10^{-3}s 10^{0}s 10^{3}s

Coarse-grained models

(Whole proteins, viruses)



Molecular dynamics

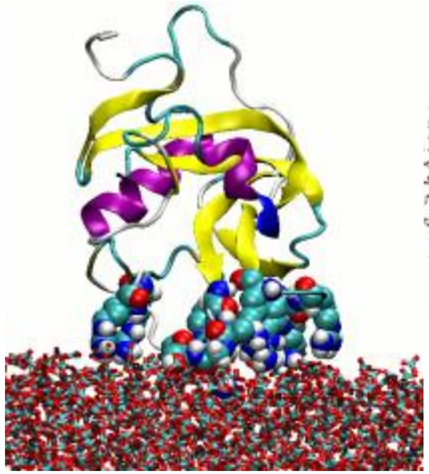
(Atomic detail)

QM simulations

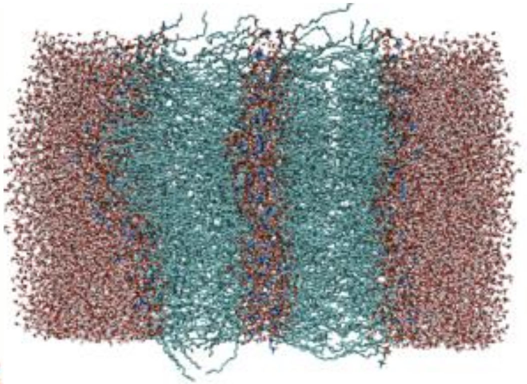
(Electrons)

(MD is fast!)

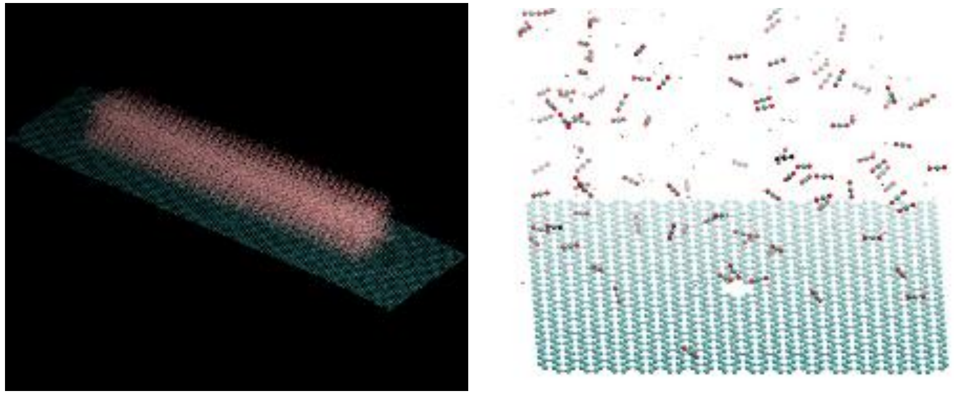
Examples of Model Systems



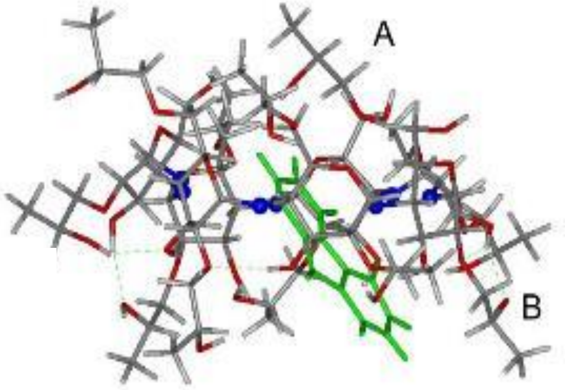
Proteins
solvation & binding



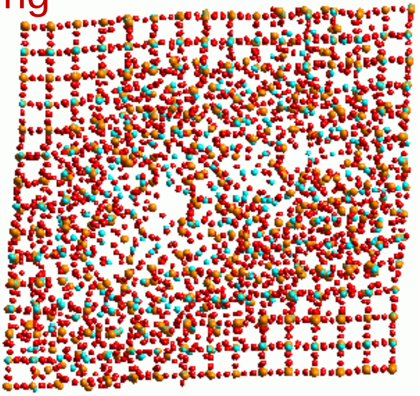
Membranes’ processes



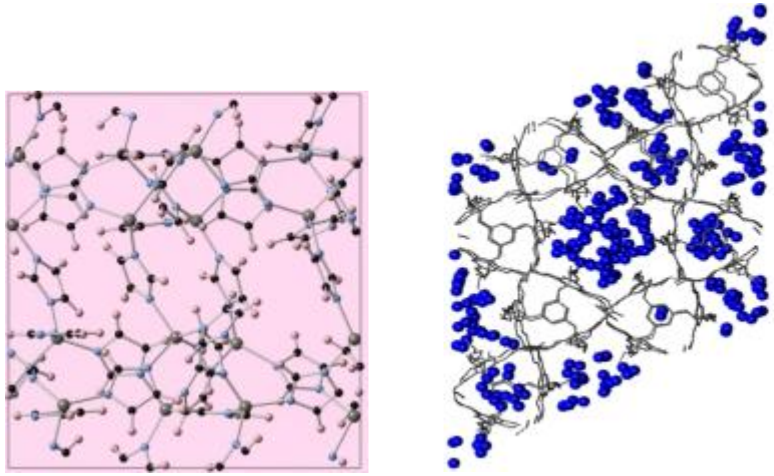
Nanocomposite materials involving graphene



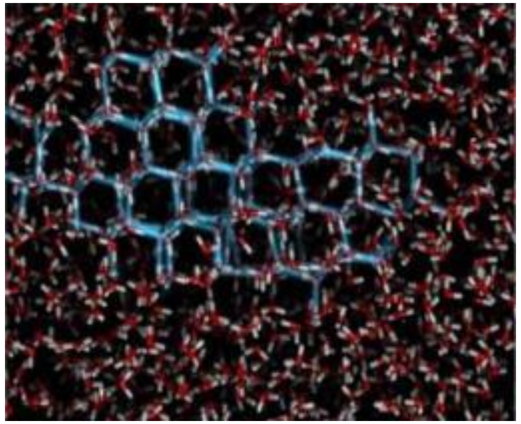
Carbohydrates and
complex drug molecules



Crystalline & Amorphous
Solids – damage and recovery



Dynamic processes in
Metal-Organic & Organic Frameworks

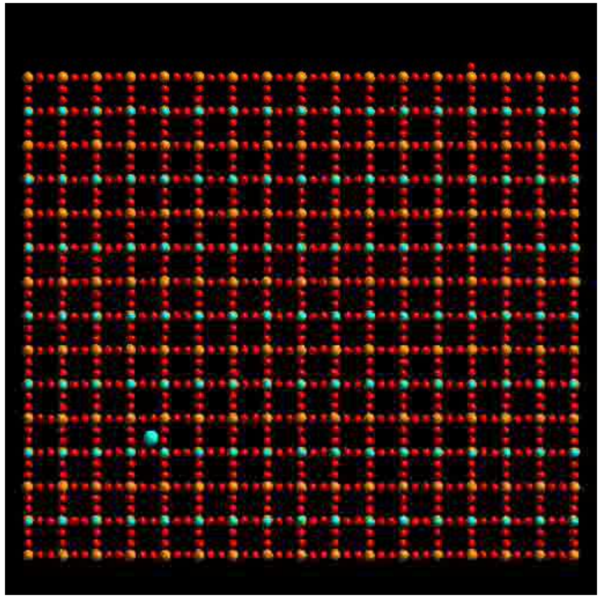


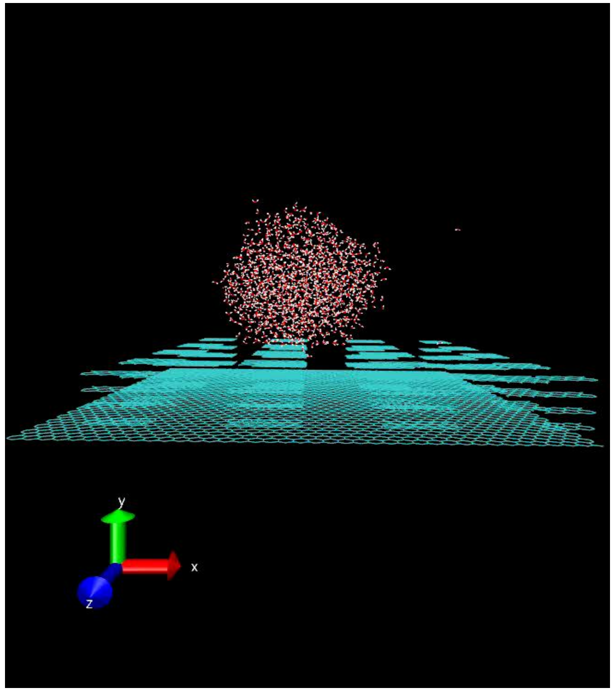
Dynamics at Interfaces &
of Phase Transformations

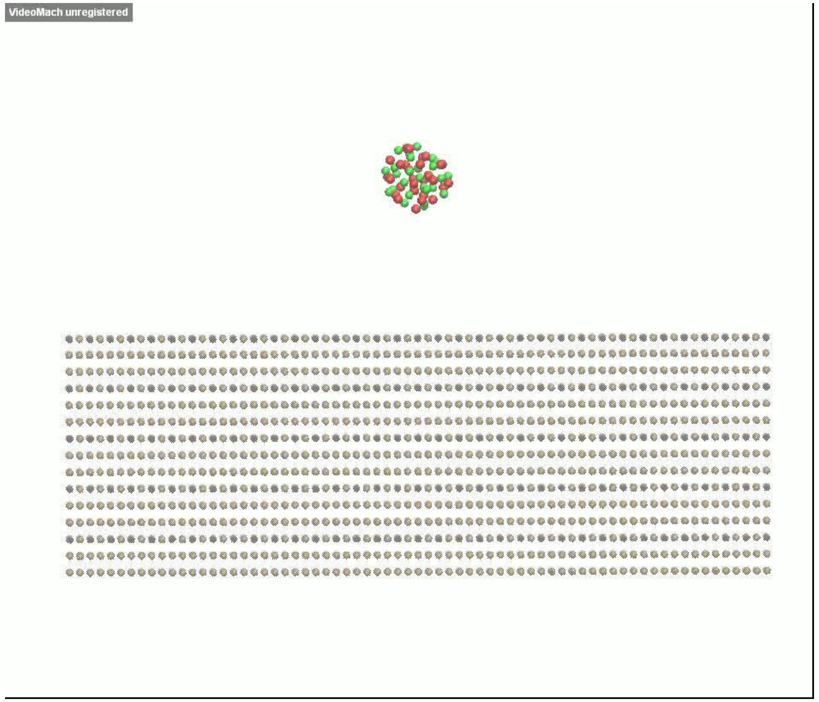
Science and
Technology
Facilities Council

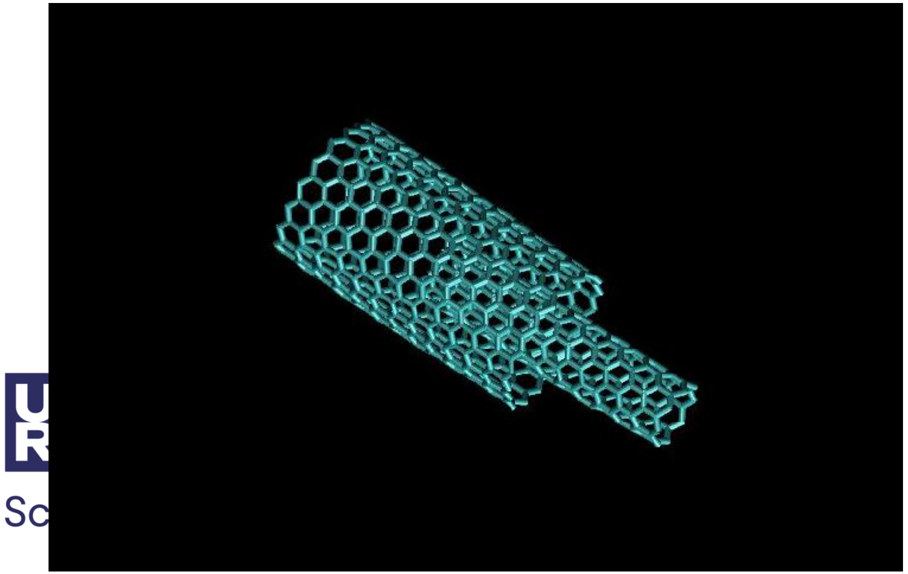
Scientific Computing

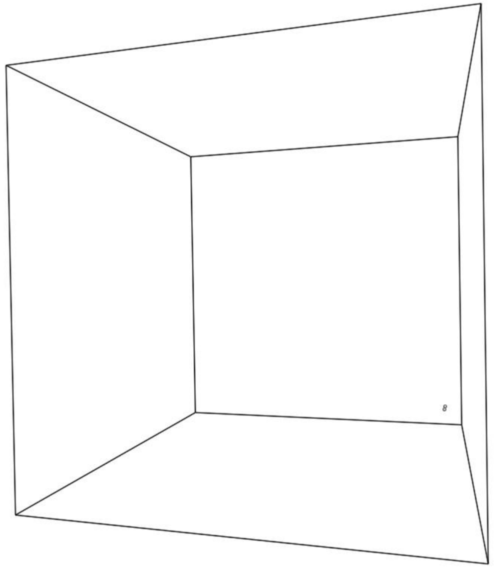
Capability Videos

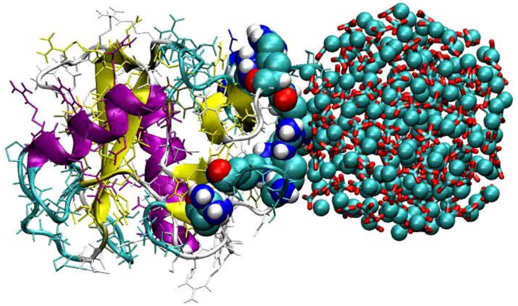












Further information

- More details on Molecular Simulations in general – including theoretical background and applications – are available in the Knowledge Centre of the DL_Software Digital Guide (DL_SDG):

https://dl-sdg.github.io/

- DL_POLY repository

https://gitlab.com/ccp5/dl-poly/

https://github.com/ccp5UK/dl-poly/

- DL_POLY community list

http:/www.jiscmail.ac.uk/DLPOLY/

Science and
Technology
Facilities Council

Scientific Computing

Molecular Dynamics for Beginners

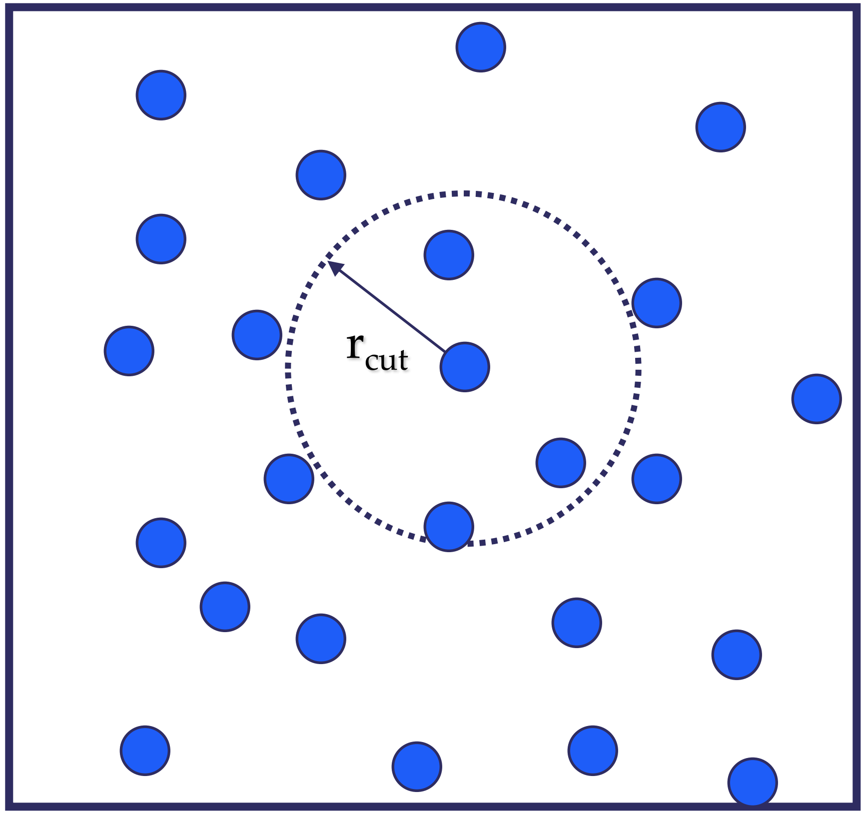
Science and
Technology
Facilities Council

Scientific Computing

Example Use of MD

- Microscopic insight: we can follow the motion of a single molecule (glass of water)
- Investigation of phase change (NaCl)
- Understanding of complex systems like polymers (plastics – hydrophilic and hydrophobic behaviour)

Example: Simulation of Argon



Pair Potential:

$$V(r_{ij}) = 4\varepsilon\left\{\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right\}$$

Lagrangian:

$$L = K(\dot{\vec{v}}) - U(\vec{r})$$

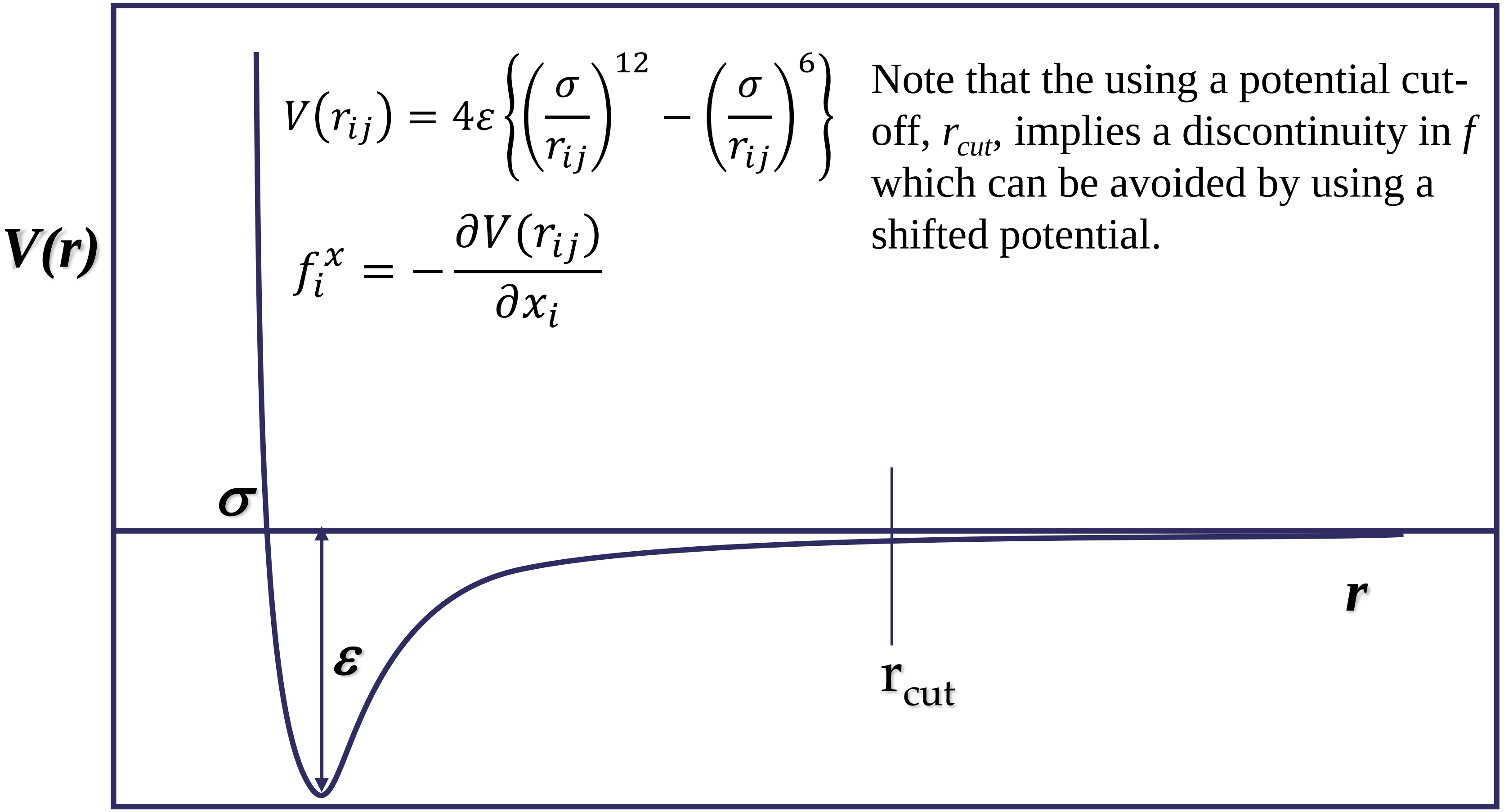
$$L(\vec{r_i},\vec{v_i}) = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 - \sum_{i=1}^{N-1}\sum_{j<i}^{N} V(r_{ij})$$

Science and
Technology
Facilities Council

Scientific Computing

Lennard-Jones Potential

Models the Pauli exclusion principle (repulsive) at short distances & the van der Waals forces (attractive) at long ones



$$V(r_{ij}) = 4\varepsilon\left\{\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right\}$$

$$f_i^x = -\frac{\partial V(r_{ij})}{\partial x_i}$$

Note that the using a potential cut-off, r_{cut}, implies a discontinuity in f which can be avoided by using a shifted potential.

Pair-wise radial distance

Science and Technology Facilities Council

Scientific Computing

Equations of Motion

Lagrange Equation – time evolution

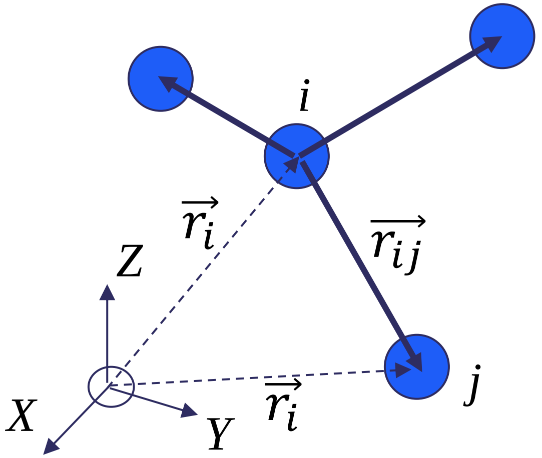
$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{v}_i}\right) = \frac{\partial L}{\partial r_i}$$

$$m_i \vec{a}_i = \vec{F}_i$$

$$\vec{F}_i = \sum_{i=1}^{N} \vec{f}_{ij}$$

$$\vec{f}_{ij} = -\vec{\nabla}_i V(r_{ij})$$

Force Evaluation – particle interactions



Pair force:

$$f_{i(j)}^{x} = -\frac{\partial V(r_{ij})}{\partial x_i} = -\frac{\partial V(r_{ij})}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial x_i}$$

$$\overrightarrow{r_{ij}} = \overrightarrow{r_j} - \overrightarrow{r_i}$$

$$r_{ij} = \left[(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2\right]^{\frac{1}{2}}$$

$$\frac{\partial r_{ij}}{\partial x_i} = -\frac{(x_j - x_i)}{r_{ij}} = \frac{r_{ij}^{x}}{r_{ij}}$$

Note this leads to equal and opposite forces on the two particles.

EoM Consequences

$$f_i = m_i \frac{\partial v_i}{\partial t}$$

The force on atom *i* for any given configuration can be calculated from the force-field. This equation relates force and acceleration.

$$\frac{\partial v_i}{\partial t} = \frac{f_i}{m_i}$$

So for any given configuration we know the acceleration of each particle.

$$v_i(\tau) = v_i(0) + \int_0^{\tau} \frac{dv_i}{dt} dt$$

If a particle has an initial velocity, $v_i(0)$, and moves under the action of this force for a time, τ, its velocity after the time, τ, will be given by integration.

$$r_i(\tau) = r_i(0) + \int_0^{\tau} v_i(t) dt$$

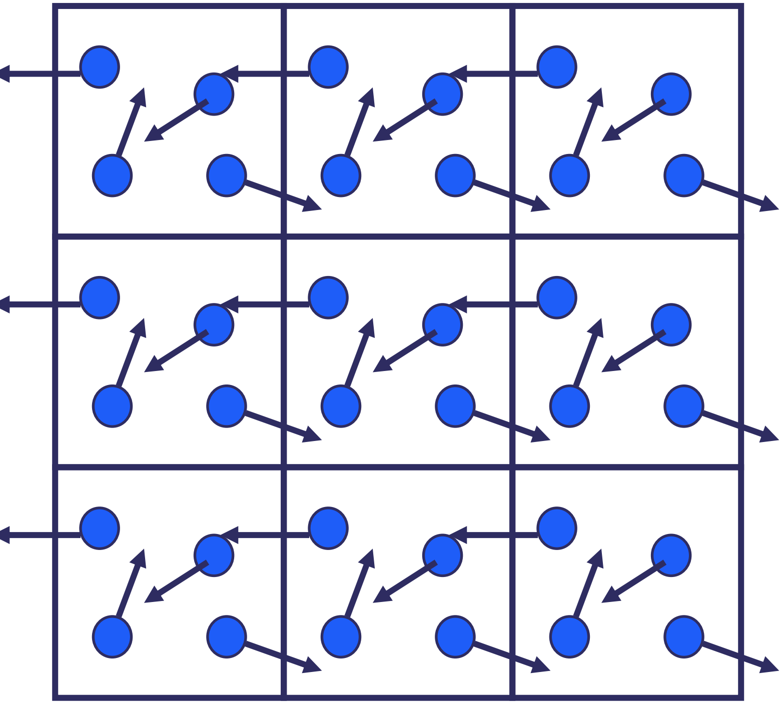
Similarly, the position of the particle after time, τ, is given by an integral of the velocity.

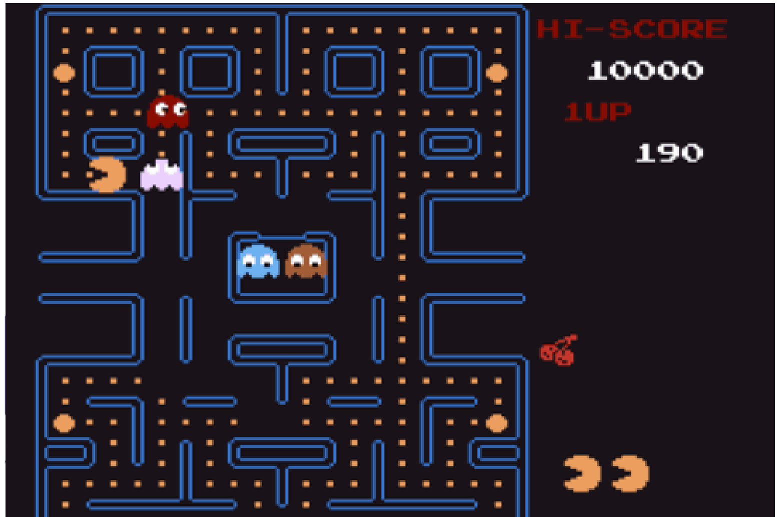
Unfortunately, once the particles move the distances governing the potential change and so the forces are altered.

Molecular dynamics is about integrating these equations of motion such that the continuous trajectories are obtained numerically.

Boundary Conditions

2D cubic periodic





- None – biopolymer simulations
- Stochastic boundaries – biopolymers
- Hard wall boundaries – pores, capillaries
- Periodic boundaries (PBC) – most MD simulations

Why PBC

Our model systems are still too small especially with respect to Avogadro's number!

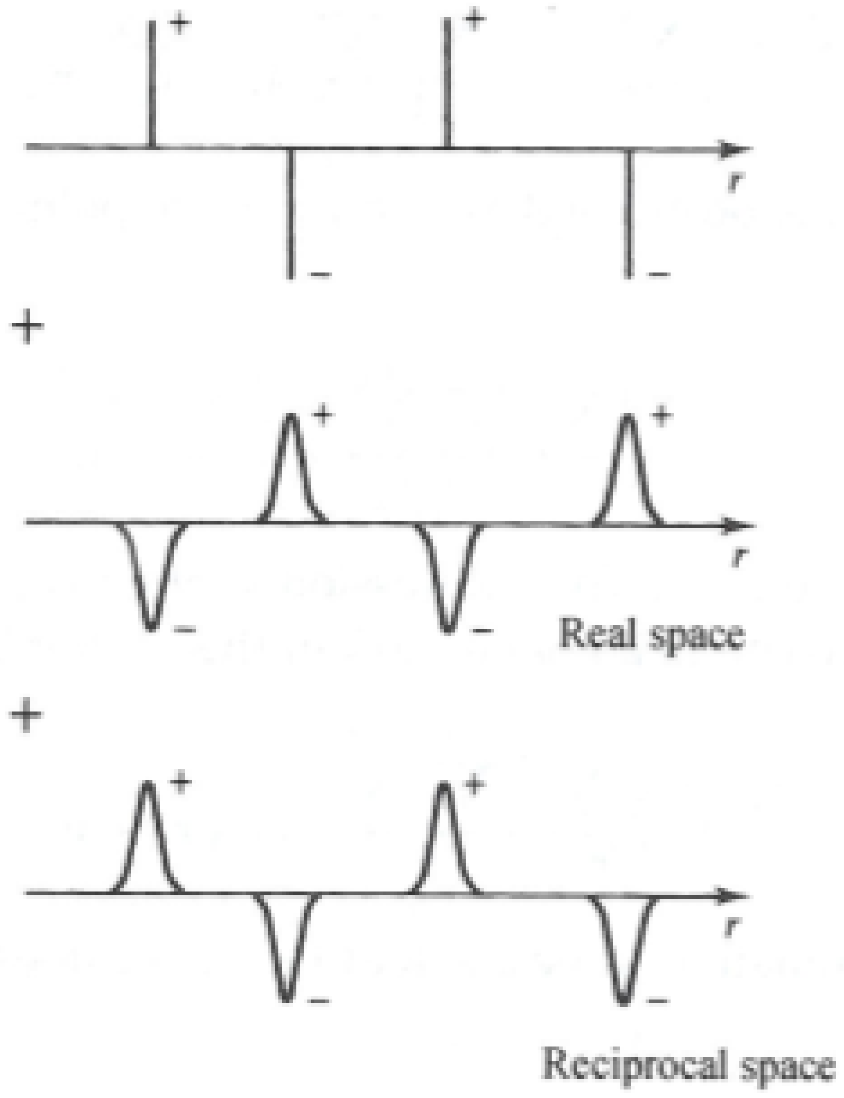
To avoid surface over bulk domination effects we resort to periodic boundaries, pretending that boundaries do not exist (like in pacman)!

PBC Consequences

- The system no longer has a *surface*.
- The system becomes *pseudo-periodic* (used to advantage for Ewald sums). It is incorrect to impose cutoff on long range interactions. However, the Ewald method can deal with infinite number of periodic images – specifically Particle Mesh Ewald (PME)
- Correlations in space beyond *half-cell width* (L/2) are artificial. For this reason, the cut-off r_{cut} is usually no greater than L/2.
- Correlations in time beyond t=L/c are (in principle) subject to *recurrence*. In practice this does not seem to be the case.
- Use with *Minimum Image* convention

Ewald Summation

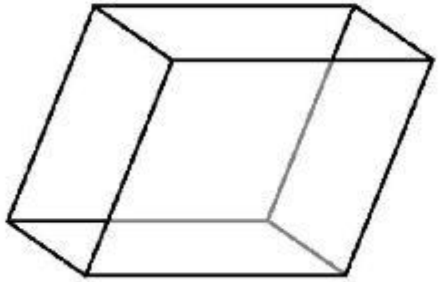
The method offers an elegant solution to solving the full electrostatic problem by splitting it in two parts – one in *real space* and one in *reciprocal space*. In real space, complying with the cutoff concept, a convenient screening function is added around all charges to make their interactions decay very fast at r_{cut}. The added screening functions can be subtracted in reciprocal space due to the periodic boundary condition by using *Fourier transforms.*



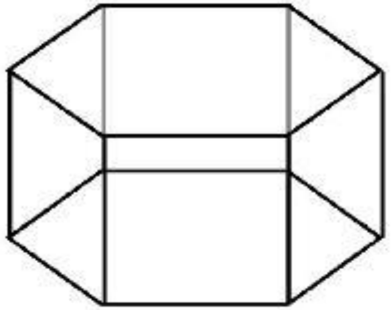
Science and Technology Facilities Council

Scientific Computing

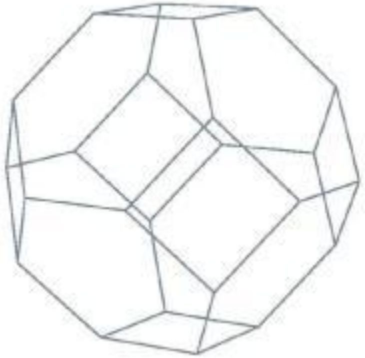
Periodic Boundary Conditions



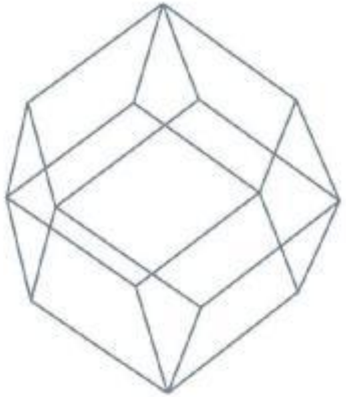
Triclinic



Hexagonal prism



Truncated octahedron

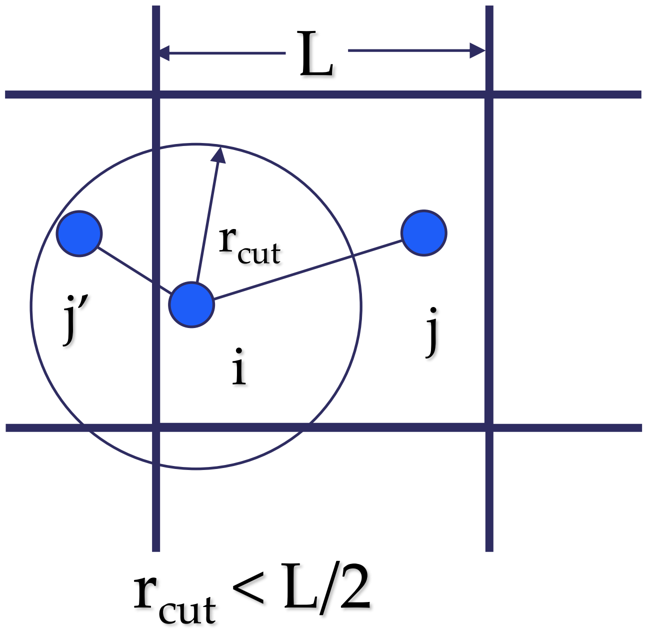


Rhombic dodecahedron

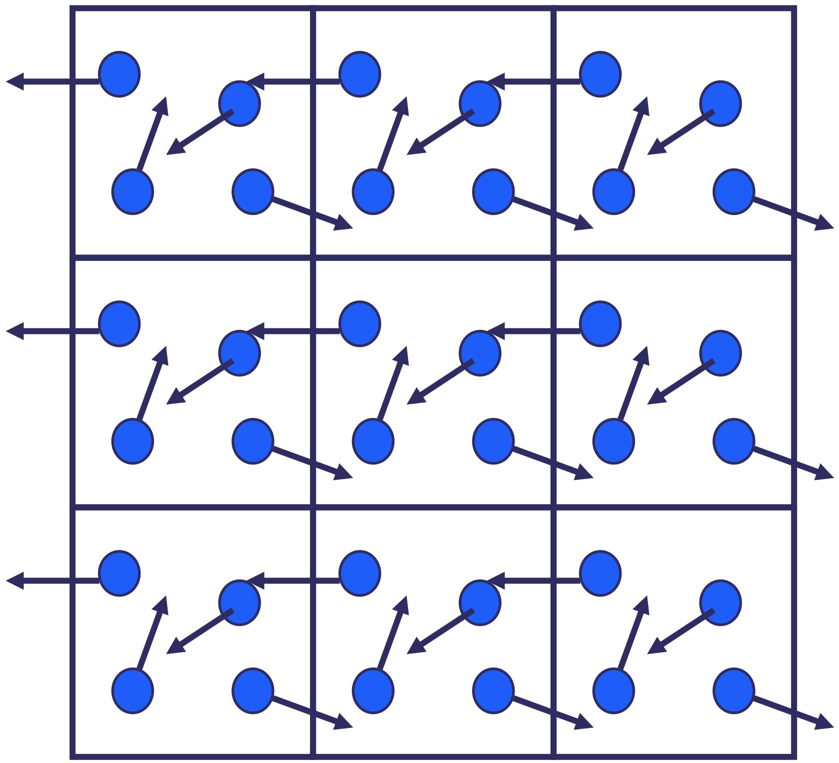
Science and
Technology
Facilities Council

Scientific Computing

Minimum Image Condition (MIC)



2D cubic periodic



Use $r_{ij'}$ not r_{ij}

$x_{ij} = x_{ij} - L^* \, \mathrm{Nint}(x_{ij}/L)$

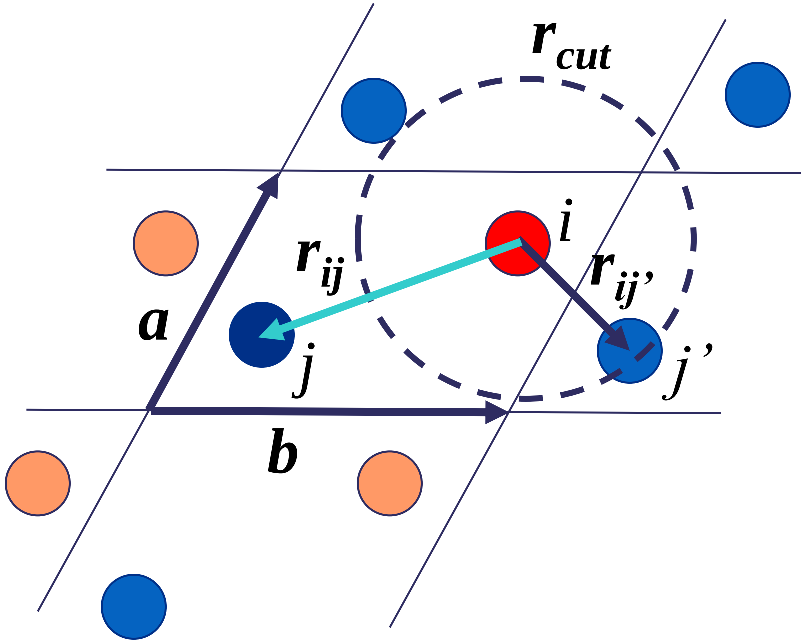
Nint(***a***)=nearest integer to ***a***

3D MIC

Use $\boldsymbol{r}_{ij'}$, not $\boldsymbol{r}_{ij}$: To find $\boldsymbol{r}_{ij'}$, work in fractional co-ordinates:

$$f_{aij} = a^{*} \bullet r_{ij} \qquad f_{bij} = b^{*} \bullet r_{ij} \qquad f_{aij} = c^{*} \bullet r_{ij}$$

(* indicates reciprocal space vectors)



These need to be in the range $-½ < f \leq ½$

$$f_{ij'} = f_{ij} - \mathrm{n\,int}\left(f_{ij}\right)$$

(nint(f)=nearest integer to f)

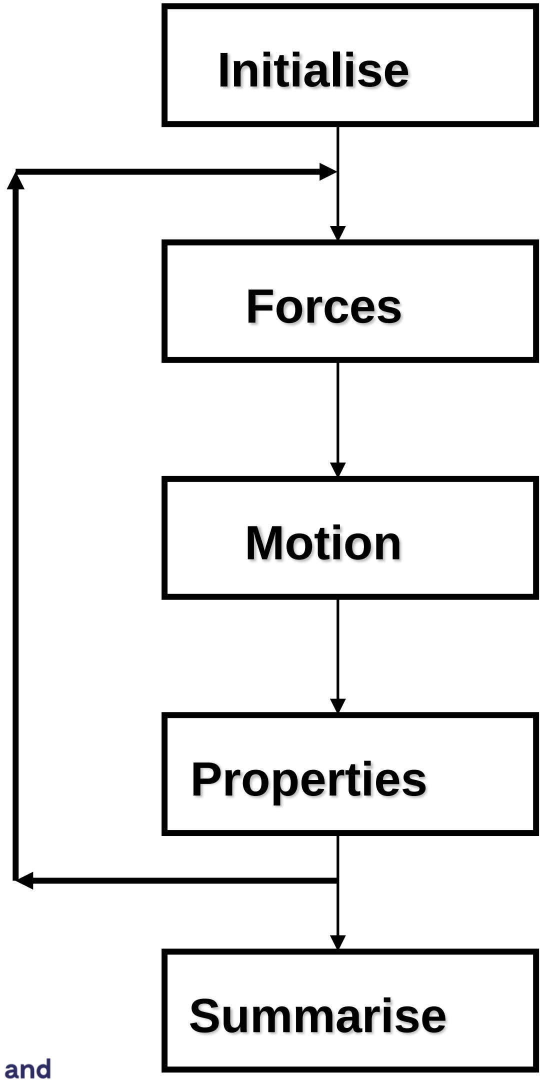
Then convert back to Cartesian:

$$\begin{pmatrix} r_{ij'x} \\ r_{ij'y} \\ r_{ij'z} \end{pmatrix} = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{pmatrix} \begin{pmatrix} f_{aij'} \\ f_{bij'} \\ f_{cij'} \end{pmatrix}$$

For van der Waals interactions use only the nearest images of atoms.

The minimum image convention limits the cut off used in the potential sum to half the shortest lattice parameter.

Key Stages in MD Simulation



Initialise

Forces

Motion

Properties

Summarise

- *Set up initial system*

  positions of atoms and initial velocities (3D Boltzmann distributed)

- *Calculate atomic forces*

  based on potential model

- *Calculate atomic motion*

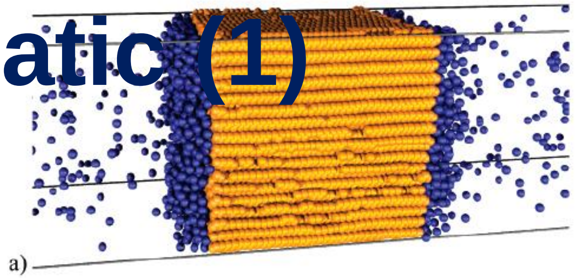
  via an integration algorithm

- *Calculate physical properties*

  basically collect instantaneous data for statistical purposes

- *Repeat !*
- *Produce final summary*

System Properties – Static (1)



- **Kinetic Energy:**

$$\langle K.E.\rangle = \left\langle \frac{1}{2}\sum_{i}^{N} m_i v_i^2 \right\rangle$$

- **Temperature:**

$$T = \frac{2}{3}\frac{\langle K.E.\rangle}{N k_B}$$

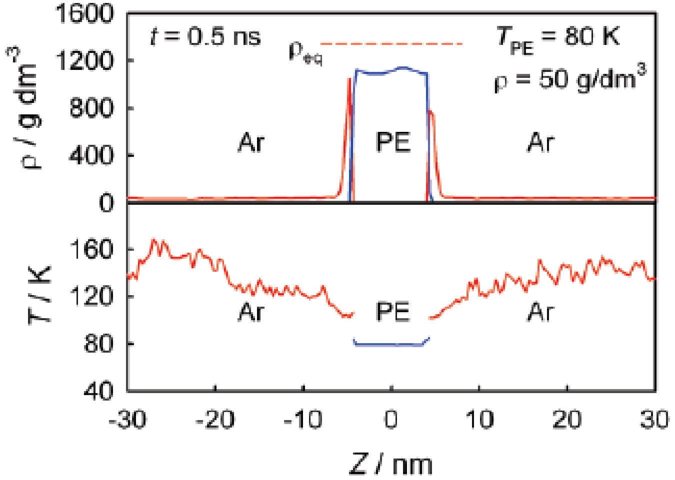
- **Configuration Energy:**

$$U_c = \left\langle \sum_{i}^{N-1}\sum_{j>i}^{N} V(r_{ij}) \right\rangle$$

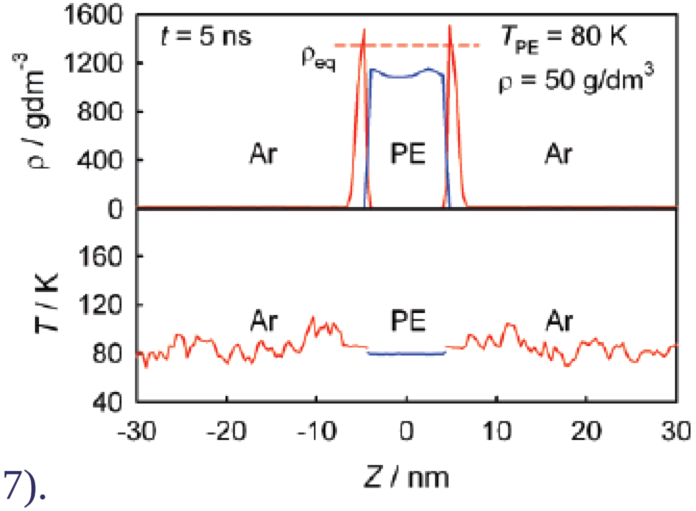
- **Pressure:**

$$PV = N k_B T - \frac{1}{3}\left\langle \sum_{i}^{N} \vec{r}_i \cdot \vec{f}_i \right\rangle$$

- **Specific heat:** $\langle \delta(U_c)^2\rangle = \frac{3}{2} N {k_B}^2 T^2 \left(1 - \frac{3}{2}\frac{N k_B}{C_v}\right)$



a)



Science and
Technology
Facilities Council

Scientific Comp

Data from: R. Rozas & T. Kraska, *J. Phys. Chem. C*, **111**, 15784, (2007).

System Properties – Static (2)

Structural Properties

- Pair correlation (Radial Distribution Function):

$$g(r) = \frac{\langle n(r) \rangle}{4\pi\rho r^2 \Delta r} = \frac{V}{N^2} \left\langle \sum_{i=1}^{N-1} \sum_{j \neq i}^{N} \delta(r - r_{ij}) \right\rangle$$

- Structure factor:

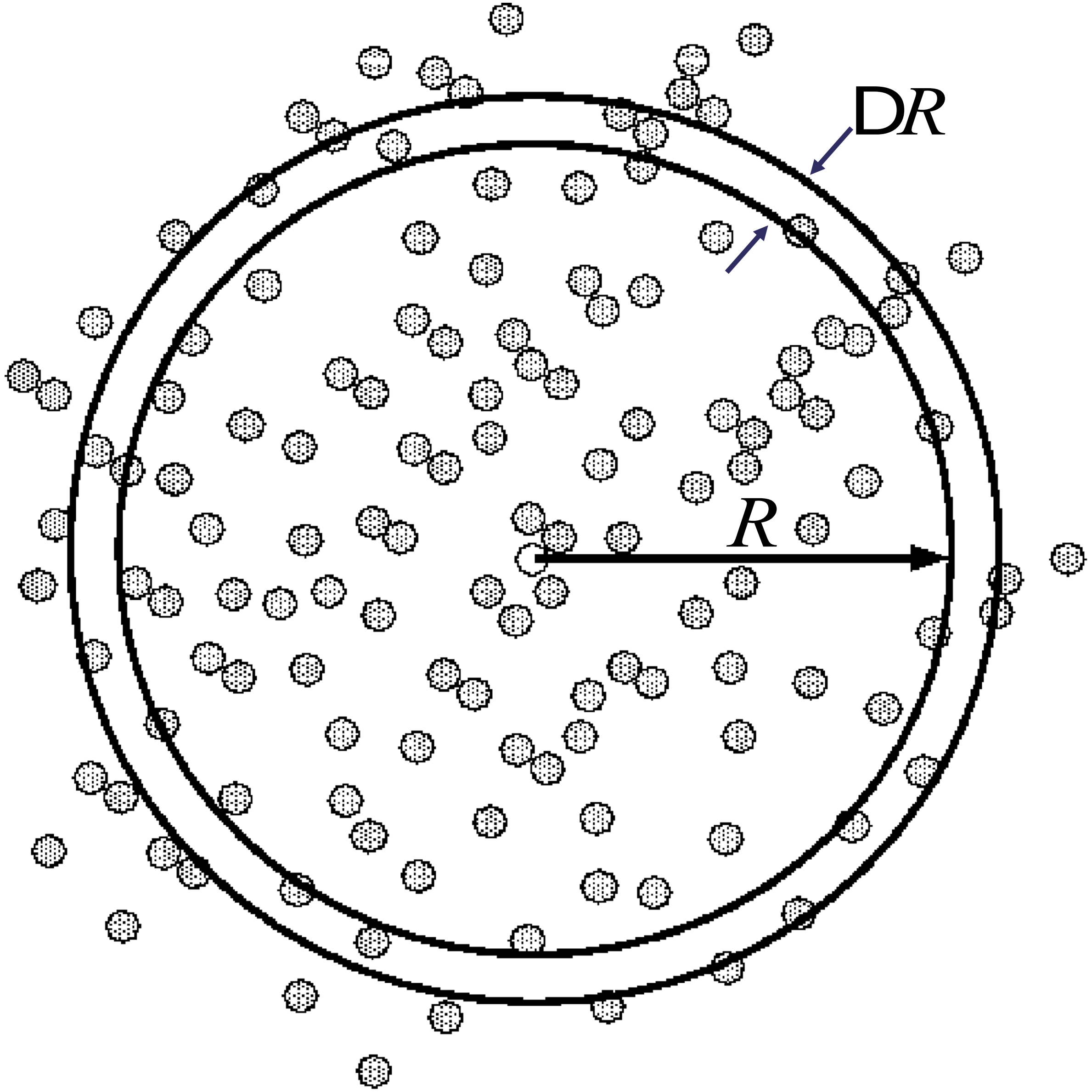
$$S(k) = 1 + 4\pi\rho \int_0^\infty \frac{\sin(kr)}{kr} (g(r) - 1)\, r^2\, dr$$

- Note: *S(k)* available from X-ray diffraction

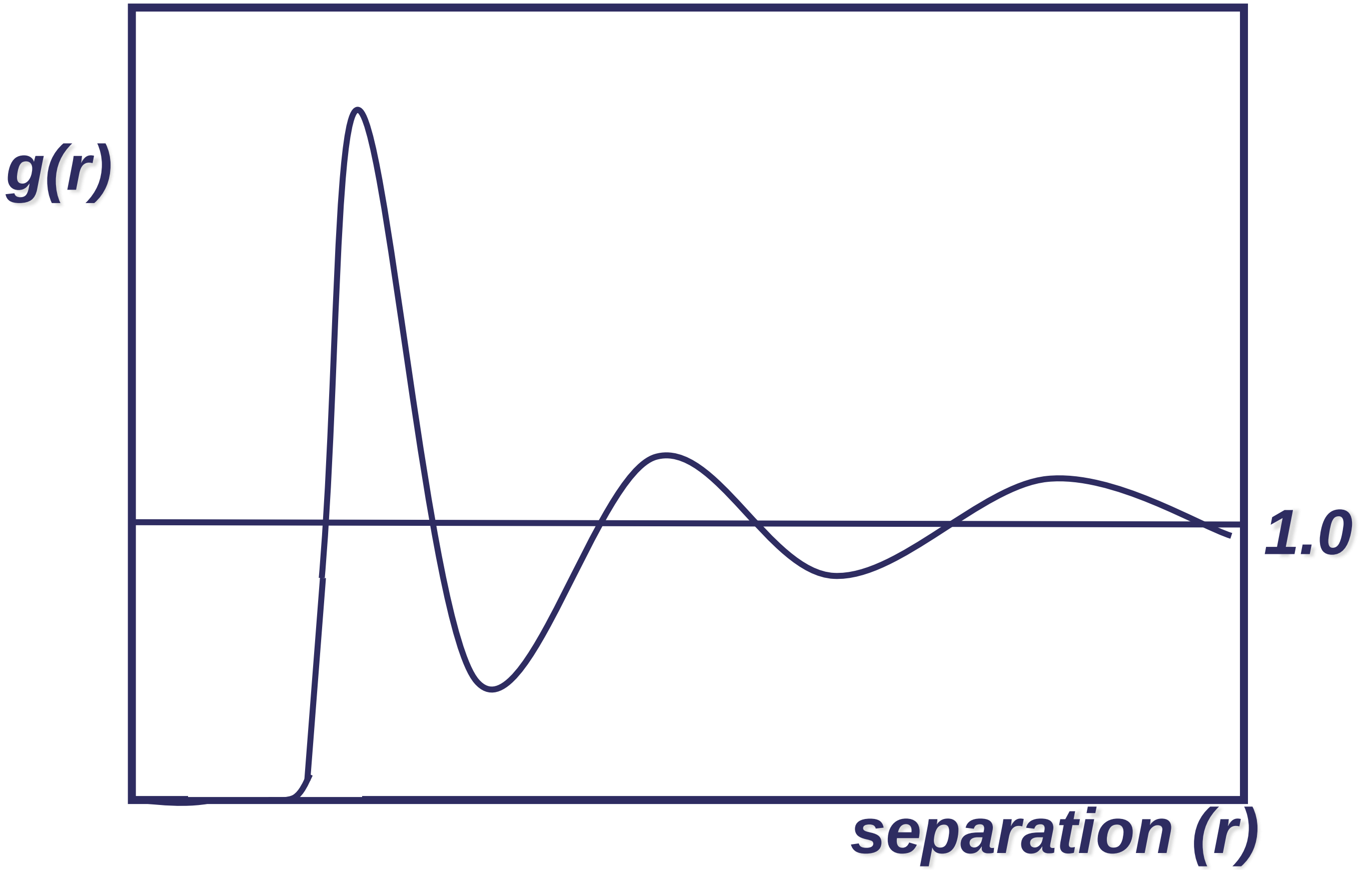
Science and
Technology
Facilities Council

Scientific Computing

Radial Distribution Function (RDF)



Typical RDF



System Properties – Dynamic (1)

Single correlation functions:

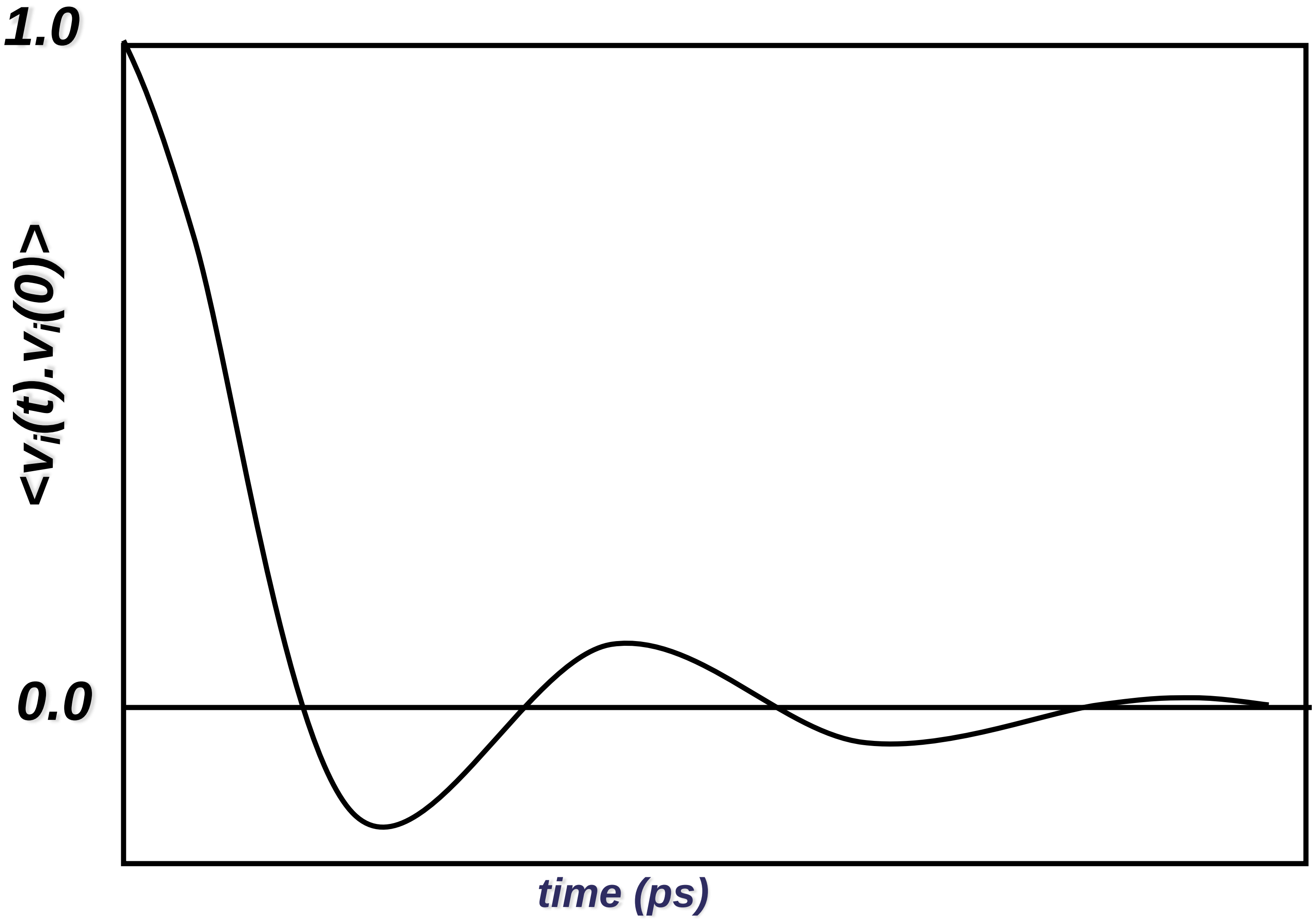
- **Mean squared displacement (Einstein relation)**

$$2Dt = \frac{1}{3}\langle |r_i(t) - r_i(0)|^2 \rangle$$

- **Velocity Autocorrelation (Green-Kubo relation)**

$$D = \frac{1}{3}\int_0^t \langle v_i(t) - v_i(0)\rangle \, dt$$

Typical VAF



System Properties – Dynamic (2)

Collective Correlation Functions: DL_POLY GUI

- **General van Hove correlation function**

$$G(\mathbf{r},t)=\frac{1}{N}\left\langle\sum_{i,j=1}^{N}\delta[r+r_i(0)-r_j(t)]\right\rangle$$

- **van Hove self-correlation function**

$$G_s(\mathbf{r},t)=\frac{1}{N}\left\langle\sum_{i}^{N}\delta[r-r_i(0)-r_i(t)]\right\rangle$$

- **van Hove distinct correlation function**

$$G_d(\mathbf{r},t)=\frac{1}{N}\left\langle\sum_{i}^{N}\sum_{j\neq i}^{N}\delta[r+r_i(0)-r_j(t)]\right\rangle$$

Science and
Technology
Facilities Council

Scientific Computing

Recommended Books

Elements of Molecular Dynamics W. Smith, 2017 (WWW)

Statistical Mechanics: Theory and Molecular Simulation Mark Tuckerman, Oxford Graduate Texts, (FE 2010, SE 2016)

Computer Simulation of Liquids M.P. Allen & D.J. Tildesley, Oxford (FE 1998, SE 2017)

The Art of Molecular Dynamics Simulation D.C. Rapaport, Cambridge University Press (2004)

Understanding Molecular Simulation Daan Frenkel & Berend Smit, Academic Press, (FE 1996, SE 2010)

Theory of Simple Liquids J.-P. Hansen and I.R. McDonald, Academic Press (1986).

Classical Mechanics H. Goldstein+ (FE 1950, SE 1980, TR 2001)

Molecular Modelling, Principles & Applications A.R. Leach, Pearson Prentice Hall (SE 2001)

UKRI Science and Technology Facilities Council

Scientific Computing

Introduction to DL_POLY

Resources

W. Smith and T.R. Forester
J. Molec. Graphics (1996), **14**, 136

W. Smith, C.W. Yong, P.M. Rodger
Molecular Simulation (2002), **28**, 385

I.T. Todorov, W. Smith, K. Trachenko, M.T. Dove
J. Mater. Chem. (2006), **16**, 1611-1618

W. Smith (Guest Editor)
Molecular Simulation (2006), **32**, 933

I.J. Bush, I.T. Todorov and W. Smith
Comp. Phys. Commun. (2006), **175**, 323-329

I. Todorov (Guest Editor)
Molecular Simulation (2021), **47**, 89

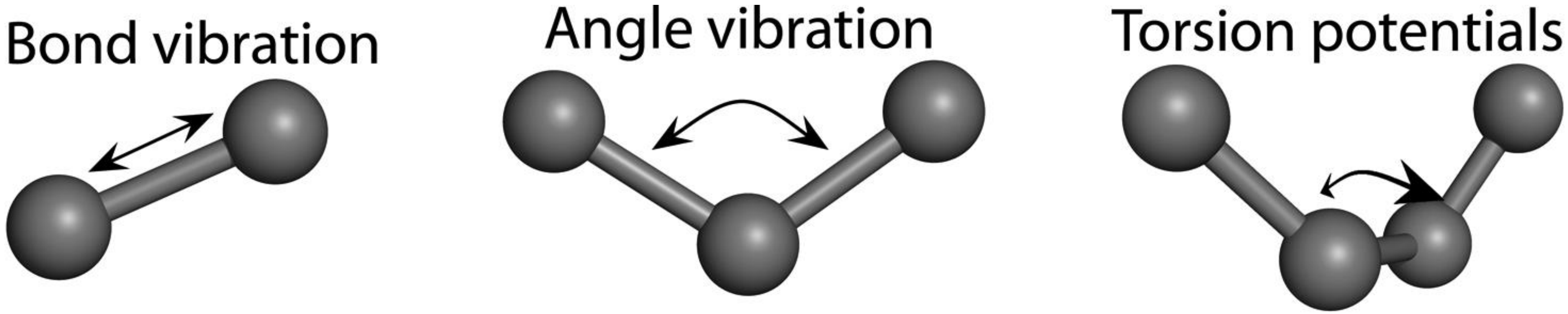
DL_POLY

General purpose classical particle dynamics simulation software package

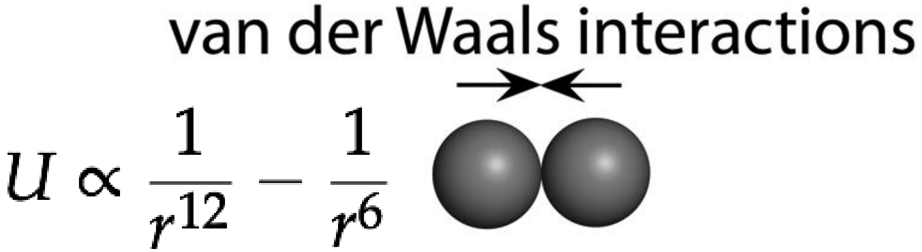
- MPI parallelised code
  - Created in 1993 as a DL project, supported as flagship by CCP5, HEC-MCC under EPSRC and NERC funding schemes
  - Support currently funded under CoSeC for HEC-MCC
- Current versions
  - 5.0+ – Domain Decomposition: released April 2021 (L-GPLv3.0) F95 (cmake)
  - 1.10 – Replicated data: last updated April 2017 (BSD2.0) – F90 (make)
  - https://gitlab.com/ccp5/dl-poly/ https://gitlab.com/DL_POLY_Classic
- Used widely by academics and industry
  - ~6000 verified users
  - 22 industrial licences 1996-2023

Simplified view of MD force field (FF)

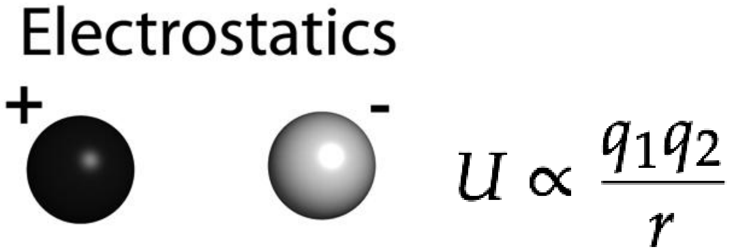
- **Bonded interactions** – also referred as *intra*-molecular



- **Non-bonded interactions** – also referred as *inter*-molecular



$$U \propto \frac{1}{r^{12}} - \frac{1}{r^{6}}$$

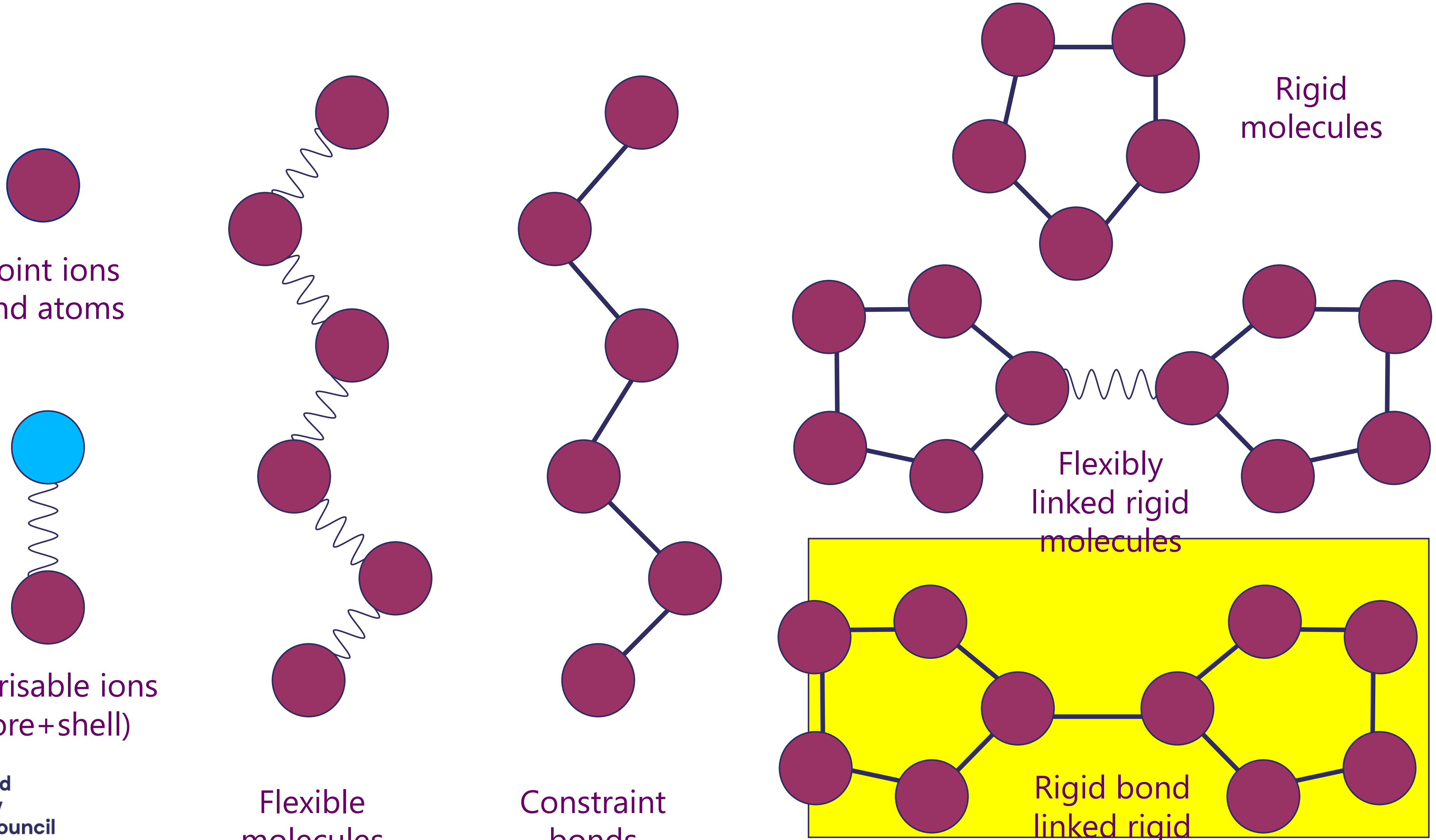


$$U \propto \frac{q_1 q_2}{r}$$

short-ranged

long-ranged

DL_POLY supported entities for MD



Point ions
and atoms

Polarisable ions
(core+shell)

Flexible
molecules

Constraint
bonds

Rigid
molecules

Flexibly
linked rigid
molecules

Rigid bond
linked rigid
molecules

DL_POLY FF mappings – I

- ***particle:*** a rigid ion or an atom (charged or not), a core or a shell of a polarisable ion (with or without associated degrees of freedom), a massless charged site. A particle is a countable object and has a **global ID index**.
- ***site:*** a particle prototype that serves to define the chemical & physical nature (topology/connectivity/stoichiometry) of a particle (mass, charge, frozen-ness). Sites are not atoms they are **prototypes**!
- ***Intra-molecular interactions:*** chemical bonds, bond angles, dihedral angles, improper dihedral angles, inversions. Usually, the members in a unit do not interact via an inter-molecular term. However, this can be overridden for some interactions. These are defined by **local *site***, ultimately relatable to the **global ID index** – ***#sites=#particles#max(gID)***
- ***Inter-molecular interactions:*** van der Waals, metal (2B/E/EAM, Gupta, Finnis-Sinclair, Sutton-Chen), Tersoff, three-body, four-body. These are defined by ***species***.

DL_POLY FF mappings – II

- ***Electrostatics:*** Standard Ewald*, Hautman-Klein (2D) Ewald*, **SPM Ewald (3D FFTs)**, Force-Shifted Coulomb, Reaction Field, Fennell damped FSC+RF, Distance dependent dielectric constant, Fuchs correction for non charge neutral MD cells.
- ***Ion polarisation*** via Dynamic (Adiabatic) or Relaxed shell model.
- ***External fields:*** Electric, Magnetic, Gravitational, Oscillating & Continuous Shear, Containing Sphere, Repulsive Wall.
- ***Intra-molecular like interactions:*** tethers, core shells units, constraint and PMF units, rigid body units. These are also defined by *site*.
- ***Potentials:*** parameterised analytical forms defining the interactions. These are always spherically symmetric!
- ***THE CHEMICAL NATURE OF PARTICLES DOES NOT CHANGE IN SPACE AND TIME!!!*** * **Except for DLC and MLIP!!!**

Science and Technology Facilities Council

Scientific Computing

DL_POLY interactions – capability

$$V(\vec{r}_1,\vec{r}_2,\ldots\ldots,\vec{r}_N) = \sum_{i,j}^{N'} U_{\text{pair}}(|\vec{r}_i-\vec{r}_j|) + \frac{1}{4\pi\varepsilon\varepsilon_0}\sum_{i,j}^{N'}\frac{q_i q_j}{|\vec{r}_i-\vec{r}_j|} +$$

$$\sum_{i,j,k}^{N'} U_{\text{Tersoff}}\left(\vec{r}_i,\vec{r}_j,\vec{r}_k\right) + \sum_{i,j,k}^{N'} U_{3-\text{body}}\left(\vec{r}_i,\vec{r}_j,\vec{r}_k\right) + \sum_{i,j,k,n}^{N'} U_{4-\text{body}}\left(\vec{r}_i,\vec{r}_j,\vec{r}_k,\vec{r}_n\right) +$$

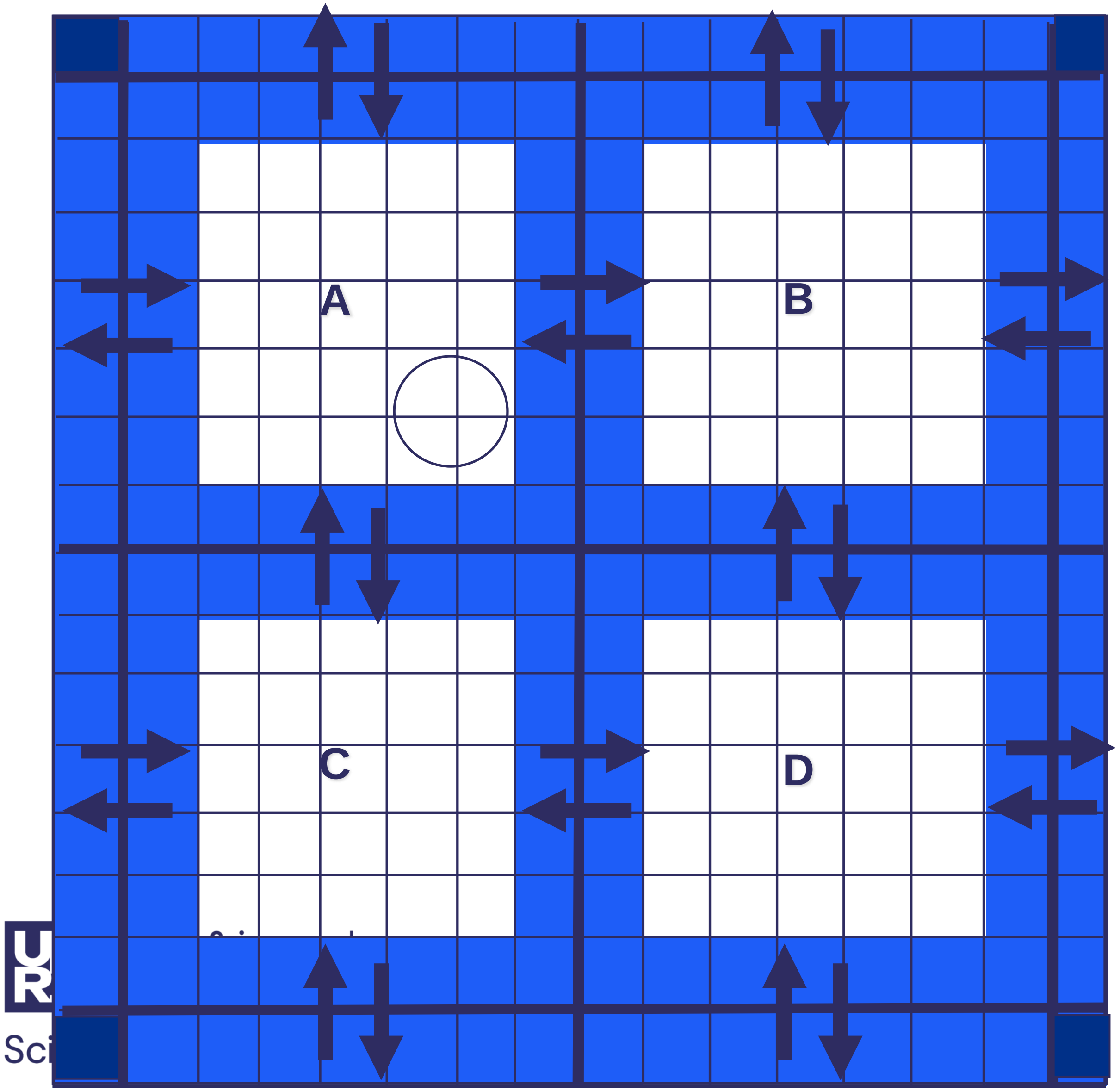
$$\varepsilon_{\text{metal}}\left(\sum_{i,j}^{N'} V_{\text{pair}}(|\vec{r}_i-\vec{r}_j|) + \sum_{i}^{N} F\left(\sum_{i,j}^{N'}\rho_{ij}(|\vec{r}_i-\vec{r}_j|)\right)\right) +$$

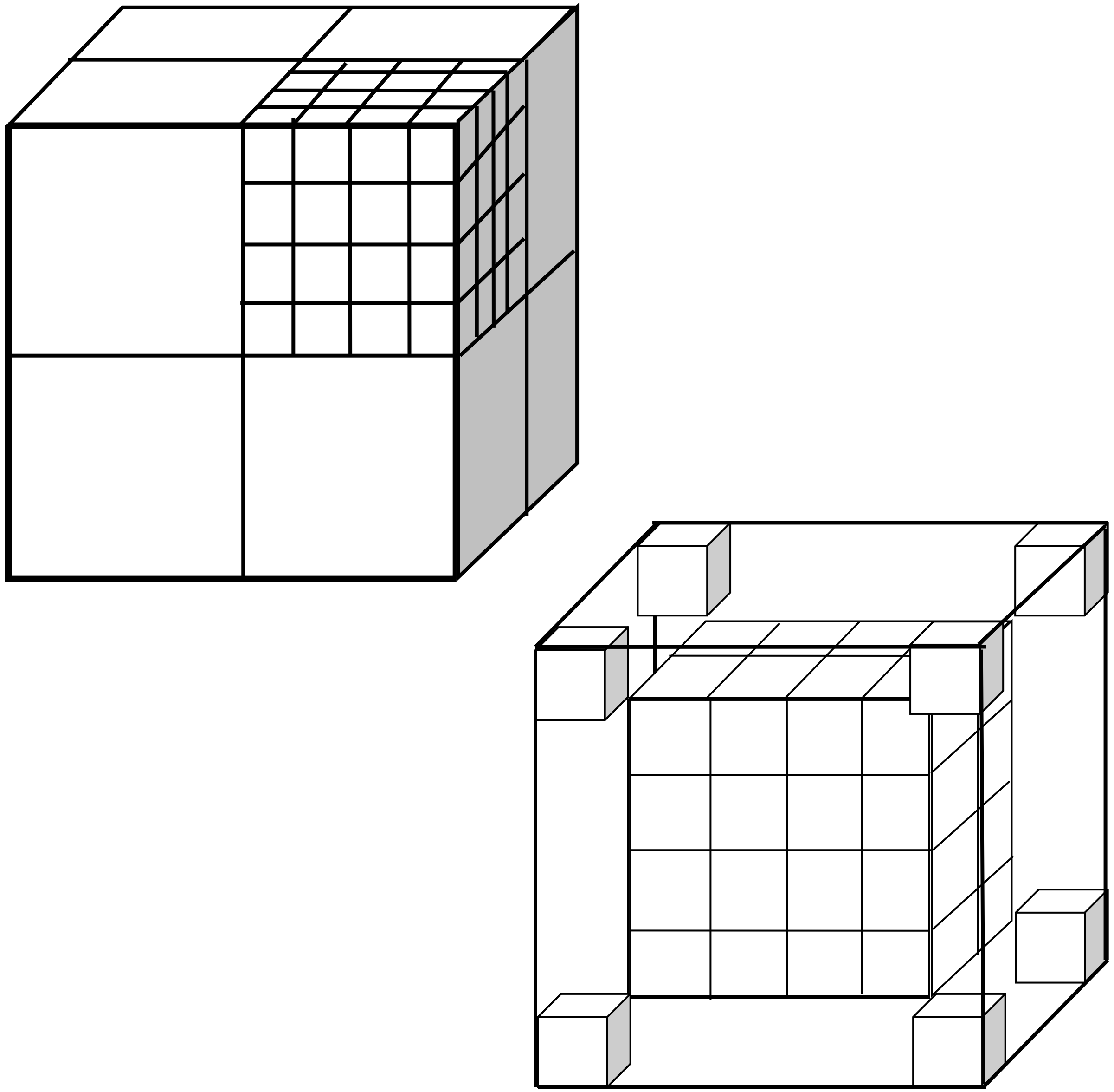
$$\sum_{i_{\text{bond}}}^{N_{\text{bond}}} U_{\text{bond}}\left(i_{\text{bond}},\vec{r}_a,\vec{r}_b\right) + \sum_{i_{\text{angle}}}^{N_{\text{angle}}} U_{\text{angle}}\left(i_{\text{angle}},\vec{r}_a,\vec{r}_b,\vec{r}_c\right) +$$

$$\sum_{i_{\text{dihed}}}^{N_{\text{dihed}}} U_{\text{dihed}}\left(i_{\text{dihed}},\vec{r}_a,\vec{r}_b,\vec{r}_c,\vec{r}_d\right) + \sum_{i_{\text{invers}}}^{N_{\text{invers}}} U_{\text{invers}}\left(i_{\text{invers}},\vec{r}_a,\vec{r}_b,\vec{r}_c,\vec{r}_d\right) +$$

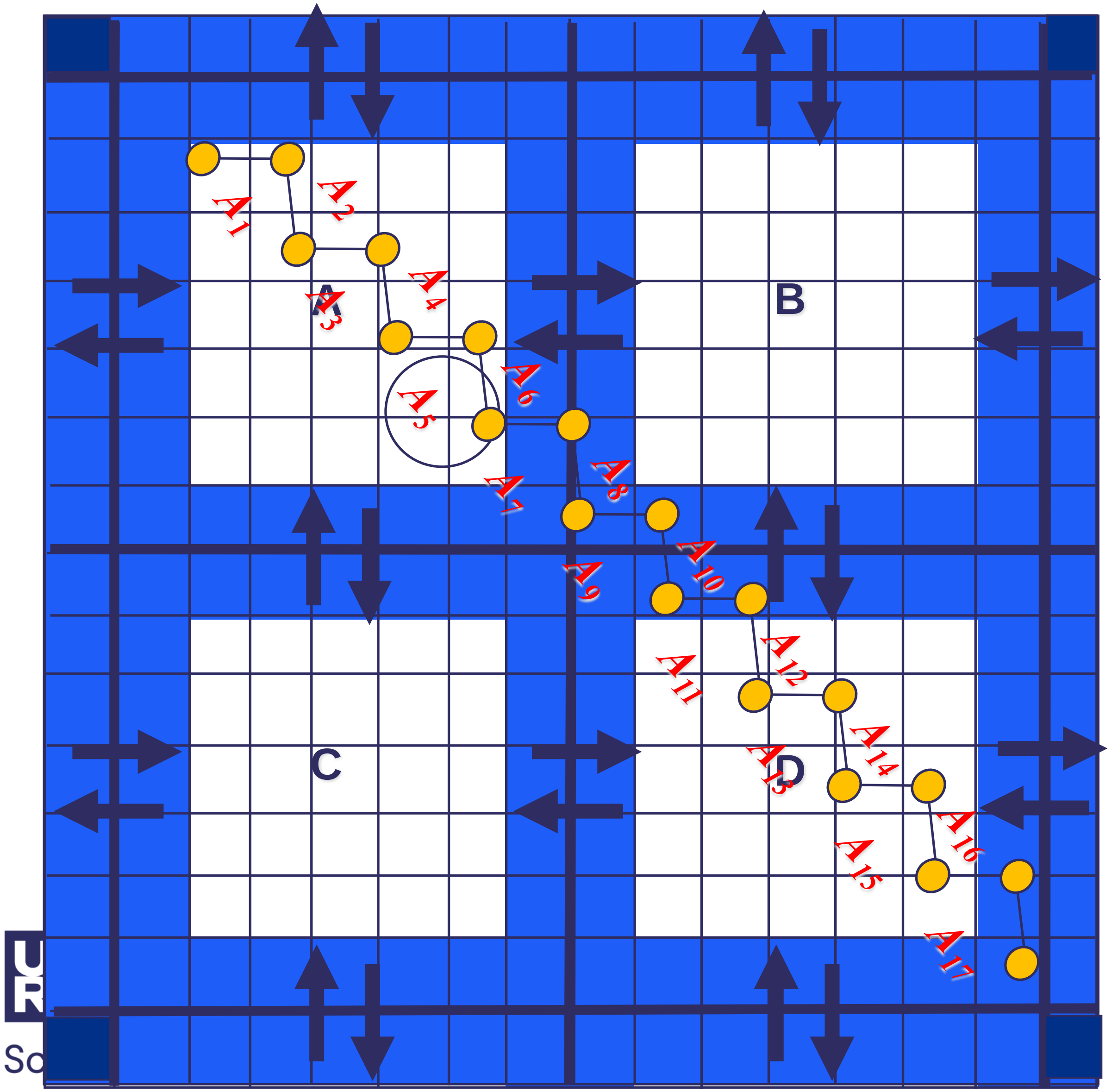
$$\sum_{i_{\text{tether}}}^{N_{\text{tether}}} U_{\text{tether}}\left(i_{\text{tether}},\vec{r}_t,\vec{r}_{t=0}\right) + \sum_{i_{\text{core-shell}}}^{N_{\text{core-shell}}} U_{\text{core-shell}}\left(i_{\text{core-shell}},|\vec{r}_i-\vec{r}_j|\right) + \sum_{i=1}^{N}\Phi_{\text{external}}\left(\vec{r}_i\right)$$

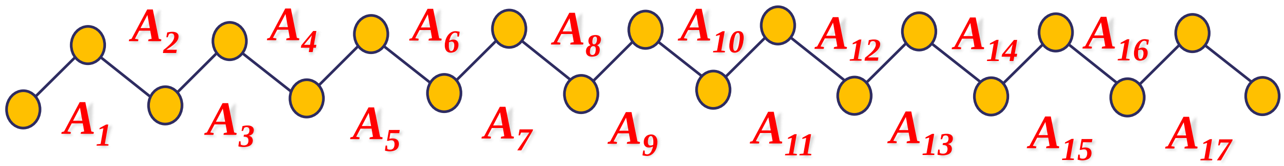
DL_POLY Domain Decomposition (particles)



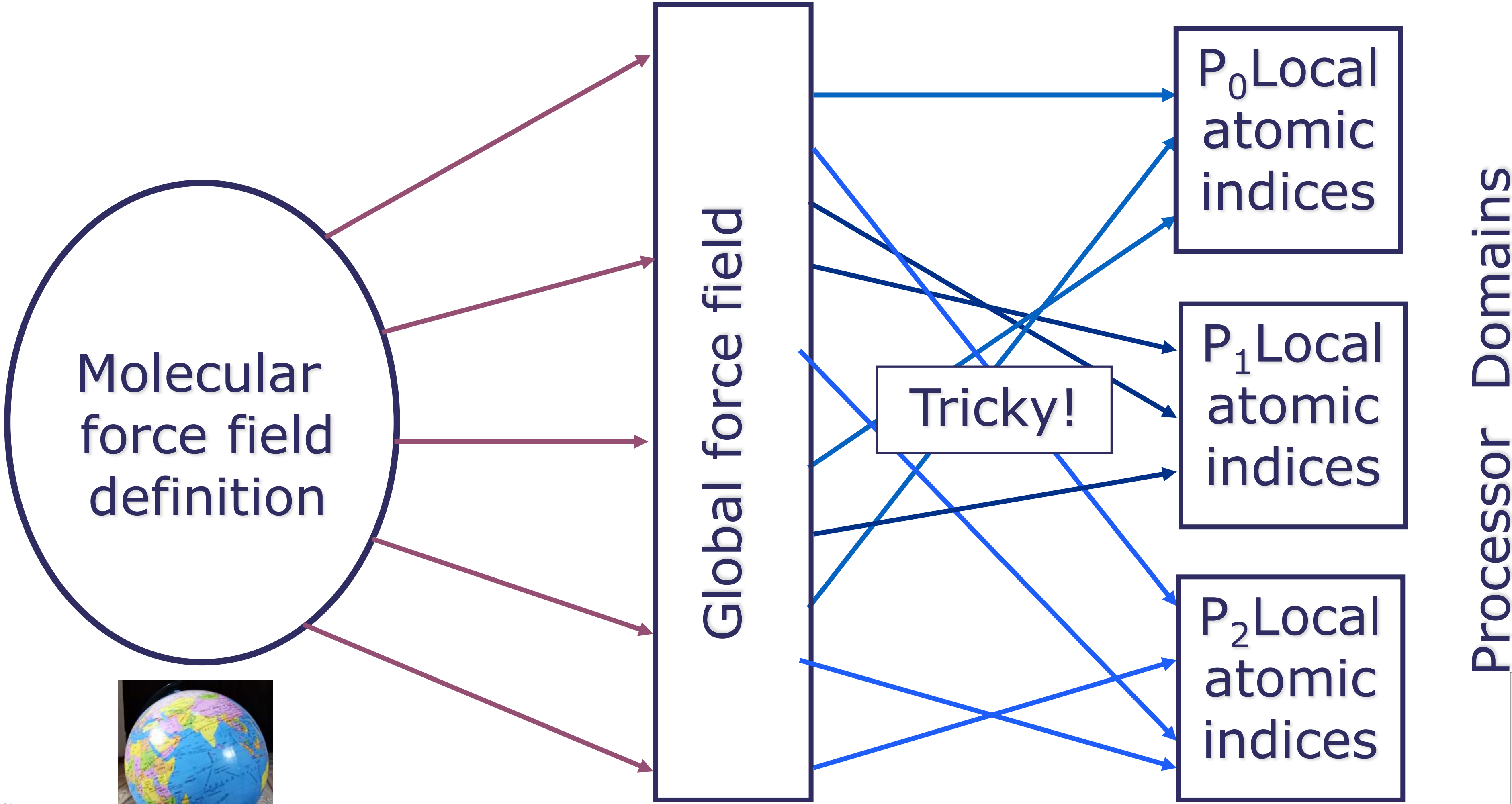


DL_POLY Domain Decomposition (topology)





DL_POLY topology (bonding int.) distribution



Molecular force field definition

Global force field

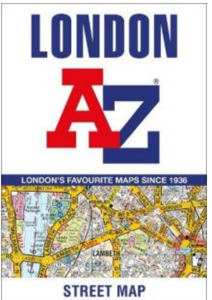
Tricky!

P_0Local atomic indices

P_1Local atomic indices

P_2Local atomic indices

Processor Domains

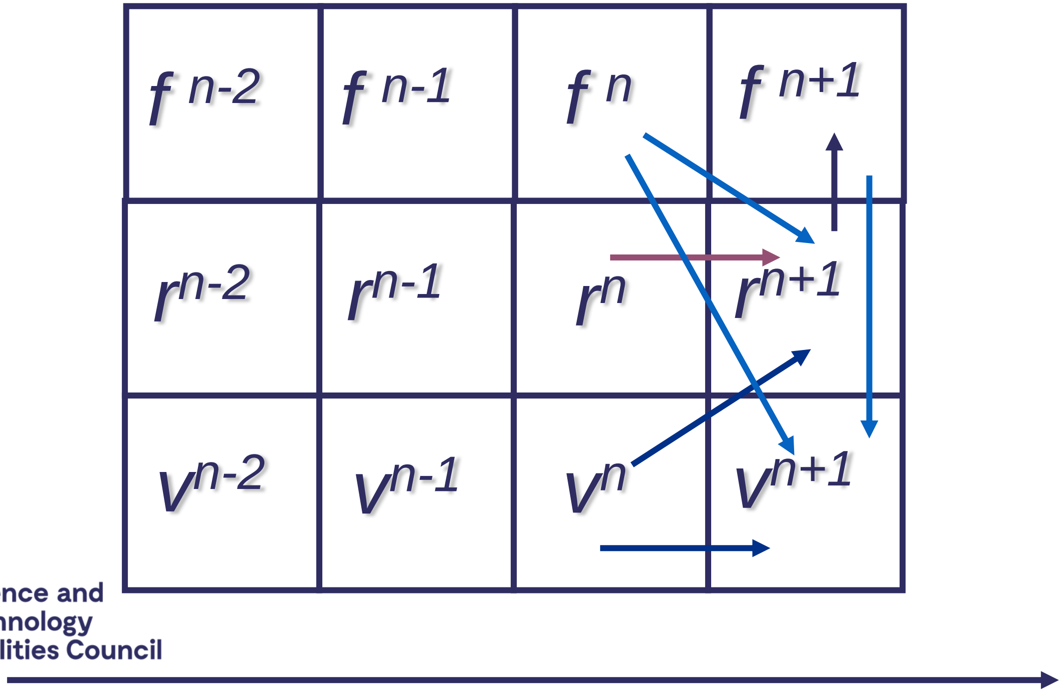


Science and Technology Facilities Council
Scientific Computing

Velocity Verlet integration

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \Delta t \vec{v}_i^{\,n} + \frac{\Delta t^2}{2m_i}\vec{f}_i^{\,n} + O(\Delta t^4)$$

$$\vec{v}_i^{\,n+1} = \vec{v}_i^{\,n} + \frac{\Delta t}{2m_i}\left(\vec{f}_i^{\,n} + \vec{f}_i^{\,n+1}\right) + O(\Delta t^2)$$



$f^{\,n-2}$	$f^{\,n-1}$	$f^{\,n}$	$f^{\,n+1}$
r^{n-2}	r^{n-1}	r^{n}	r^{n+1}
v^{n-2}	v^{n-1}	v^{n}	v^{n+1}

Discrete time

Application in Practice

$$\vec{v}_i^{\,n+1/2} = \vec{v}_i^{\,n} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n}$$

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \Delta t \vec{v}_i^{\,n+1/2}$$

$$\vec{v}_i^{\,n+1} = \vec{v}_i^{\,n+1/2} + \frac{\Delta t}{2m_i}\vec{f}_i^{\,n+1}$$

DL_POLY Integrators

- **NVE^**
- **NV*T* (E_{kin})** Evans^
- **NVT dpdS0/1/2^** Sharlow 0th,1st,2nd order splitting, **dpd mdvv**
- **NVT** Andersen, Langevin, Berendsen, Nosé-Hoover ^, GST^
- **NPT** Langevin, Berendsen, Nosé-Hoover, Martyna-Tuckerman-Klein
- **NPT dpd** Langevin^
- **NσT/NPnAT/NPnγT** Langevin^, Berendsen, Nosé-Hoover, Martyna-Tuckerman-Klein^

Note: *CoM motion is removed from non-conserving integrators!*

Constraints & Rigid Body Solvers:

- **RATTLE, No_Squish**

DL_POLY Special Features

- **Radiation damage driven features:**
  - defects analysis
  - boundary/stochastic thermostats
  - volumetric expansion (integer) – *nfold Nx Ny Nz*
  - replay history
  - variable time step algorithm
- **Extra ensembles:**
  - DPD, Langevin, Andersen, MTK, GST
  - extensions of NsT to NP_nAT and $NP_n\gamma T$
- **Infrequent k-space Ewald evaluation**
- **Direct VdW/Metal**
- **Force shifted VdW**
- **I/O driven features Parallel I/O & netCDF**
- **Potential tapering**
- **Extensions: PLUMED, OpenKIM**
- **VdW potentials mixing schemes**

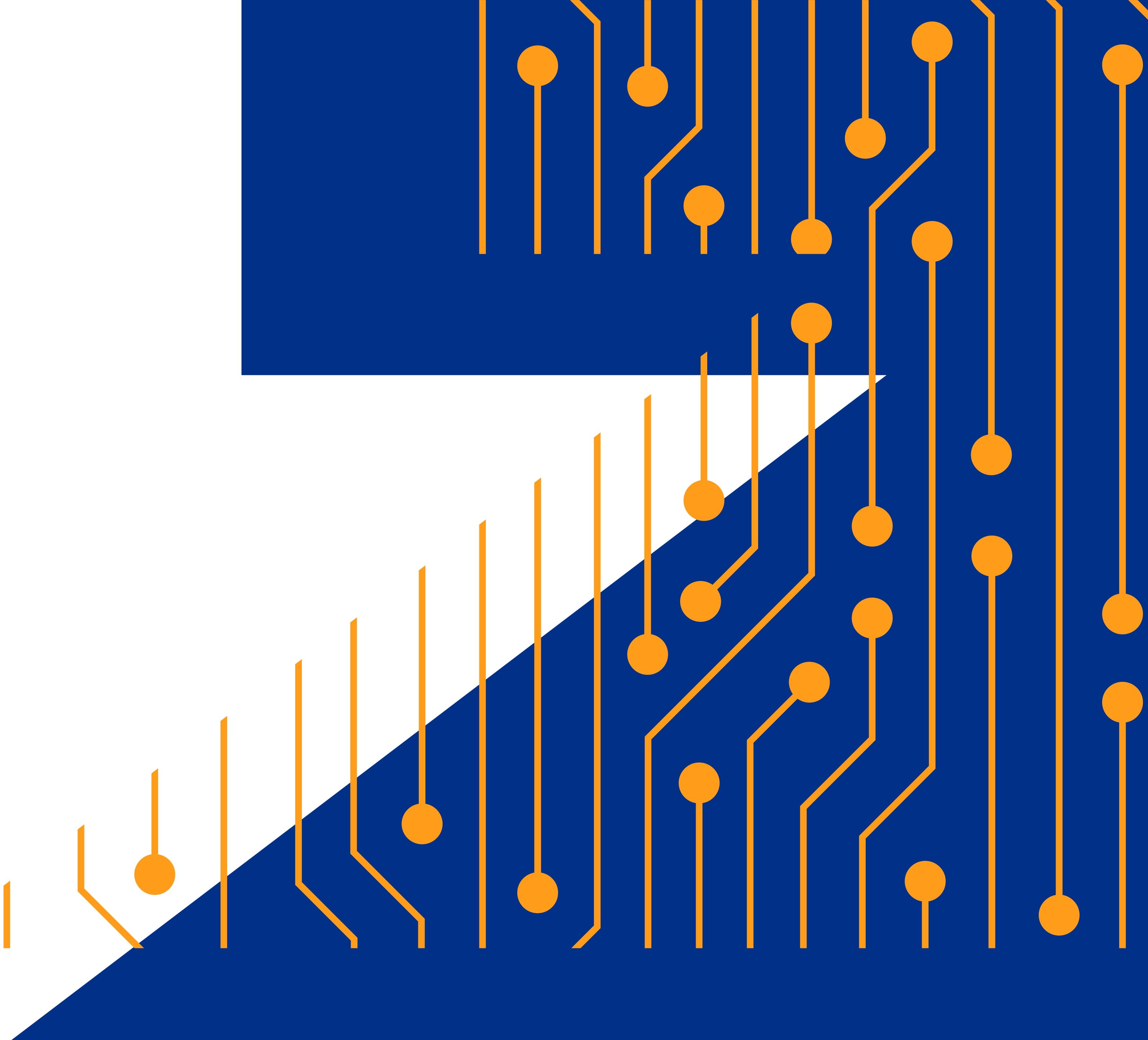
Science and Technology Facilities Council

Scientific Computing

Science and
Technology
Facilities Council

Scientific Computing

DL_POLY I/O

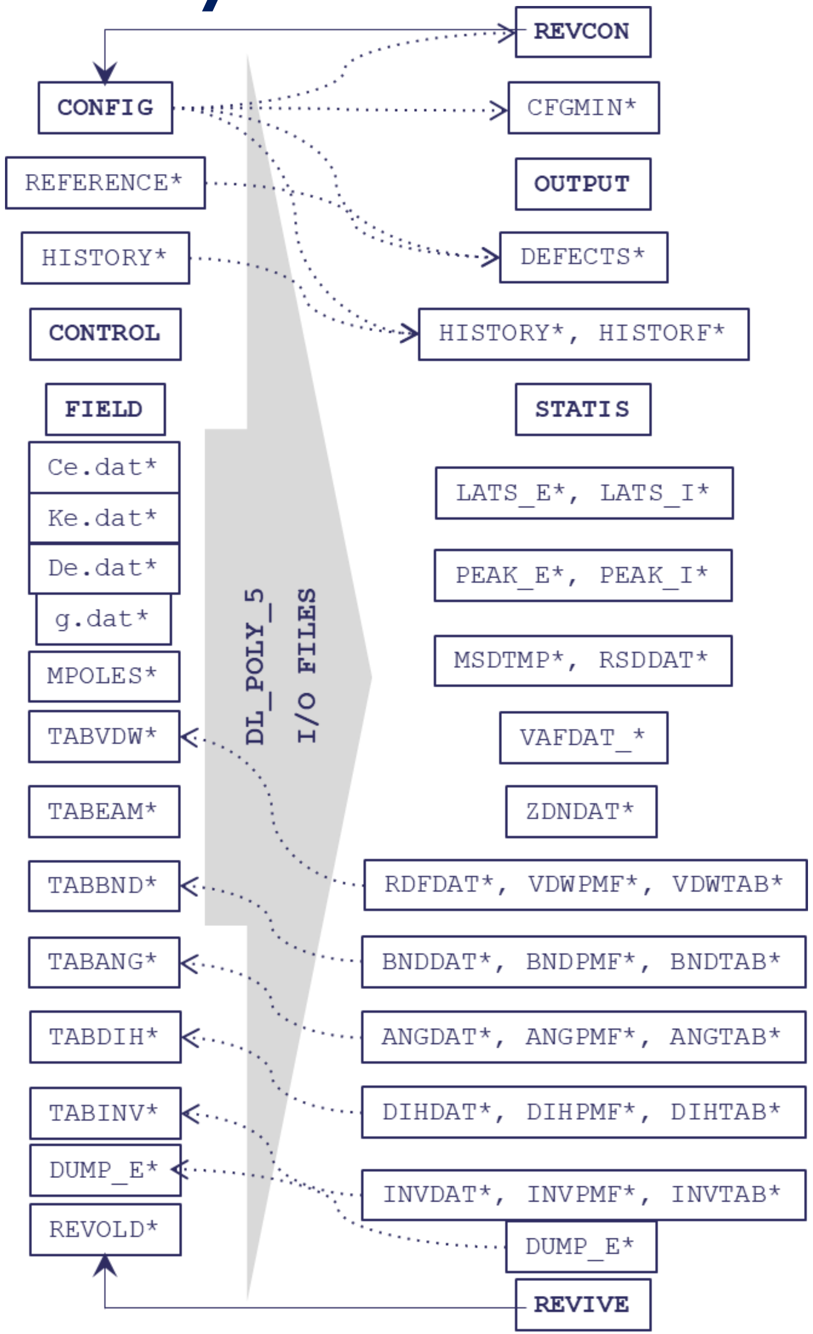


Science and
Technology
Facilities Council

Scientific Computing

I/O Files

- Crystallographic (Dynamic) data
- Reference data for DEFECTS
- Traj. data for replay
- Simulation controls
- Molecular/Topological Data
- Tabulated interactions
- Restart data



CONFIG
REFERENCE*
HISTORY*
CONTROL
FIELD
Ce.dat*
Ke.dat*
De.dat*
g.dat*
MPOLES*
TABVDW*
TABEAM*
TABBND*
TABANG*
TABDIH*
TABINV*
DUMP_E*
REVOLD*

DL_POLY_5
I/O FILES

REVCON
CFGMIN*
OUTPUT
DEFECTS*
HISTORY*, HISTORF*
STATIS
LATS_E*, LATS_I*
PEAK_E*, PEAK_I*
MSDTMP*, RSDDAT*
VAFDAT_*
ZDNDAT*
RDFDAT*, VDWPMF*, VDWTAB*
BNDDAT*, BNDPMF*, BNDTAB*
ANGDAT*, ANGPMF*, ANGTAB*
DIHDAT*, DIHPMF*, DIHTAB*
INVDAT*, INVPMF*, INVTAB*
DUMP_E*
REVIVE

- Final & CGM configurations
- Best CGM configuration
- Simulation summary data
- Trajectory data
- Defects data
- Statistics data
- RSD, MSD & T_{inst} data
- VAF data
- Intra PDF data
- Inter PDF/RDF data
- Z density data
- Restart data

Science and Technology Facilities Council
Scientific Computing

DL_POLY Units

Internally, DL_POLY uses atomic scale units:

- Mass – mass of H atom (D) [Daltons]
- Charge – charge on proton (e)
- Length – Angstroms (Å)
- Time – picoseconds (ps)
- Force – D Å ps^{-2}
- Energy – D Å^{2} ps^{-2} [10 J mol^{-1}]
- angles – rad [radians]

- For I/O purposes:
  - Temperature is expressed in Kelvin
  - Pressure is expressed in k-atm
  - Angles are expressed in degrees (not radians)

However, input- and output-wise there is a great flexibility of community specific units, including DPD units.

Acceptable DL_POLY Units

UNITS directive in FIELD file allows to opt for the following energy units

- Internal DL_POLY units – 10 J mol^{-1}
- Electron-volts – eV
- kilo calories per mol – k-cal mol^{-1}
- kilo Joules per mol – k-J mol^{-1}
- Kelvin per Boltzmann – K $Boltzmann^{-1}$
- DPD relative units – Boltzmann Temperature

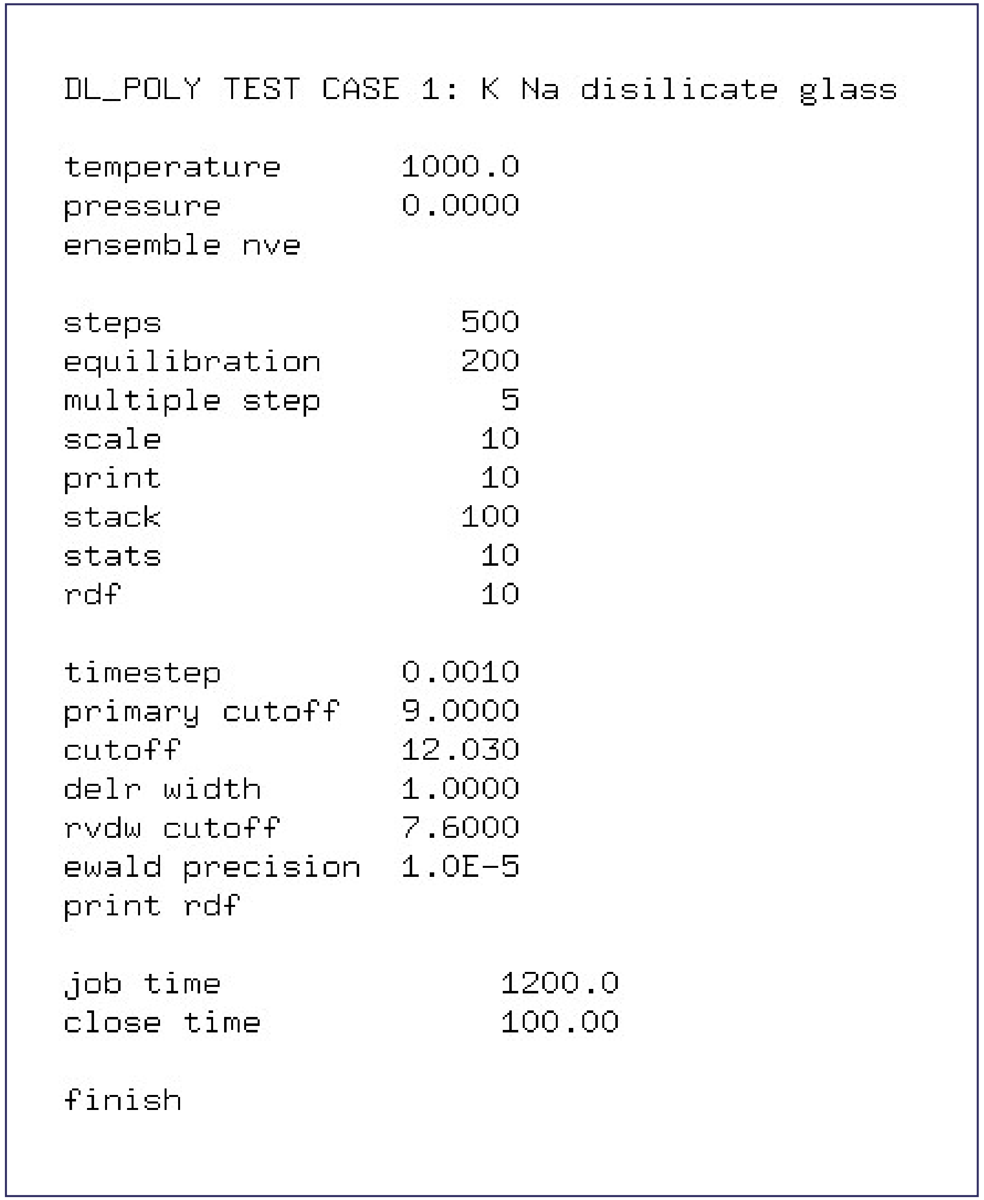
All interaction MUST have the same energy units!
Not only in FIELD but in TABVDW, TABEAM, TAB*INT*!

CONTROL File

- **SIMULATION CONTROL**
- **Free Format**
- **Mandatory**
- **Driven by *keywords:***

keyword [options] {data}
e.g.:

ensemble NPT Hoover 1.0 8.0



```
DL_POLY TEST CASE 1: K Na disilicate glass

temperature      1000.0
pressure         0.0000
ensemble nve

steps               500
equilibration       200
multiple step         5
scale                10
print                10
stack               100
stats                10
rdf                  10

timestep         0.0010
primary cutoff   9.0000
cutoff           12.030
delr width       1.0000
rvdw cutoff      7.6000
ewald precision  1.0E-5
print rdf

job time             1200.0
close time           100.00

finish
```

CONTROL File

- **SIMULATION CONTROL**
- **Free Format**
- **Mandatory**
- **Driven by *keywords:***

keyword [options] {data}
e.g.:

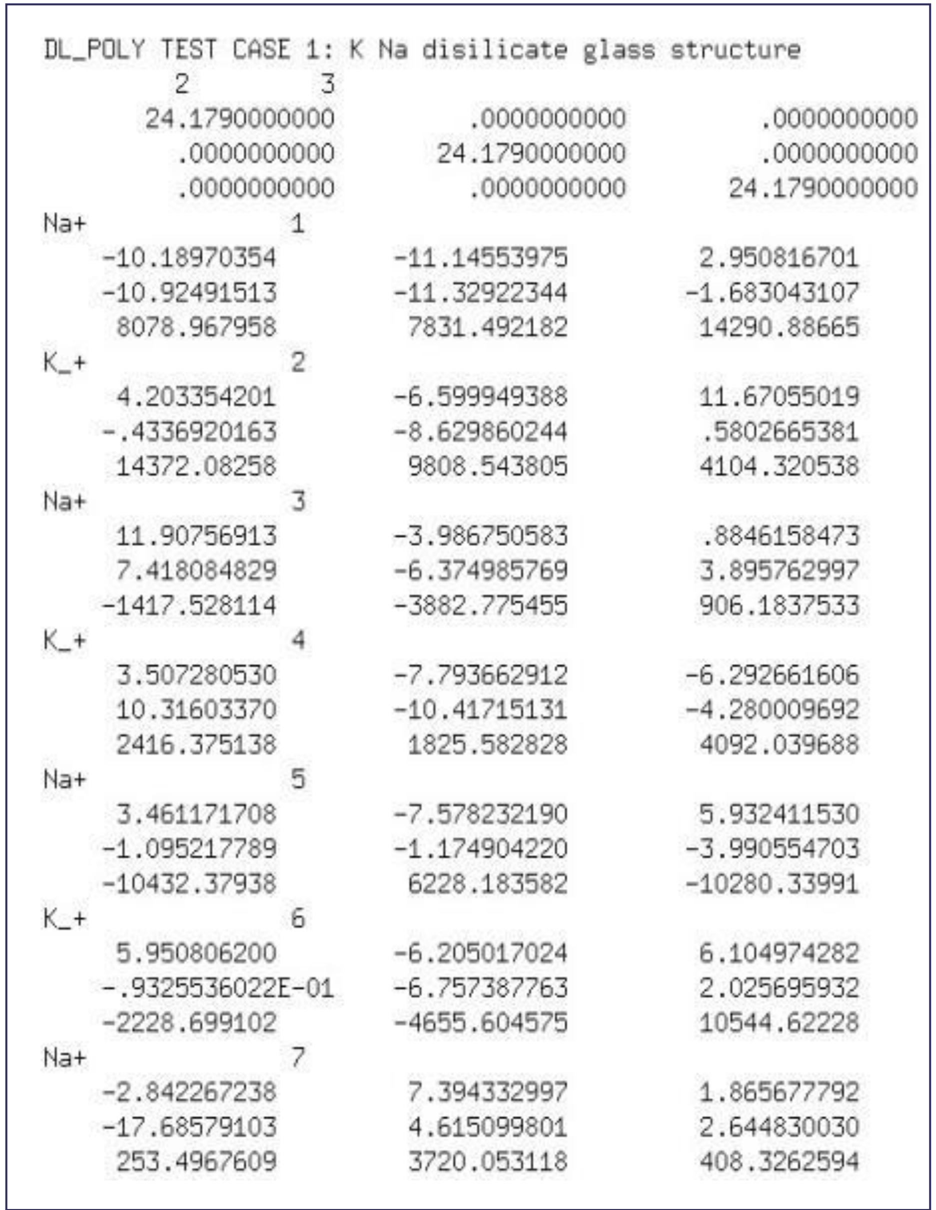
ensemble NPT Hoover 1.0 8.0

```
title DL_POLY_5 CONTROL DIRECTIVES

# RESTART OPTIONS
restart noscale

# SYSTEM TARGET TEMPERATURE AND PRESSURE
pressure_hydrostatic  0.001 katm
temperature  300.0 K

# SYSTEM CUTOFFS AND ELECTROSTATICS
vdw_cutoff  8 ang
padding  0.35 ang
cutoff  10.0 ang
coul_method spme
spme_precision 1e-05

# INTEGRATION FLAVOUR, ENSEMBLE AND PSEUDO THERMOSTAT
ensemble nst
ensemble_method berendsen
ensemble_thermostat_coupling  0.5 ps
ensemble_barostat_coupling  1.5 ps

# INTEGRATION TIMESTEP
timestep  0.001 ps

# SIMULATION & EQUILIBRATION LENGTH
time_run  10000 steps
time_equilibration  1000 steps

# EQUILIBRATION DIRECTIVES
reset_temperature_interval  1 steps


# EXECUTION TIME
time_job  1000.0 s
time_close  10.0 s
```

Science and
Technology
Facilities Council
Scientific Computing

CONFIG [REVCON,CFGMIN] File

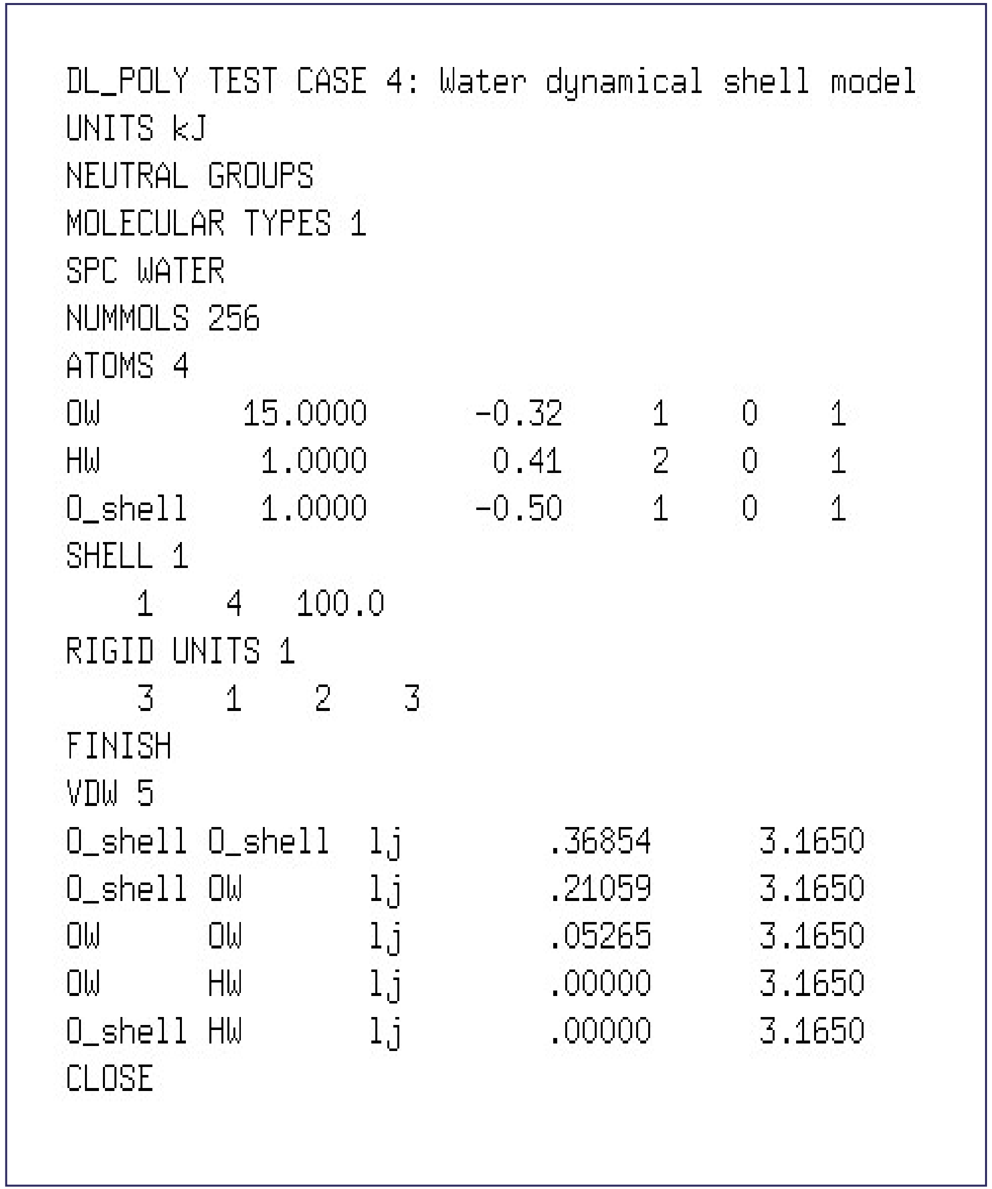
- **Initial atomic coordinates**
- **Format**
  - **Integers (I10)**
  - **Reals (F20)**
  - **Names (A8)**
- **Mandatory**
- **Units:**
  - **Position – Angstroms (Å)**
  - **Velocity – Å ps^{-1}**
  - **Force – D Å ps^{-2}**
- **Construction: A GUI or DL_FIELD essential for complex systems**



```
DL_POLY TEST CASE 1: K Na disilicate glass structure
         2         3
       24.1790000000         .0000000000         .0000000000
         .0000000000       24.1790000000         .0000000000
         .0000000000         .0000000000       24.1790000000
Na+             1
        -10.18970354        -11.14553975         2.950816701
        -10.92491513        -11.32922344        -1.683043107
         8078.967958         7831.492182         14290.88665
K_+             2
         4.203354201        -6.599949388         11.67055019
        -.4336920163        -8.629860244         .5802665381
         14372.08258         9808.543805         4104.320538
Na+             3
         11.90756913        -3.986750583         .8846158473
         7.418084829        -6.374985769         3.895762997
        -1417.528114        -3882.775455         906.1837533
K_+             4
         3.507280530        -7.793662912        -6.292661606
         10.31603370        -10.41715131        -4.280009692
         2416.375138         1825.582828         4092.039688
Na+             5
         3.461171708        -7.578232190         5.932411530
        -1.095217789        -1.174904220        -3.990554703
        -10432.37938         6228.183582        -10280.33991
K_+             6
         5.950806200        -6.205017024         6.104974282
    -.9325536022E-01        -6.757387763         2.025695932
        -2228.699102        -4655.604575         10544.62228
Na+             7
        -2.842267238         7.394332997         1.865677792
        -17.68579103         4.615099801         2.644830030
         253.4967609         3720.053118         408.3262594
```

FIELD File

- Force Field specification
- Mandatory
- Format:
  - Integers (I5)
  - Reals (F12)
  - Names (A8)
  - Keywords (A4)
- Maps on to CONFIG file structure
- Construction
  - Small systems – by hand
  - Large systems – nfold or GUI or DL_FIELD!

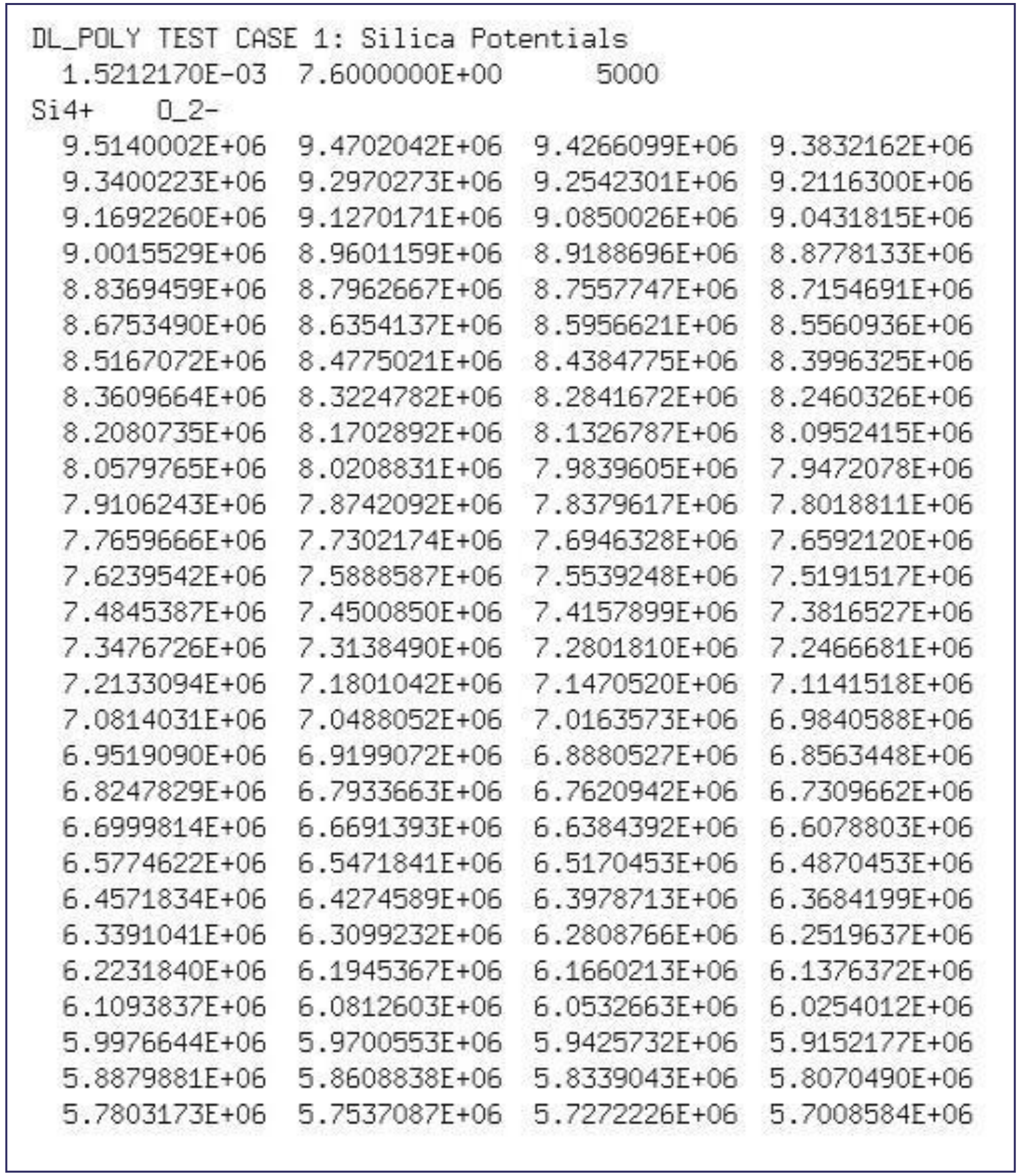


```
DL_POLY TEST CASE 4: Water dynamical shell model
UNITS kJ
NEUTRAL GROUPS
MOLECULAR TYPES 1
SPC WATER
NUMMOLS 256
ATOMS 4
OW         15.0000      -0.32     1    0    1
HW          1.0000       0.41     2    0    1
O_shell     1.0000      -0.50     1    0    1
SHELL 1
    1    4   100.0
RIGID UNITS 1
    3    1    2    3
FINISH
VDW 5
O_shell O_shell  lj        .36854      3.1650
O_shell OW       lj        .21059      3.1650
OW      OW       lj        .05265      3.1650
OW      HW       lj        .00000      3.1650
O_shell HW       lj        .00000      3.1650
CLOSE
```

TABLE File

- Defines non-analytic pair (vdw) potentials
- Format
    - **Integers(I10)**
    - **Reals (F15)**
    - **Names (A8)**
- Conditional, activated by FIELD file option
- Potential & Force
- NB force (here) is:

$$G(r) = -r \frac{\partial}{\partial r} U(r)$$



```
DL_POLY TEST CASE 1: Silica Potentials
  1.5212170E-03  7.6000000E+00      5000
Si4+    O_2-
  9.5140002E+06  9.4702042E+06  9.4266099E+06  9.3832162E+06
  9.3400223E+06  9.2970273E+06  9.2542301E+06  9.2116300E+06
  9.1692260E+06  9.1270171E+06  9.0850026E+06  9.0431815E+06
  9.0015529E+06  8.9601159E+06  8.9188696E+06  8.8778133E+06
  8.8369459E+06  8.7962667E+06  8.7557747E+06  8.7154691E+06
  8.6753490E+06  8.6354137E+06  8.5956621E+06  8.5560936E+06
  8.5167072E+06  8.4775021E+06  8.4384775E+06  8.3996325E+06
  8.3609664E+06  8.3224782E+06  8.2841672E+06  8.2460326E+06
  8.2080735E+06  8.1702892E+06  8.1326787E+06  8.0952415E+06
  8.0579765E+06  8.0208831E+06  7.9839605E+06  7.9472078E+06
  7.9106243E+06  7.8742092E+06  7.8379617E+06  7.8018811E+06
  7.7659666E+06  7.7302174E+06  7.6946328E+06  7.6592120E+06
  7.6239542E+06  7.5888587E+06  7.5539248E+06  7.5191517E+06
  7.4845387E+06  7.4500850E+06  7.4157899E+06  7.3816527E+06
  7.3476726E+06  7.3138490E+06  7.2801810E+06  7.2466681E+06
  7.2133094E+06  7.1801042E+06  7.1470520E+06  7.1141518E+06
  7.0814031E+06  7.0488052E+06  7.0163573E+06  6.9840588E+06
  6.9519090E+06  6.9199072E+06  6.8880527E+06  6.8563448E+06
  6.8247829E+06  6.7933663E+06  6.7620942E+06  6.7309662E+06
  6.6999814E+06  6.6691393E+06  6.6384392E+06  6.6078803E+06
  6.5774622E+06  6.5471841E+06  6.5170453E+06  6.4870453E+06
  6.4571834E+06  6.4274589E+06  6.3978713E+06  6.3684199E+06
  6.3391041E+06  6.3099232E+06  6.2808766E+06  6.2519637E+06
  6.2231840E+06  6.1945367E+06  6.1660213E+06  6.1376372E+06
  6.1093837E+06  6.0812603E+06  6.0532663E+06  6.0254012E+06
  5.9976644E+06  5.9700553E+06  5.9425732E+06  5.9152177E+06
  5.8879881E+06  5.8608838E+06  5.8339043E+06  5.8070490E+06
  5.7803173E+06  5.7537087E+06  5.7272226E+06  5.7008584E+06
```

TABEAM File

- Defines embedded atom potentials
- Format
    - Integers(I10)
    - Reals (F15)
    - Names (A8)
- Conditional, activated by FIELD file option
- Potentials only
- pair, embed & dens keywords for atom types followed by data records (4 real numbers per record)
- Individual interpolation arrays

REVOLD [REVIVE] File

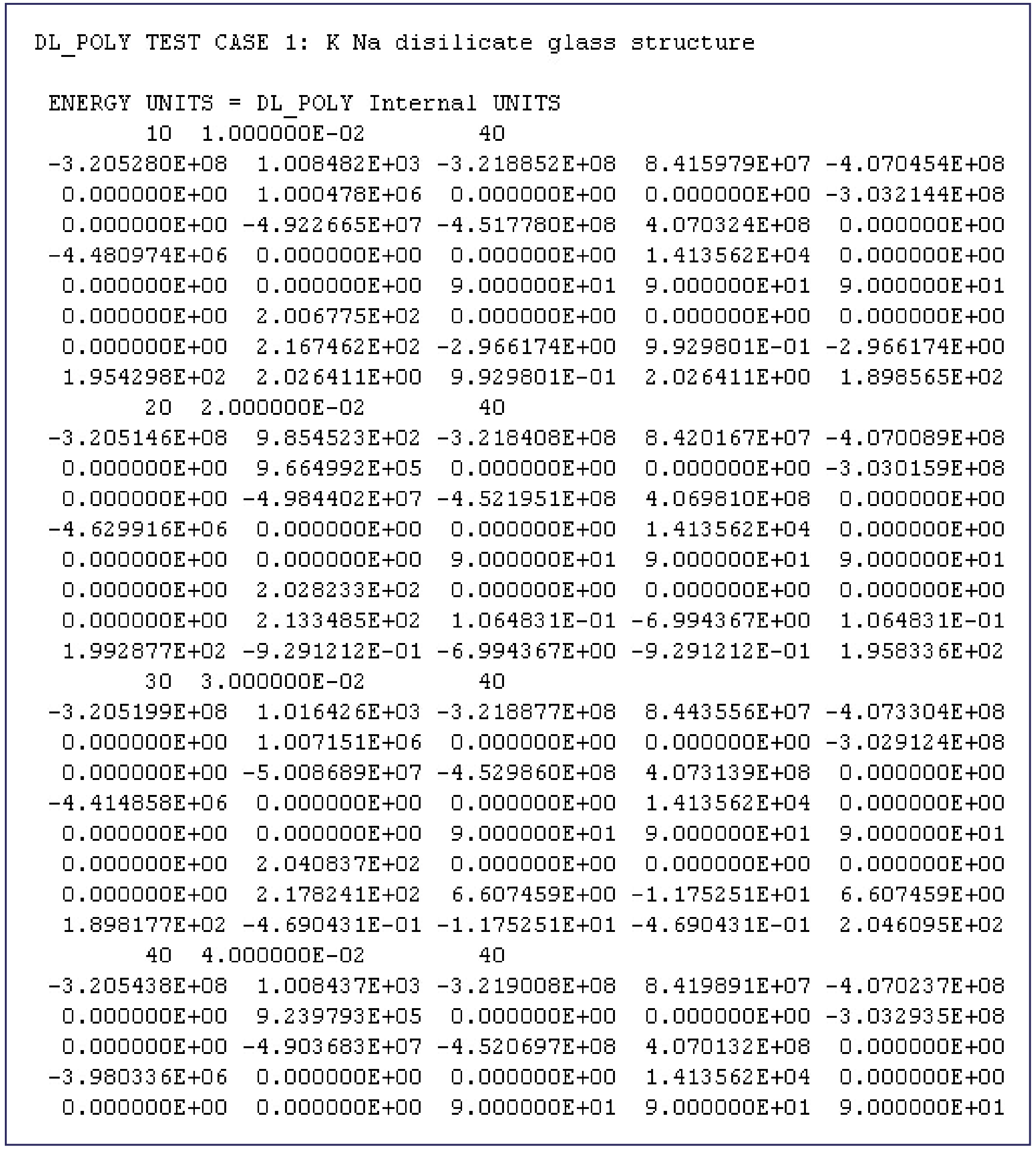
- Provides program restart capability
- File is unformatted (not human readable)
- Contains thermodynamic accumulators, RDF data, MSD data and other checkpoint data
- REVIVE (output file) ---> REVOLD (input file)

OUTPUT File

- Provides Job Summary (mandatory!)
- Formatted to be human readable
- Contents:
  - Summary of input data
  - Instantaneous thermodynamic data at selected intervals
  - Rolling averages of thermodynamic data
  - Statistical averages
  - Final configuration
  - Radial distribution data
  - Estimated mean-square displacements and 3D diffusion coefficient
- Plus:
  - Timing data, CGM and relaxed shell model iteration data
  - Warning & Error reports

STATIS File

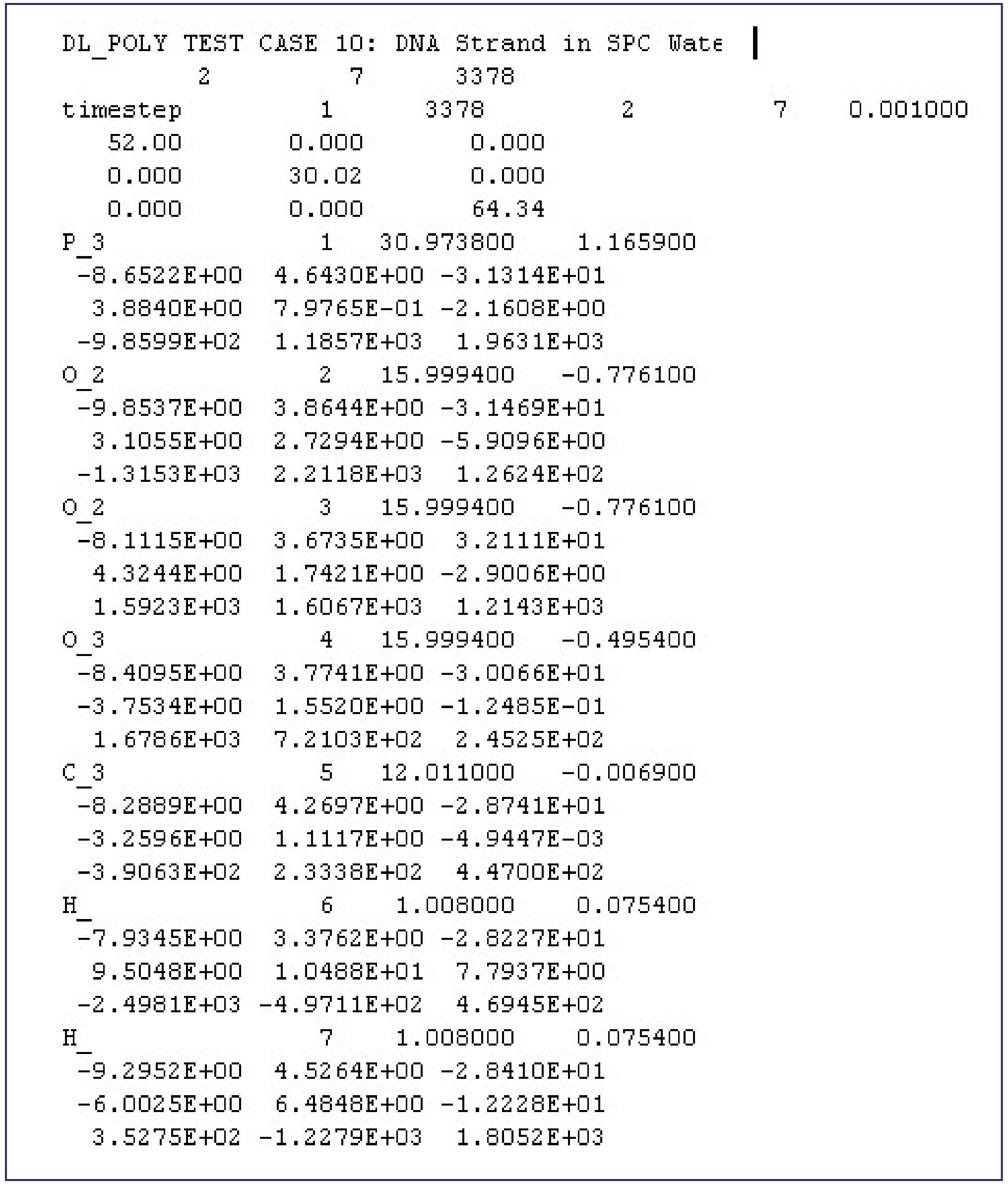
- System properties at intervals selected by user
- Optional
- Formatted (I10,E14)
- Intended use: statistical analysis (e.g. error) and plotting vs. time.
- Recommend use with GUI!
- Header:
  - Title
  - Units
- Data:
  - Time step, time, #entries
  - System data



```
DL_POLY TEST CASE 1: K Na disilicate glass structure

 ENERGY UNITS = DL_POLY Internal UNITS
        10  1.000000E-02        40
 -3.205280E+08  1.008482E+03 -3.218852E+08  8.415979E+07 -4.070454E+08
  0.000000E+00  1.000478E+06  0.000000E+00  0.000000E+00 -3.032144E+08
  0.000000E+00 -4.922665E+07 -4.517780E+08  4.070324E+08  0.000000E+00
 -4.480974E+06  0.000000E+00  0.000000E+00  1.413562E+04  0.000000E+00
  0.000000E+00  0.000000E+00  9.000000E+01  9.000000E+01  9.000000E+01
  0.000000E+00  2.006775E+02  0.000000E+00  0.000000E+00  0.000000E+00
  0.000000E+00  2.167462E+02 -2.966174E+00  9.929801E-01 -2.966174E+00
  1.954298E+02  2.026411E+00  9.929801E-01  2.026411E+00  1.898565E+02
        20  2.000000E-02        40
 -3.205146E+08  9.854523E+02 -3.218408E+08  8.420167E+07 -4.070089E+08
  0.000000E+00  9.664992E+05  0.000000E+00  0.000000E+00 -3.030159E+08
  0.000000E+00 -4.984402E+07 -4.521951E+08  4.069810E+08  0.000000E+00
 -4.629916E+06  0.000000E+00  0.000000E+00  1.413562E+04  0.000000E+00
  0.000000E+00  0.000000E+00  9.000000E+01  9.000000E+01  9.000000E+01
  0.000000E+00  2.028233E+02  0.000000E+00  0.000000E+00  0.000000E+00
  0.000000E+00  2.133485E+02  1.064831E-01 -6.994367E+00  1.064831E-01
  1.992877E+02 -9.291212E-01 -6.994367E+00 -9.291212E-01  1.958336E+02
        30  3.000000E-02        40
 -3.205199E+08  1.016426E+03 -3.218877E+08  8.443556E+07 -4.073304E+08
  0.000000E+00  1.007151E+06  0.000000E+00  0.000000E+00 -3.029124E+08
  0.000000E+00 -5.008689E+07 -4.529860E+08  4.073139E+08  0.000000E+00
 -4.414858E+06  0.000000E+00  0.000000E+00  1.413562E+04  0.000000E+00
  0.000000E+00  0.000000E+00  9.000000E+01  9.000000E+01  9.000000E+01
  0.000000E+00  2.040837E+02  0.000000E+00  0.000000E+00  0.000000E+00
  0.000000E+00  2.178241E+02  6.607459E+00 -1.175251E+01  6.607459E+00
  1.898177E+02 -4.690431E-01 -1.175251E+01 -4.690431E-01  2.046095E+02
        40  4.000000E-02        40
 -3.205438E+08  1.008437E+03 -3.219008E+08  8.419891E+07 -4.070237E+08
  0.000000E+00  9.239793E+05  0.000000E+00  0.000000E+00 -3.032935E+08
  0.000000E+00 -4.903683E+07 -4.520697E+08  4.070132E+08  0.000000E+00
 -3.980336E+06  0.000000E+00  0.000000E+00  1.413562E+04  0.000000E+00
  0.000000E+00  0.000000E+00  9.000000E+01  9.000000E+01  9.000000E+01
```

HISTORY [HISTORF] File

- Configuration data at user selected intervals
  - Formatted
  - Optional
- Header:
  - Title
  - Data level, cell key, number
- Configuration data:
  - Time step and data keys
  - Cell Matrix
  - Atom name, mass, charge
  - X,Y,Z coordinates (level 0)
  - X,Y,Z velocities (level 1)
  - X,Y,Z forces (level 2)



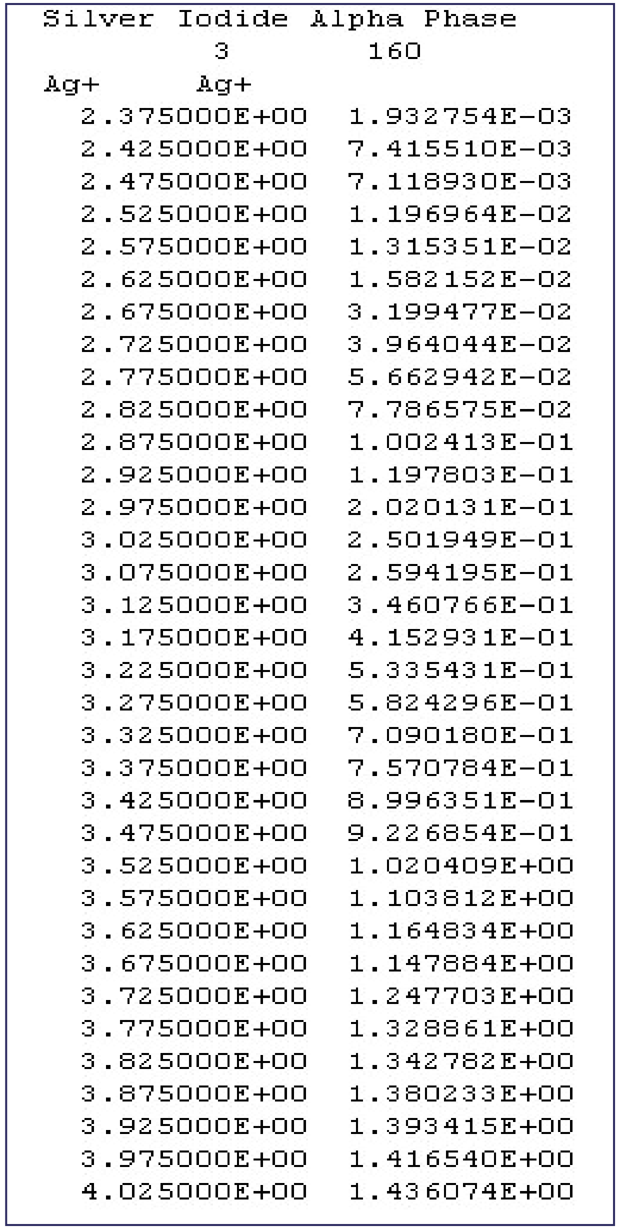
```
DL_POLY TEST CASE 10: DNA Strand in SPC Wate
          2          7       3378
timestep           1       3378          2          7    0.001000
   52.00       0.000       0.000
   0.000       30.02       0.000
   0.000       0.000       64.34
P_3                1   30.973800    1.165900
 -8.6522E+00  4.6430E+00 -3.1314E+01
  3.8840E+00  7.9765E-01 -2.1608E+00
 -9.8599E+02  1.1857E+03  1.9631E+03
O_2                2   15.999400   -0.776100
 -9.8537E+00  3.8644E+00 -3.1469E+01
  3.1055E+00  2.7294E+00 -5.9096E+00
 -1.3153E+03  2.2118E+03  1.2624E+02
O_2                3   15.999400   -0.776100
 -8.1115E+00  3.6735E+00  3.2111E+01
  4.3244E+00  1.7421E+00 -2.9006E+00
  1.5923E+03  1.6067E+03  1.2143E+03
O_3                4   15.999400   -0.495400
 -8.4095E+00  3.7741E+00 -3.0066E+01
 -3.7534E+00  1.5520E+00 -1.2485E-01
  1.6786E+03  7.2103E+02  2.4525E+02
C_3                5   12.011000   -0.006900
 -8.2889E+00  4.2697E+00 -2.8741E+01
 -3.2596E+00  1.1117E+00 -4.9447E-03
 -3.9063E+02  2.3338E+02  4.4700E+02
H_                 6    1.008000    0.075400
 -7.9345E+00  3.3762E+00 -2.8227E+01
  9.5048E+00  1.0488E+01  7.7937E+00
 -2.4981E+03 -4.9711E+02  4.6945E+02
H_                 7    1.008000    0.075400
 -9.2952E+00  4.5264E+00 -2.8410E+01
 -6.0025E+00  6.4848E+00 -1.2228E+01
  3.5275E+02 -1.2279E+03  1.8052E+03
```

Science and Technology Facilities Council

Scientific Computing

RDFDAT [ZDNDAT] File

- Formatted (A8,I10,E14)
- Plotable
- Optional
- RDFs from pair forces
- Header:
  - Title
  - No. plots & length of plot
- RDF data:
  - Atom symbols (2)
  - Radius (A) & RDF
  - Repeated…
- ZDNDAT file has same format



```
Silver Iodide Alpha Phase
         3       160
Ag+      Ag+
    2.375000E+00  1.932754E-03
    2.425000E+00  7.415510E-03
    2.475000E+00  7.118930E-03
    2.525000E+00  1.196964E-02
    2.575000E+00  1.315351E-02
    2.625000E+00  1.582152E-02
    2.675000E+00  3.199477E-02
    2.725000E+00  3.964044E-02
    2.775000E+00  5.662942E-02
    2.825000E+00  7.786575E-02
    2.875000E+00  1.002413E-01
    2.925000E+00  1.197803E-01
    2.975000E+00  2.020131E-01
    3.025000E+00  2.501949E-01
    3.075000E+00  2.594195E-01
    3.125000E+00  3.460766E-01
    3.175000E+00  4.152931E-01
    3.225000E+00  5.335431E-01
    3.275000E+00  5.824296E-01
    3.325000E+00  7.090180E-01
    3.375000E+00  7.570784E-01
    3.425000E+00  8.996351E-01
    3.475000E+00  9.226854E-01
    3.525000E+00  1.020409E+00
    3.575000E+00  1.103812E+00
    3.625000E+00  1.164834E+00
    3.675000E+00  1.147884E+00
    3.725000E+00  1.247703E+00
    3.775000E+00  1.328861E+00
    3.825000E+00  1.342782E+00
    3.875000E+00  1.380233E+00
    3.925000E+00  1.393415E+00
    3.975000E+00  1.416540E+00
    4.025000E+00  1.436074E+00
```

Other Extra Files

- REFERENCE file
  - Reference structure to compare against
- DEFECTS file
  - Trajectory file of vacancies and interstitials migration
- MSDTMP file
  - Trajectory like file containing particles’ Sqrt(MSD_{mean}) and T_{mean}
- RSDDAT file
  - Trajectory like file containing particles’ Sqrt(RSD from origin)
- TAB*INT* file
  - Table file for *INT*ra-molecular interactions
- *INT*DAT file
  - Probability Distribution Functions for *INT*ra-molecular interactions
- HISTORF file
  - Force replayed HISTORY
- …

Science and
Technology
Facilities Council

DL_POLY
Scalable Performance

Science and
Technology
Facilities Council
Scientific Computing

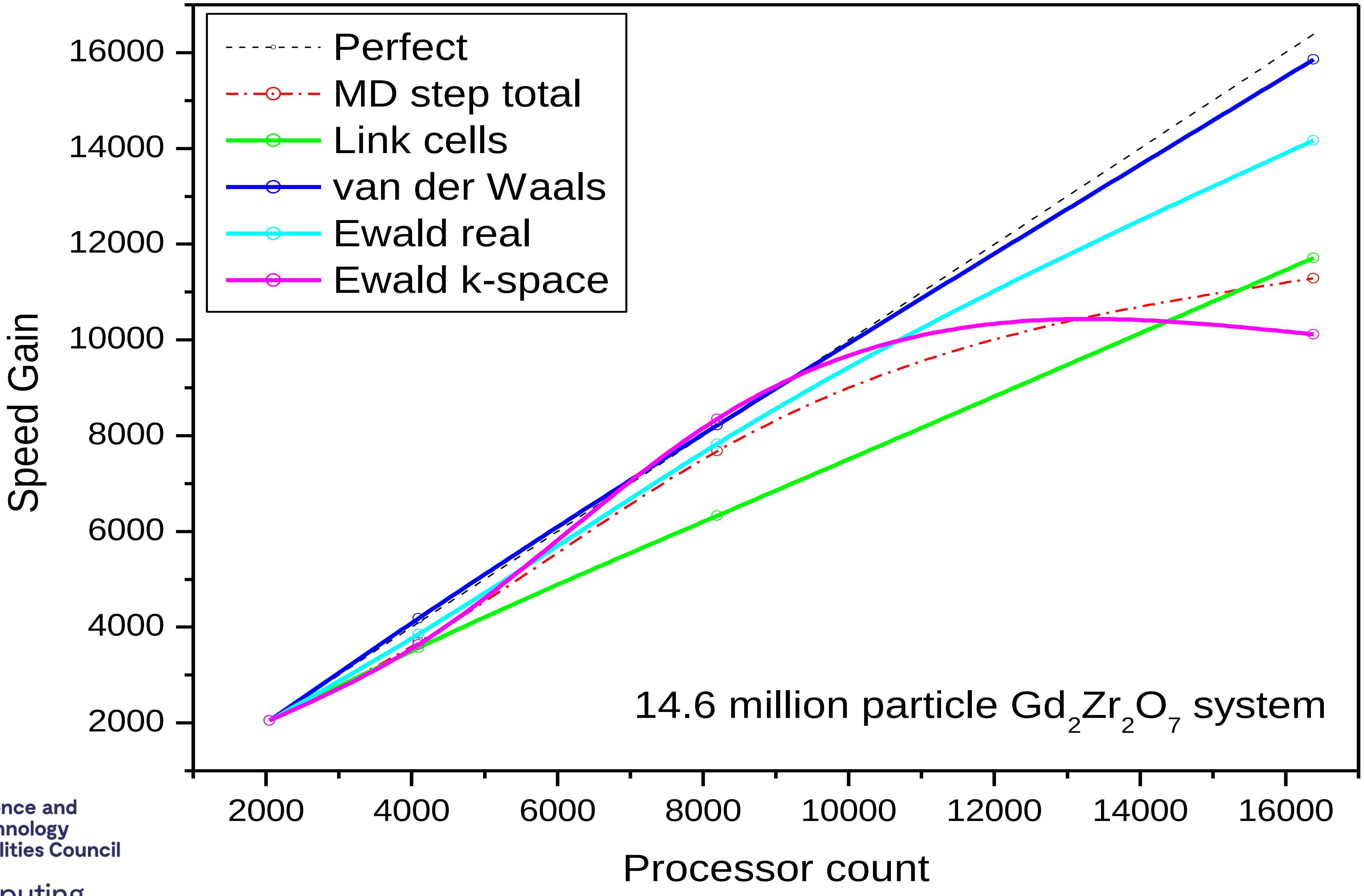
Proof of Concept

300,763,000 NaCl with full SPME electrostatics evaluation on 1024 CPU cores

	IBM p575 2004/5	**Cray XE6 2013/4**
Start-up time	≈ 60 min	≈ 15 min
Timestep time	≈ 68 sec	≈ 23 sec
FFT evaluation	≈ 55 sec	≈ 18 sec

In theory ,the system can be seen by the eye. Although you would need a very good microscope – the MD cell size for this system is $2\mu m$ along the side and as the wavelength of the visible light is $0.5\mu m$ so it should be theoretically possible.

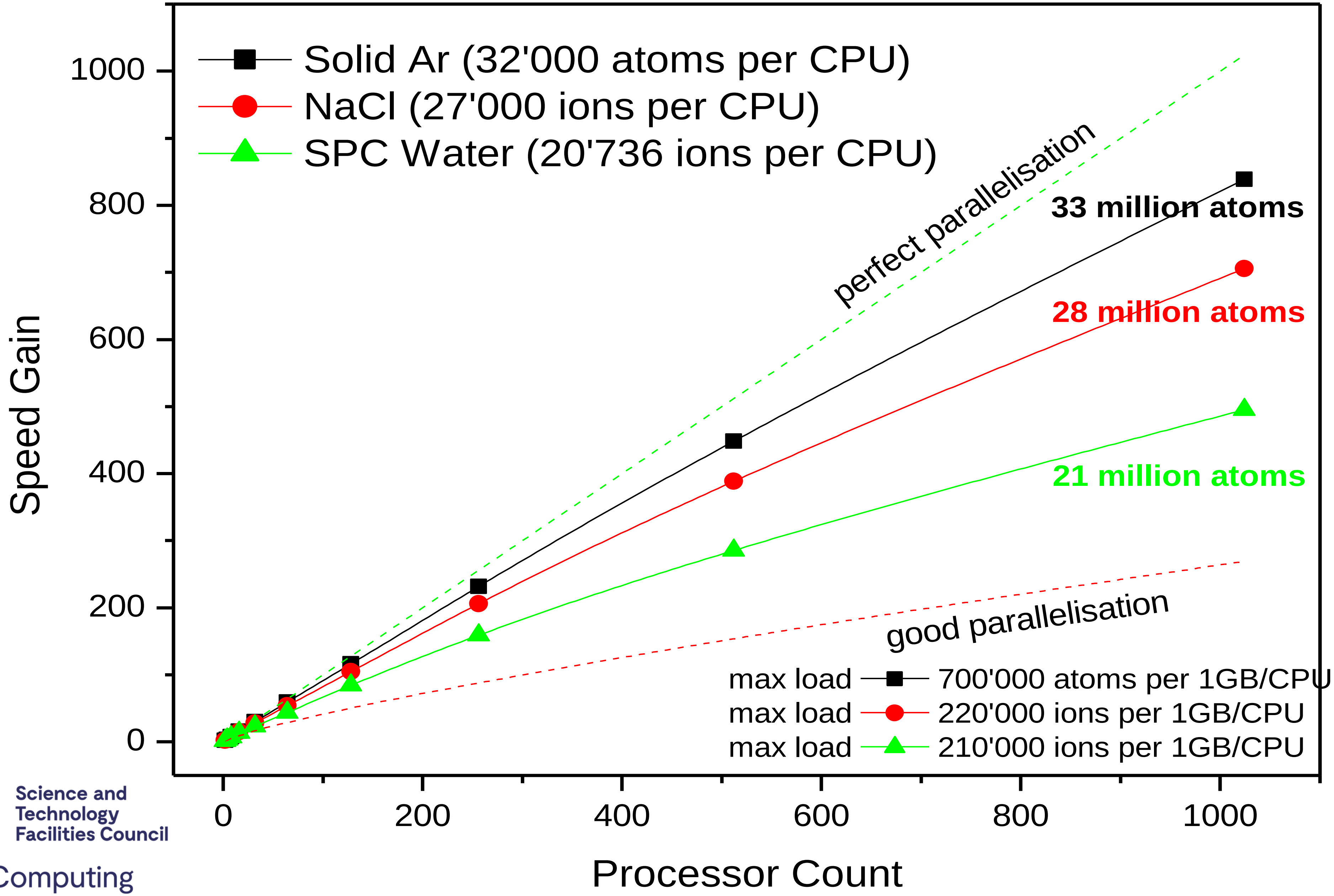
Benchmarking BG/L Jülich 2007



Science and Technology Facilities Council

Scientific Computing

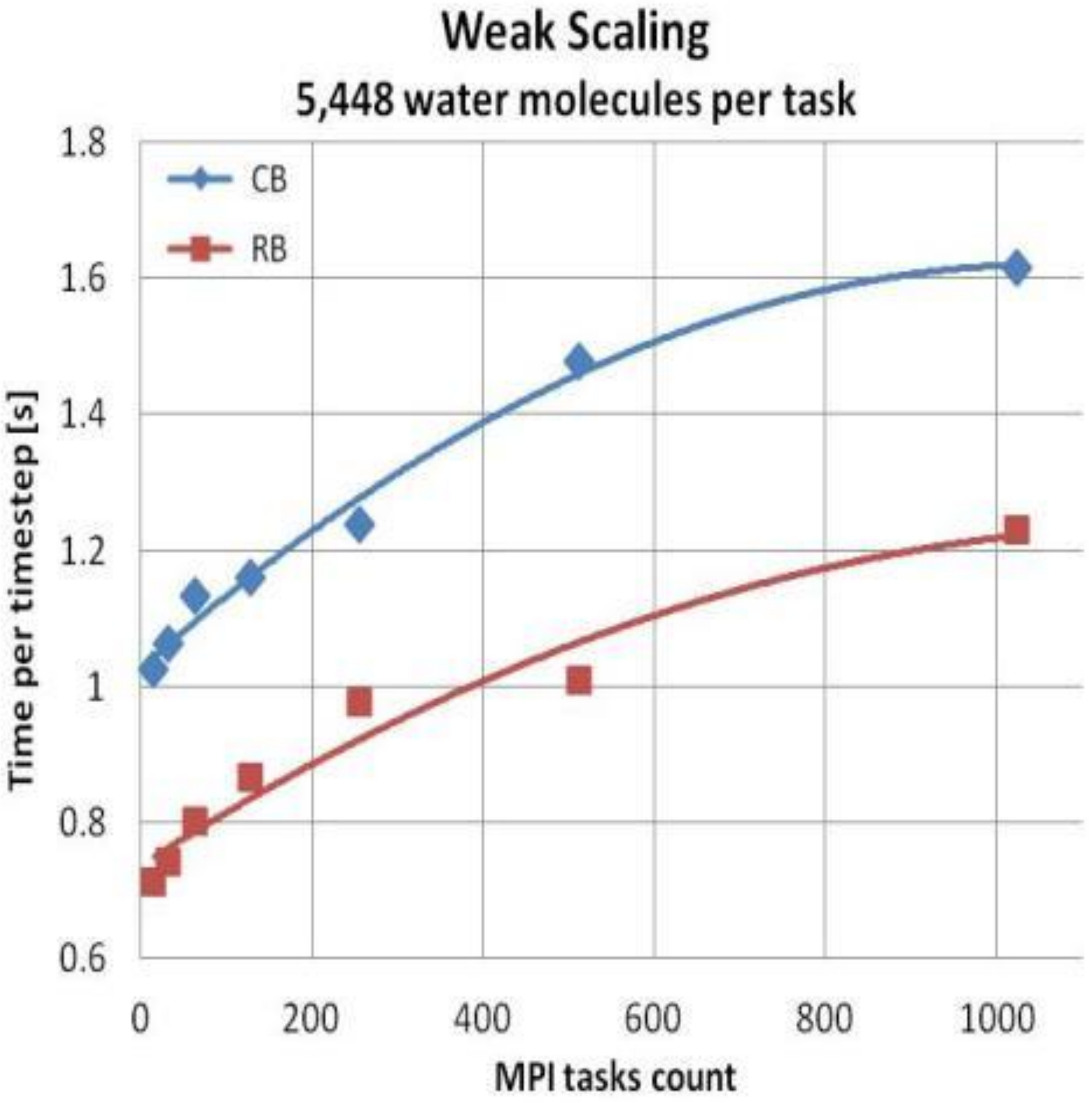
Weak Scaling

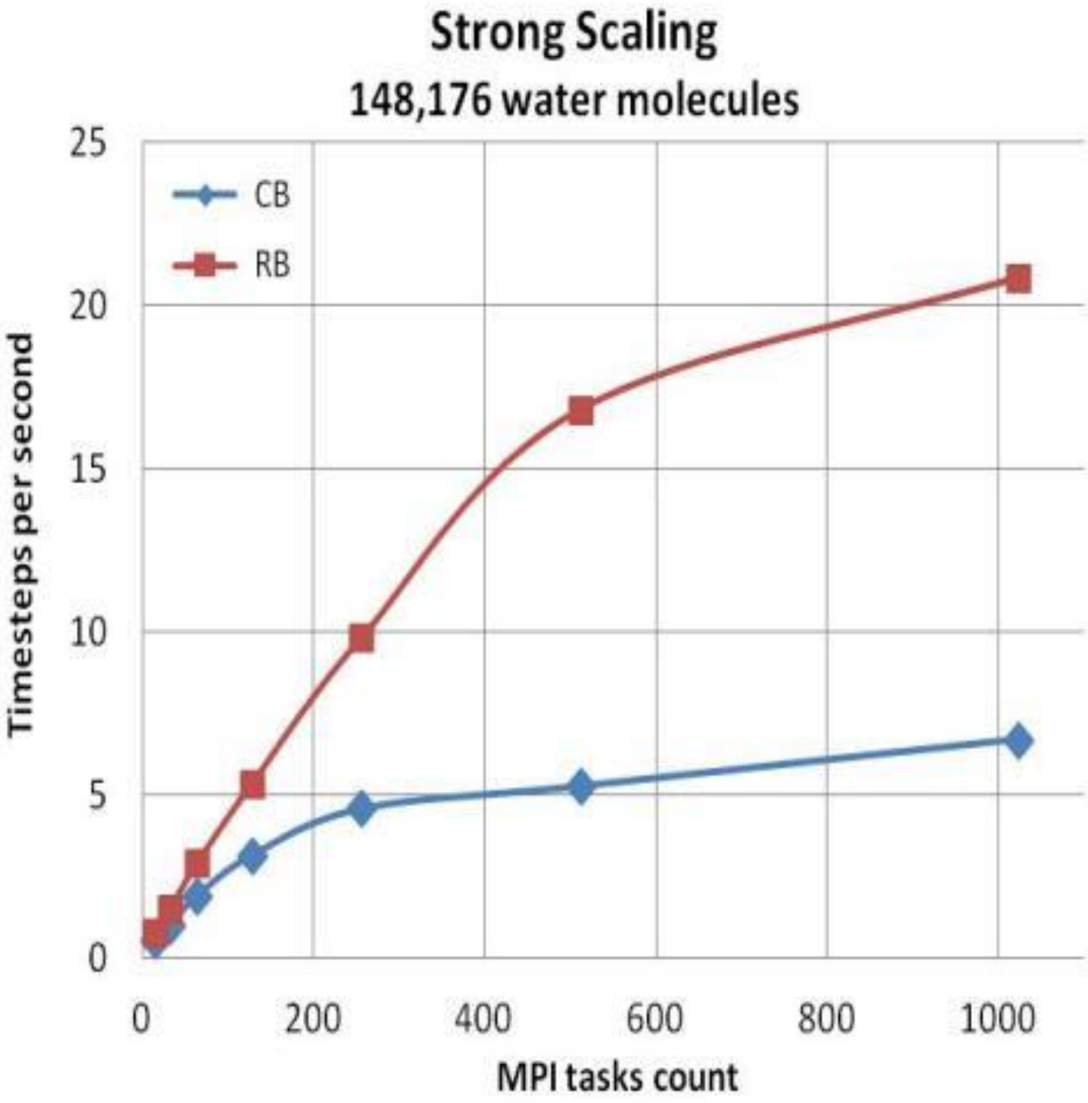


Science and Technology Facilities Council
Scientific Computing

RB v/s CB Performance & Scalability

HECToR (Cray XE6) 2013

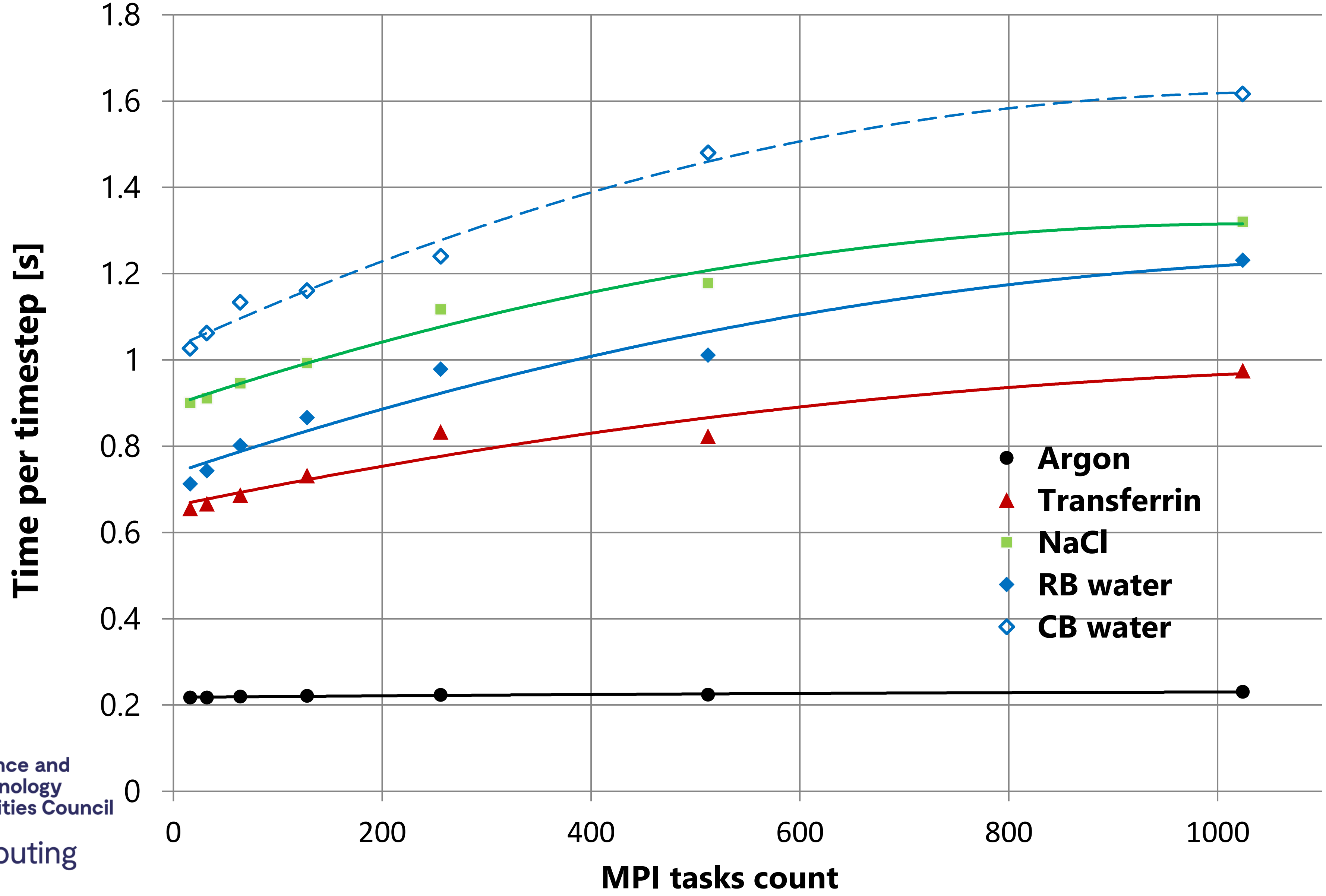




Scie

Weak Scaling and Cost Complexity

HECToR (Cray XE6) 2013



Science and Technology Facilities Council

Scientific Computing

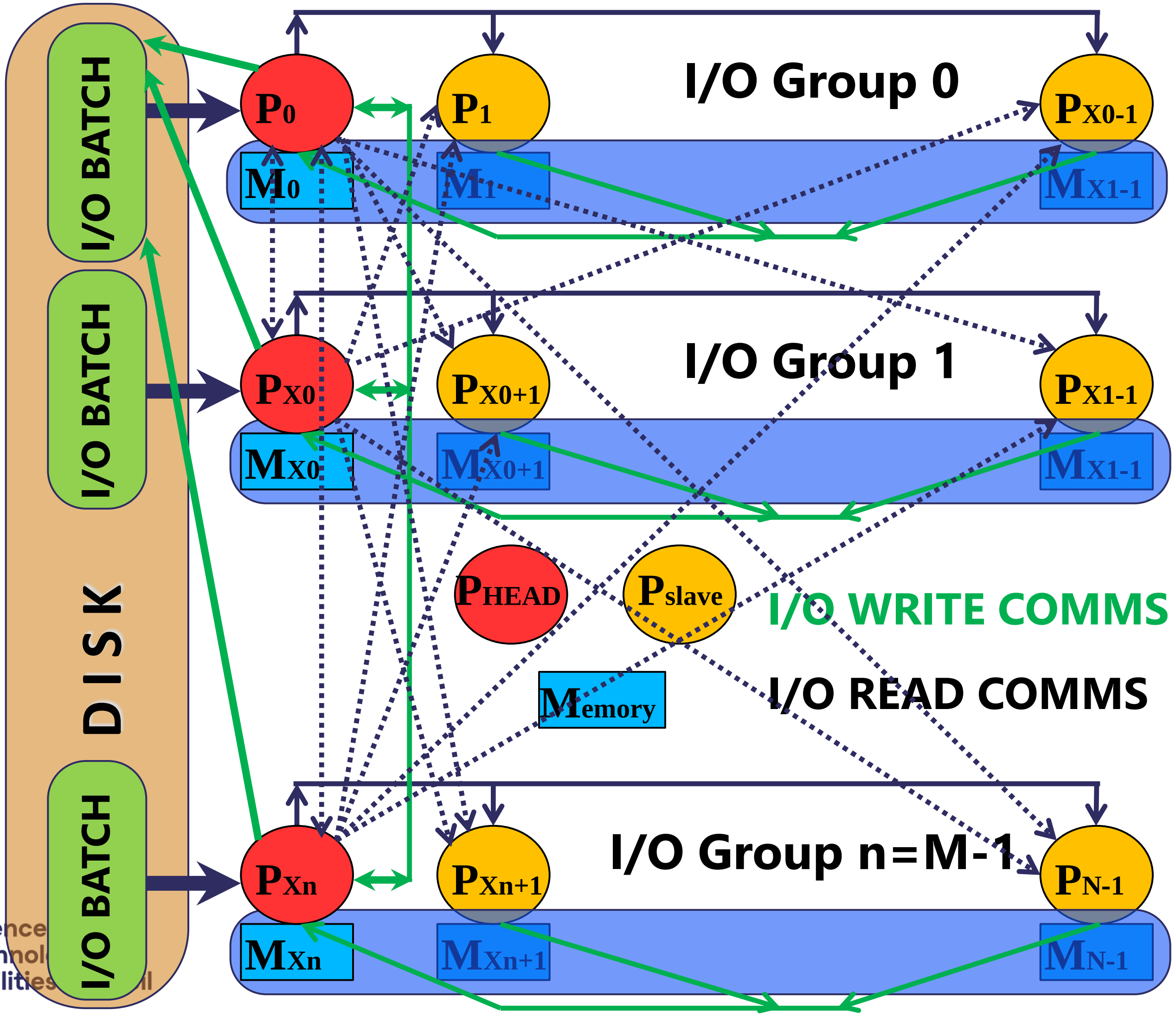
I/O Solutions

1. **Serial read and write (sorted/unsorted)** – where only a single MPI task, the master, handles it all and all the rest communicate in turn to or get broadcasted to while the master completes writing a configuration of the time evolution.
2. **Parallel write via direct access or MPI-I/O (sorted/unsorted)** – where **ALL / SOME** MPI tasks print in the same file in some orderly manner so (no overlapping occurs using Fortran direct access printing. However, it should be noted that the behaviour of this method is not defined by the Fortran standard, and in particular we have experienced problems when disk cache is not coherent with the memory).
3. **Parallel read via MPI-I/O or Fortran**
4. **Serial NetCDF read and write** using NetCDF libraries for machine-independent data formats of array-based, scientific data (widely used by various scientific communities).

The Advanced Parallel I/O Strategy

HECToR (Cray XE6) 2013

N compute cores of which M < N do I/O



- 72 I/O NODES
- READ ~ 50-300 Mbyte/s with best performance on 16 to 128 I/O Groups
- WRITE ~ 50-150 Mbyte/s with best performance on 64 to 512 I/O Groups
- Performance depends on user defined number of I/O groups, and I/O batch (memory CPU to disk) and buffer (memory of comms transactions between CPUs)
- Reasonable defaults as a function of all MPI tasks are provided

Science and
Technology
Facilities Council

Scientific Computing

DL_POLY Compiling, Running and Labs

https://dl-sdg.github.io/

Science and
Technology
Facilities Council

Scientific Computing

Science and Technology Facilities Council

Scientific Computing

Professor Kostyantyn Trachenko

1st February 1971 – 20th April 2025

Science and
Technology
Facilities Council

Scientific Computing

Thank you

scd.stfc.ac.uk

@SciComp_STFC

Science and
Technology
Facilities Council

DL_Software and Dissipative Particle Dynamics (DPD)

Michael Seaton
UKRI STFC Daresbury Laboratory
michael.seaton@stfc.ac.uk

archer2

DL SOFTWARE
Enabling Discovery

CCP5
Collaborative
Computational Project
Computer Simulation of Condensed Phases

UK COMES
Consortium on Mesoscale Engineering Sciences

Science and
Technology
Facilities Council

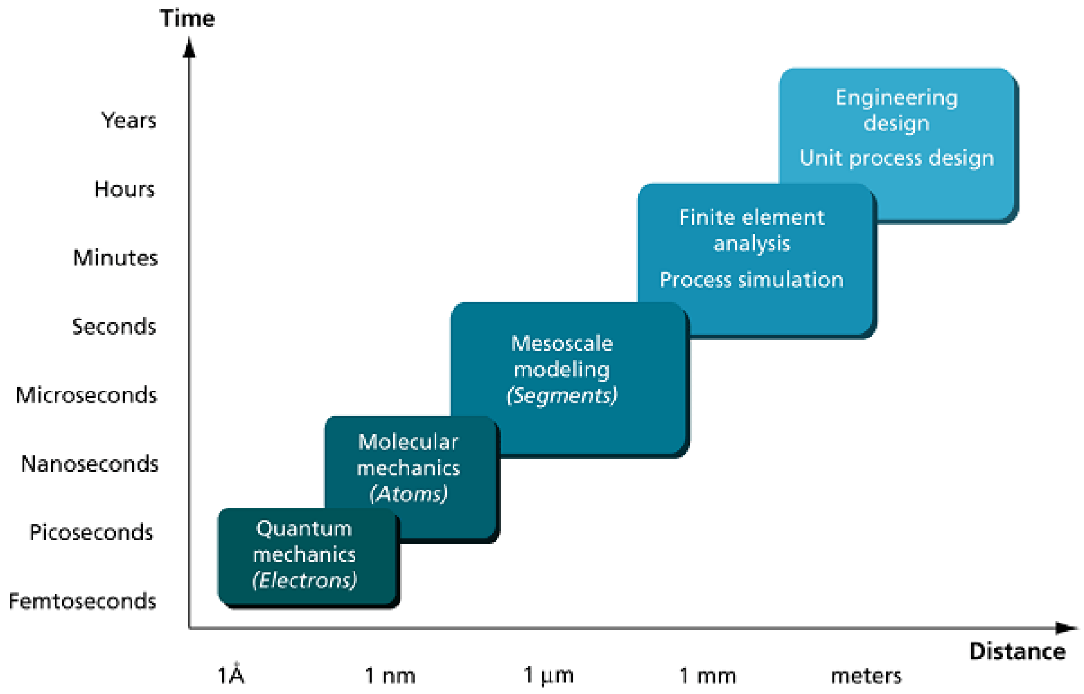
The mesoscale

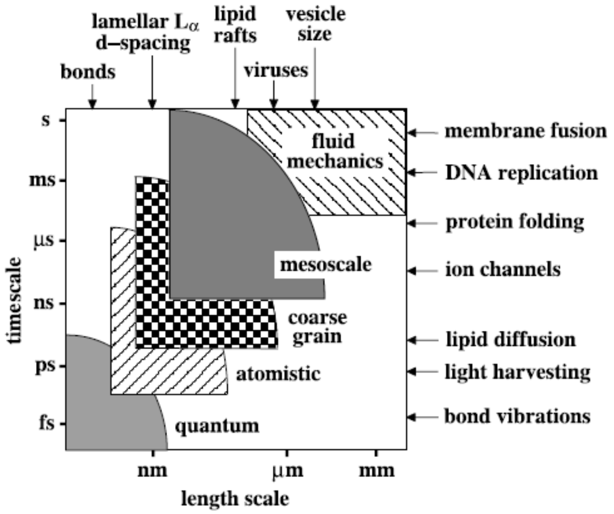
‘If life is going to exist in a Universe of this size, then the one thing it cannot afford to have is a sense of proportion.’
- Douglas Adams, *The Restaurant at the End of the Universe*

Getting a sense of (middle) scale

Mesoscale

- Bridges a ‘hinterland’ between atomistic and continuum scales
  - From 10nm+, 1ns+
  - Upper limit depends on computing capability
- Many interesting scientific applications operate at mesoscale
  - Approaching engineering (process) scales





- Correct thermodynamics <u>and</u> hydrodynamics needed
  - Parameterisation may involve bottom-up (microscopic) <u>and</u> top-down (macroscopic) approaches
- Models often involve bigger-than-atom particles (‘beads’)

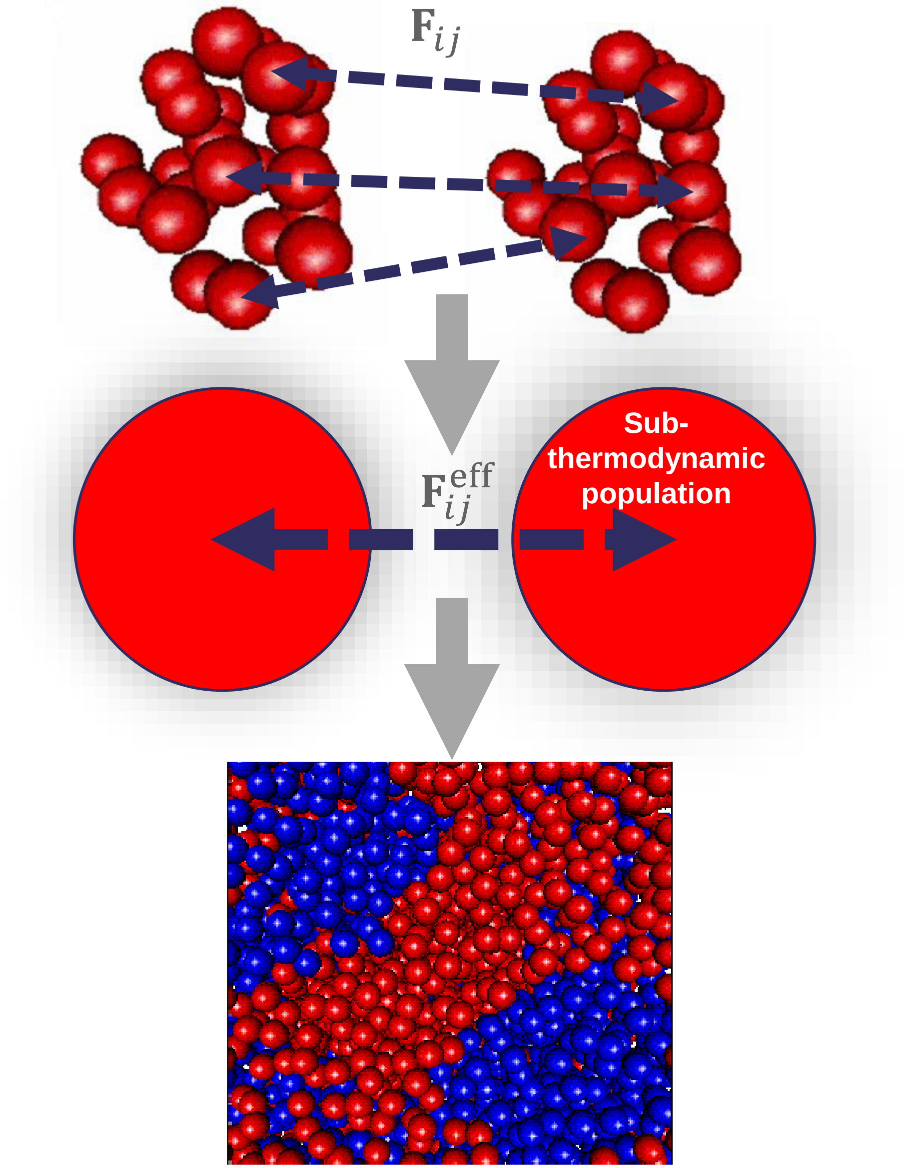
Mesoscale modelling approaches

Bottom-up: from the microscale

- Coarse-grain atoms/molecules into beads as ‘sub-thermodynamic populations’
- Obtain effective interactions between beads to represent those between atoms

Top-down: from the macroscale

- Reverse-engineer simulation rules to give required behaviours
- Beads are ‘carriers of momentum’: mesoscopic representation of continuum fluids
  - Direct connection to atoms/molecules not needed



Science and
Technology
Facilities Council

Dissipative Particle Dynamics (DPD)

‘The world is a thing of utter inordinate complexity and richness and strangeness that is absolutely awesome. I mean, the idea that such complexity can arise not only out of such simplicity, but probably absolutely out of nothing, is the most fabulous extraordinary idea.’
- Douglas Adams

Dissipative Particle Dynamics (DPD)

From molecular to mesoscopic dynamics

- Modelling condensed phase systems using particles (‘beads’) with pairwise potentials
- Integrate forces on particles to move them around

Basic DPD algorithm very similar to classical MD: can use MD code to carry out DPD calculations without much modification

DPD is actually two things …

- Heat bath coupling using additional pairwise forces: **DPD thermostat**
- Soft (usually) repulsive pair potentials and forces to represent bead interactions at mesoscale: **DPD interactions**

Pairwise thermostatting provides Galilean invariance: can control temperature without disrupting flows

Can use DPD thermostat and DPD interactions separately, but usually used together for mesoscopic simulations

Science and Technology Facilities Council

DPD thermostat: pairwise Langevin dynamics

Two pairwise forces working in tandem[1]

- Dissipative force: $\mathbf{F}_{ij}^{D} = -\gamma w^{D}(r_{ij})(\hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij})\hat{\mathbf{r}}_{ij}$
- Random force: $\mathbf{F}_{ij}^{R} = \sigma w^{R}(r_{ij})\theta_{ij}\hat{\mathbf{r}}_{ij}$

$$\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$$
$$\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$$
$$\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$$
$$\langle \theta_{ij}(t) \rangle = 0$$
$$\langle \theta_{ij}(t)\theta_{i'j'}(t') \rangle = (\delta_{ii'}\delta_{jj'} + \delta_{ij'}\delta_{ji'})\delta(t - t')$$

- Fokker-Planck fluctuation-dissipation theory – obtain thermal equilibrium when:

$$w^{D}(r_{ij}) = \left(w^{R}(r_{ij})\right)^{2}$$
$$\sigma^{2} = 2k_{B}T\gamma$$

Under these conditions, $\mathbf{F}_{ij}^{D}$ and $\mathbf{F}_{ij}^{R}$ control system temperature

- Can also make connection to Navier-Stokes equations[2,3]
  - Dissipative force parameter γ and switching function w^{D} related to viscosity and diffusivity

1. Español and Warren, *EPL* **30** (4), 191–196 (1995)
2. Marsh *et al.*, *Phys Rev E* **56** (2), 1676–1691 (1997)
3. Visser *et al.*, *J Chem Phys* **214**, 491–504 (2006)

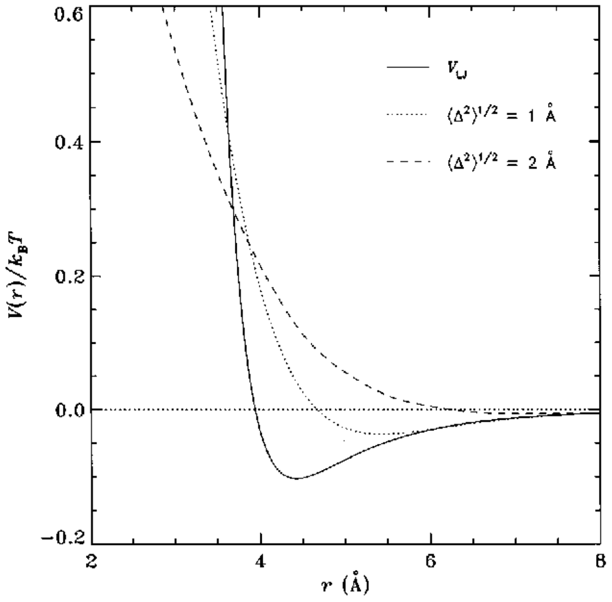
Summary: DPD gives correct hydrodynamic behaviour

DPD interactions: how should beads interact?

Simplest possible force/potential[1]

- No real restrictions on how beads should interact, but ideally want something representative at mesoscale
  - As accurately as we can with as few particles as possible, simple to speed up calculations
- Coarse-graining polymer melts gives pairwise potentials that differ markedly from atomistic forms[2,3]
- Groot and Warren chose:

$$\mathbf{F}_{ij}^{C} = \begin{cases} A\left(1 - \frac{r_{ij}}{r_c}\right)\hat{\mathbf{r}}_{ij}, & r_{ij} < r_c \\ 0, & r_{ij} \geq r_c \end{cases}$$



Equivalent potential:

$$V(r_{ij}) = \frac{1}{2}Ar_c\left(1 - \frac{r_{ij}}{r_c}\right)^2 \text{ for } r_{ij} < r_c$$

- Finite ranged, bounded
- No divergence for $r_{ij} \rightarrow 0$
- Entirely repulsive (no attraction)

1. Groot and Warren, *J Chem Phys* **107** (11), 4423–4435 (1997)
2. Forrest and Suter, *J Chem Phys* **102** (18), 7256–504 (1995)
3. Flekkøy and Coveney, *PRL* **83** (9), 1775–1778 (1999)

DPD interactions: how should beads interact?

Simplest possible force/potential[1]

$$\mathbf{F}_{ij}^{C} = \begin{cases} A\left(1 - \frac{r_{ij}}{r_c}\right)\hat{\mathbf{r}}_{ij}, & r_{ij} < r_c \\ 0, & r_{ij} \geq r_c \end{cases}$$

- Above choice of conservative interaction force gives quadratic equation of state (EOS):

$$p \approx \rho k_B T + 0.101 A r_c^4 \rho^2$$

  - Applicable for $\rho r_c^3 > 2$, when radial distribution functions $g(r)$ do not change with ρ
  - Provides some (limited) non-ideal behaviour

Parameterisation mainly for A (repulsion strength)

- Can use EOS to fit isothermal compressibility for each component

$$\left.\frac{\partial p}{\partial \rho}\right|_T = \frac{V_{bead}}{\kappa_T} \approx k_B T + 0.202 A r_c^4 \rho$$

- A between components can be related to mixing energies, e.g. Groot and Warren's connection to Flory-Huggins solution theory[1]

$$\chi^{AB} \propto \left(A^{AB} - A^{AA}\right)$$

Science and Technology Facilities Council

1. Groot and Warren, *J Chem Phys* **107** (11), 4423–4435 (1997)

Using DPD for mesoscale modelling

Can take similar approach to classical MD

- Algorithm described as pairwise interaction forces within a cutoff
  - Thermostatting 'for free'!
  - No velocity field sampling required to use DPD thermostat with flow fields (cf. neMD)
- Mass (m) and length (r_c) scales from bead contents, energy scale ($k_B T$) from system temperature
- Time scale often given as $\tau = r_c \sqrt{\frac{m}{k_B T}}$

Example – water with 3 molecules per bead at room temperature (298 K), bead density $\rho = 3r_c^{-3}$

$m \approx 9.00 \times 10^{-26}$ kg (54.046 u)
$r_c \approx 6.46 \times 10^{-10}$ m (0.646 nm)
$k_B T \approx 4.12 \times 10^{-21}$ J (2.48 kJ mol^{-1})
$\tau \approx 2.50 \times 10^{-12}$ s (2.50 ps)
$A \approx 78 k_B T / r_c$

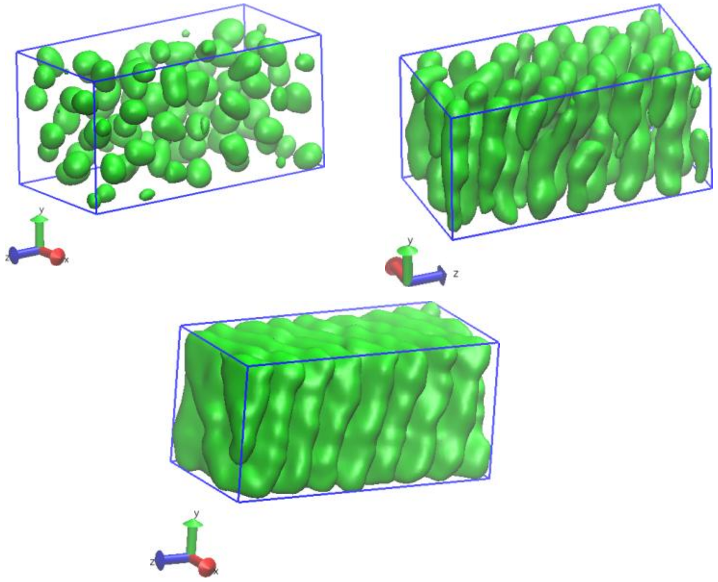
- Can add springs to connect beads together for larger molecules

Bonds not normally 'chemical' as DPD beads often quite large: DPD interactions between connected beads not excluded

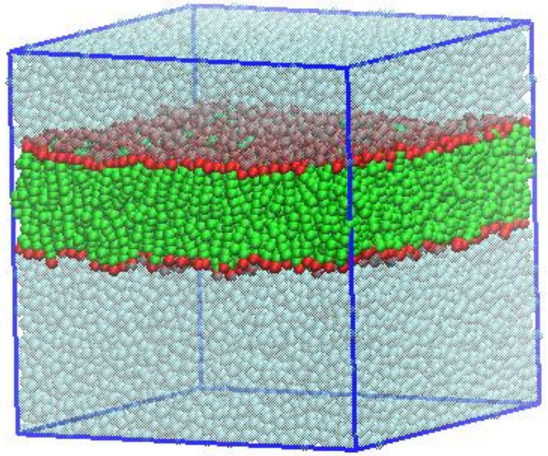
- Soft repulsive interactions simplify initial simulation setup: no numerical issues if beads overlap

Using DPD for mesoscale modelling

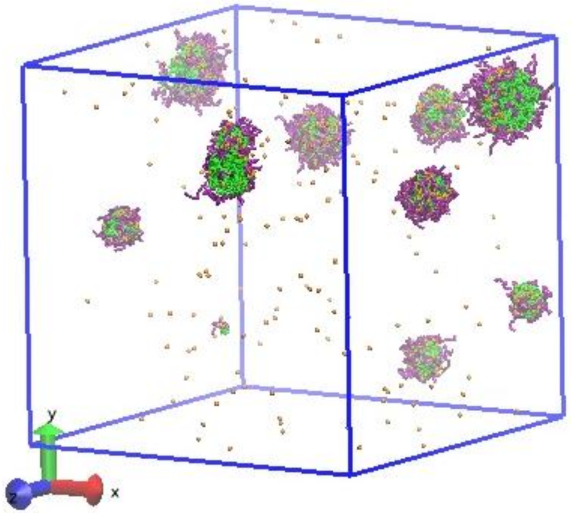
Structure formation by surfactants/lipids/copolymers



Mesophases formed from amphiphilic dimers[1] (two-bead surfactants) of different concentrations in solvent



Bilayer spontaneously formed from solution of liposomes[2]



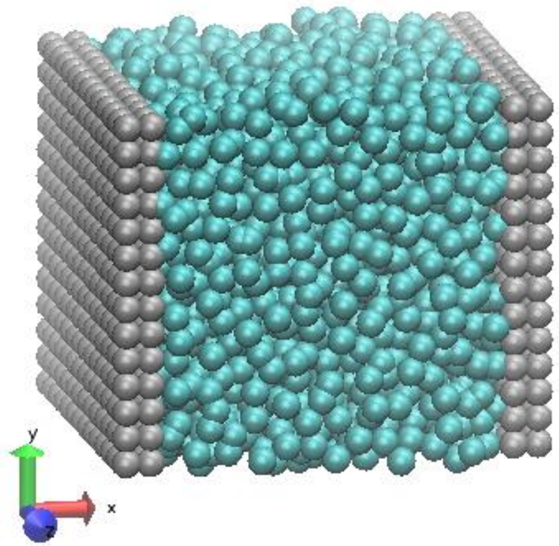
Formation and destruction of copolymer vesicles (based on pH) for drug loading/delivery[3]: A obtained from χ values calculated using atomistic MD

Science and
Technology
Facilities Council

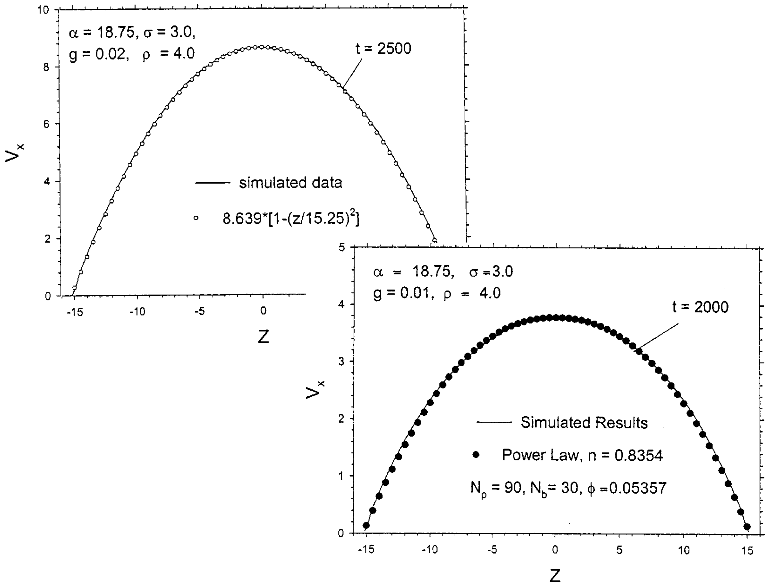
1. Prinsen *et al.*, *PRL* **89** (14), 148302 (2002)
2. Shillcock and Lipowsky, *J Chem Phys* **117** (10), 5048–5061 (2002)
3. Luo and Jiang, *J Control Release* **162** (1), 185–193 (2012)

Using DPD for mesoscale modelling

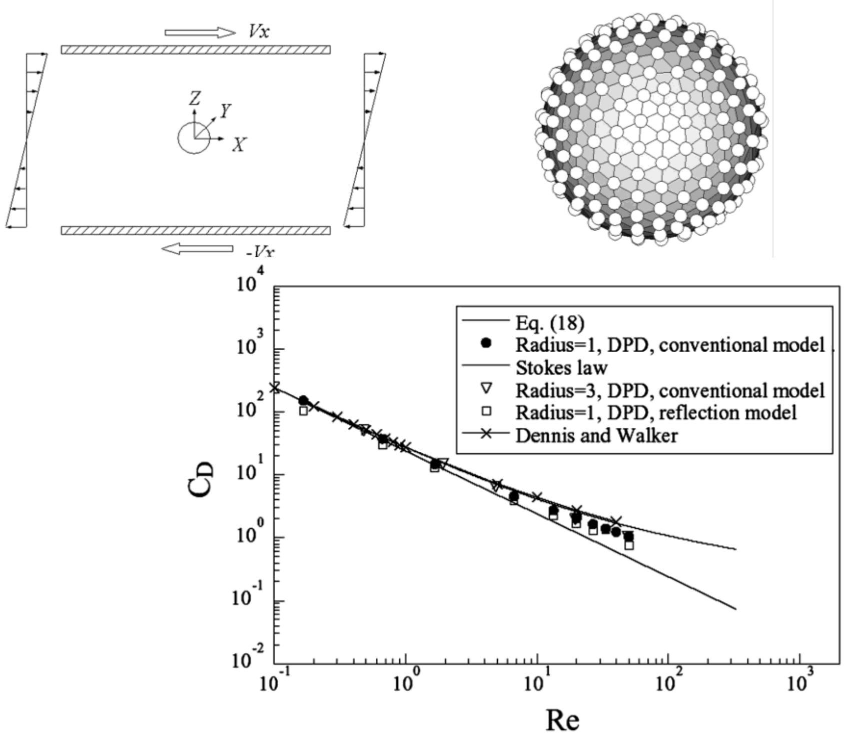
Systems with fluid flows



Pressure-driven (Poiseuille) flow of fluid between frozen bead plates



Velocity profiles of Poiseuille flow for simple fluid (top) and dilute suspension of molecules with FENE bonds (bottom)[1]



Linear shear flow past stationary sphere (consisting of frozen beads) constrained by plates: comparison of drag forces with experiments/Stokes' law[2]

Science and Technology Facilities Council

1. Fan *et al.*, *Phys Fluids* **15** (1), 11–21 (2003)
2. Chen *et al.*, *Phys Fluids* **18** (10), 103605 (2006)

DPD challenges: Parameterisation

- Unlike atomistic and CG-MD, currently no* ‘standard’ force fields (sets of interaction parameters) available for DPD simulations
- Different methods available to get hold of A between components:
  - Flory-Huggins solution theory via χ-parameters[1]
  - Infinite dilution activity coefficients (γ_i^∞)[2]
  - Fitting to water/octanol partition coefficients ($\log P$)[3]
  - Ab initio (DFT) calculations[4]
- Dissipative force parameters γ harder to obtain, but could try:
  - Fitting to diffusivities or velocity autocorrelation functions
  - Ensemble-averaging forces/velocities from MD calculations[6]

* Attempts at DPD FFs for surfactants[3] and phospholipids[5] available

If flows do not need to be quantified, can just choose γ to give decent temperature control and quick equilibration

1. Groot and Warren, *J Chem Phys* **107** (11), 4423–4435 (1997)
2. Vishnyakov *et al.*, *J Phys Chem Lett* **4** (5), 797–802 (2013)
3. Anderson *et al.*, *J Chem Phys* **147** (9), 094503 (2017)
4. Sepehr and Paddison, *Chem Phys Lett* **645**, 20–26 (2016)
5. Wan *et al.*, *PLOS ONE* **13** (5), 1–31 (2018)
6. Sokhan and Todorov, *Mol Simul* **47** (2–3), 248–256 (2021)

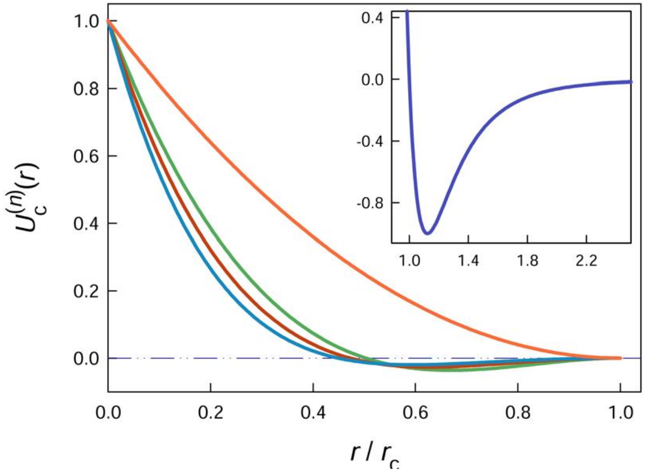
DPD challenges: Correct thermodynamics

- Usual (Groot-Warren) interactions provide limited thermodynamics, e.g. no stable phase coexistence
- Alternative pairwise interactions exist to provide more realistic behaviours, e.g.
  - Exponential function for colloids, gas/liquid[1]

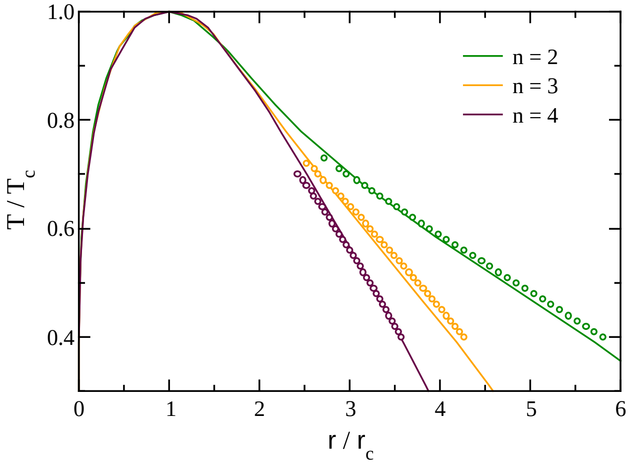
$$\mathbf{F}_{ij}^{C} = \frac{A}{1-e^{-b}}\left(e_{ij}^{-br_{ij}/r_c} - e^{-b}\right)\hat{\mathbf{r}}_{ij}$$

  - Extended Groot-Warren with attraction (nDPD)[2]

$$\mathbf{F}_{ij}^{C} = A\left[b\left(1-\frac{r_{ij}}{r_c}\right)^{n} - \left(1-\frac{r_{ij}}{r_c}\right)\right]\hat{\mathbf{r}}_{ij}$$



nDPD potential form (above) and resulting vapour-liquid coexistence (below)



Science and Technology Facilities Council

1. Pan *et al.*, *Langmuir* **26** (1), 133–142 (2010)
2. Sokhan *et al.*, *Soft Matter* **19** (30), 5824–5834 (2021)

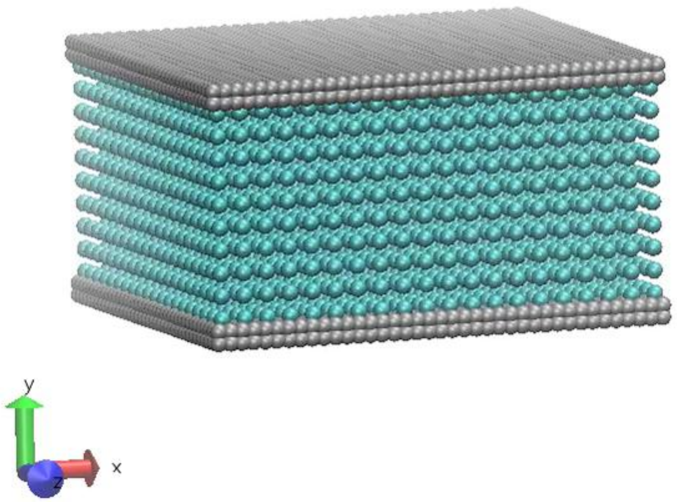
DPD challenges: Correct thermodynamics

- Also possible to <u>impose</u> required free energy density (or EOS) using **many-body DPD** approach[1,2]
  - Define localised density as sum of pairwise weighting functions, $\tilde{\rho}_i = \sum_{j \neq i} w(r_{ij})$
  - Pairwise conservative force given as functions of excess free energy $\psi^{ex}(\rho)$:

$$\mathbf{F}_{ij}^{C} = -\left[\frac{\partial \psi^{ex}}{\partial \rho}(\tilde{\rho}_i) + \frac{\partial \psi^{ex}}{\partial \rho}(\tilde{\rho}_j)\right]\frac{dw}{dr_{ij}}(r_{ij})\hat{\mathbf{r}}_{ij}$$

  - Simple form gives vapour-liquid coexistence[3] (cubic EOS), can be extended for e.g. solid-liquid systems[4]

Similar approach to Embedded Atom Models (EAMs) for metal potentials

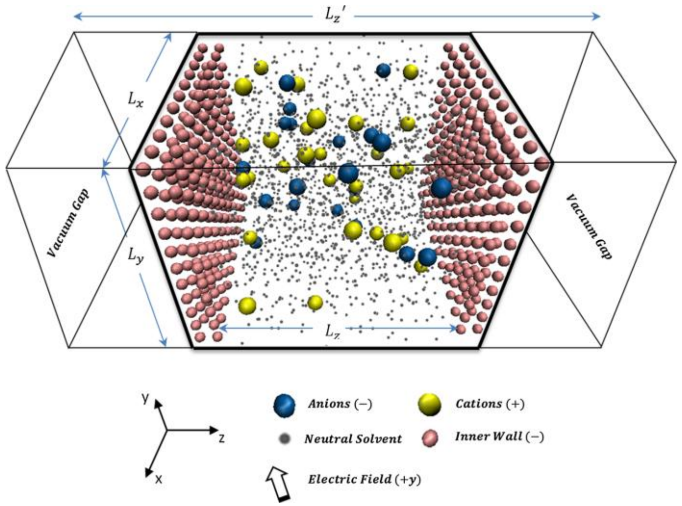


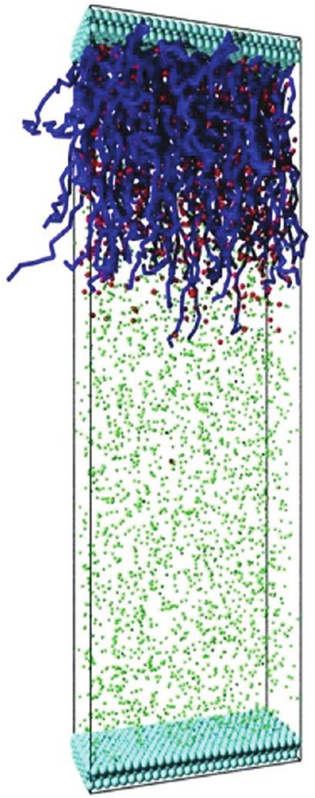
Courtesy: Erik Johansson,
Department of Energy Sciences,
Lund University

1. Pagonabarraga and Frenkel, *J Chem Phys* **115** (11), 5015–5026 (2001)
2. Trofimov *et al.*, *J Chem Phys* **117** (20), 9383–9394 (2002)
3. Warren, *PRE* **68** (6), 066702 (2003)
4. Vanya and Elliot, *PRE* **102** (1), 013312 (2020)

DPD challenges: Electrostatics

- Long-range charge interactions (sometimes) unavoidable!
  - Hydrophilic head groups in surfactants/lipids, electroosmotic flows[1], (poly)electrolytes[2] etc.
- Can use similar approaches to MD calculations (e.g. P^3M[3] and Ewald sum[4,5] methods)
- Not always possible to use point charges with DPD: attractions might overwhelm soft repulsions[6]
  - Often modify Coulombic potential to smear out charges over finite volumes





1. Moshfegh and Jabbarzadeh, *Microfluid Nanofluid* **20** (4), 1–17 (2016)
2. Ibergay *et al.*, *J Chem Theory Comput* **5** (12), 3245–3259 (2009)
3. Groot, *J Chem Phys* **118** (24), 11265–11277 (2003)
4. González-Melchor *et al.*, *J Chem Phys* **125** (22), 224107 (2009)
5. Warren *et al.*, *J Chem Phys* **138** (20), 204907 (2013)
6. Terrón-Mejia *et al.*, *J Phys: Condens Matter* **28** (42), 425101 (2016)

Science and
Technology
Facilities Council

DPD challenges: Thermostats and non-isochoric processes

- Simplest integration of DPD thermostat forces less accurate for higher Δt
  - More sophisticated integration of $\mathbf{F}_{ij}^{D}$ and $\mathbf{F}_{ij}^{R}$[1] (separately to $\mathbf{F}_{ij}^{C}$) helps
- Low Schmidt numbers ($Sc = \nu/D$): difficult to get viscosities for liquids
  - Choose w^{D} to increase viscosity
  - Alternative pairwise thermostats designed for higher Sc, e.g. pairwise form of Andersen thermostat[2]
- Can couple barostat into system alongside DPD thermostat[3] to control pressure, surface area, interfacial tension etc.

1. Shardlow, *SIAM J Sci Comput* **24** (4), 1267–1282 (2001)
2. Lowe, *EPL* **47** (2), 145–151 (1999)
3. Jakobsen, *J Chem Phys* **122** (12), 124901 (2005)

More information about DPD

- Knowledge Centre in DL_Software Digital Guide:

https://dl-sdg.github.io/RESOURCES/knowledge.html

  - Includes pages on coarse-graining, DPD and further details (how to set scales in DPD calculations, parameterisation approaches etc.)

Science and
Technology
Facilities Council

1. Shardlow, *SIAM J Sci Comput* **24** (4), 1267–1282 (2001)
2. Lowe, *EPL* **47** (2), 145–151 (1999)
3. Jakobsen, *J Chem Phys* **122** (12), 124901 (2005)

DPD in DL_Software: DL_MESO_DPD and DL_POLY_5

‘We are stuck with technology when what we really want is just stuff that works.’
- Douglas Adams, *The Salmon of Doubt*

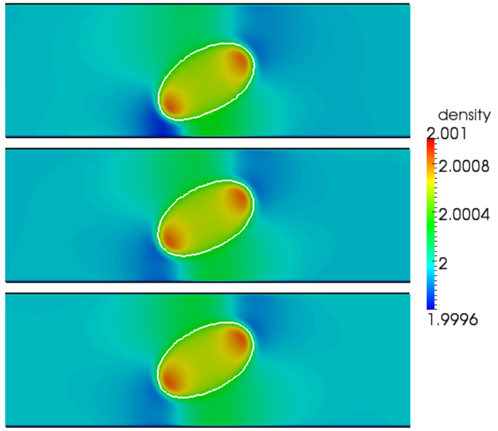
Two DPD simulation codes

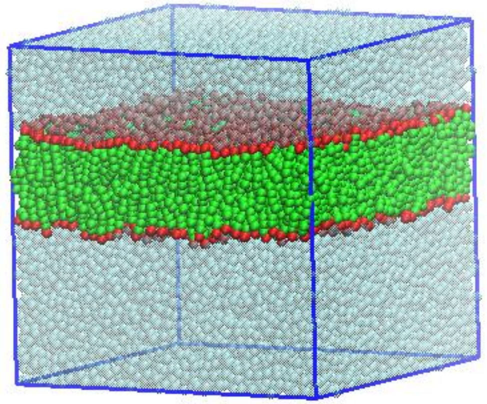
- Bespoke code for DPD calculations: **DL_MESO_DPD**
  - DPD code in DL_MESO (general-purpose mesoscale simulation package)
- An MD code that can also do DPD calculations: **DL_POLY_5**
  - General-purpose MD code with DPD functionalities
- Both codes are similar ‘under the hood’ and do many of the same things
  - Derived from the same ‘ancestor code’ (MDMEGA: CCP5 Program Library)
  - Use MPI-based domain decomposition for parallel running
  - Some interoperability due to similar input file formats

DL_MESO

General purpose mesoscopic simulation software package

- MPI domain-decomposed codes with optional OpenMP multithreading for:
  - Dissipative Particle Dynamics (DPD)
  - Lattice Boltzmann Equation (LBE)
- Created in 2004 as CCP5 flagship project: **www.ccp5.ac.uk/DL_MESO**
  - Development currently funded under CoSeC for UKCOMES
- Current version (2.7): released December 2018, 800+ registered academic users



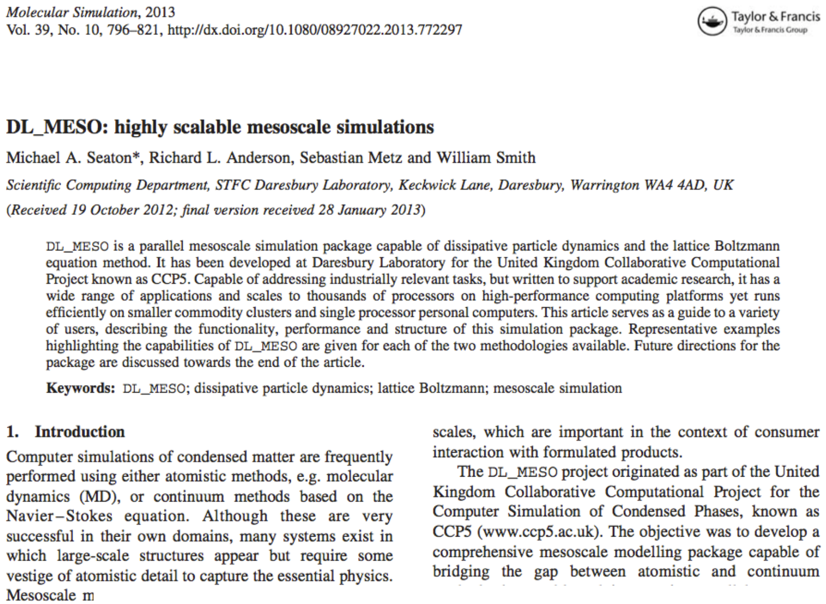


DL_MESO

General purpose mesoscopic simulation software package

- Two articles available in Molecular Simulation on DL_MESO
- Cited ~200 times over past 12 years
- Available under dual licence:
    - Free for academic users
    - Annual subscription for commercial users

Science and Technology Facilities Council



Molecular Simulation, 2013
Vol. 39, No. 10, 796–821, http://dx.doi.org/10.1080/08927022.2013.772297

Taylor & Francis
Taylor & Francis Group

DL_MESO: highly scalable mesoscale simulations

Michael A. Seaton*, Richard L. Anderson, Sebastian Metz and William Smith

Scientific Computing Department, STFC Daresbury Laboratory, Keckwick Lane, Daresbury, Warrington WA4 4AD, UK

(*Received 19 October 2012; final version received 28 January 2013*)

DL_MESO is a parallel mesoscale simulation package capable of dissipative particle dynamics and the lattice Boltzmann equation method. It has been developed at Daresbury Laboratory for the United Kingdom Collaborative Computational Project known as CCP5. Capable of addressing industrially relevant tasks, but written to support academic research, it has a wide range of applications and scales to thousands of processors on high-performance computing platforms yet runs efficiently on smaller commodity clusters and single processor personal computers. This article serves as a guide to a variety of users, describing the functionality, performance and structure of this simulation package. Representative examples highlighting the capabilities of DL_MESO are given for each of the two methodologies available. Future directions for the package are discussed towards the end of the article.

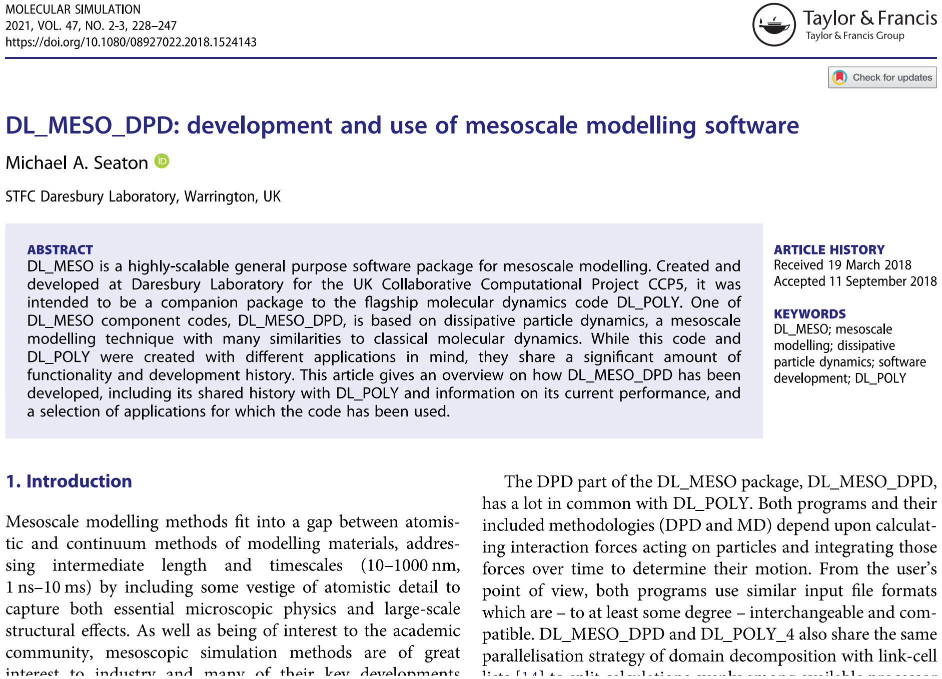
Keywords: DL_MESO; dissipative particle dynamics; lattice Boltzmann; mesoscale simulation

1. Introduction

Computer simulations of condensed matter are frequently performed using either atomistic methods, e.g. molecular dynamics (MD), or continuum methods based on the Navier–Stokes equation. Although these are very successful in their own domains, many systems exist in which large-scale structures appear but require some vestige of atomistic detail to capture the essential physics. Mesoscale m

scales, which are important in the context of consumer interaction with formulated products.

The DL_MESO project originated as part of the United Kingdom Collaborative Computational Project for the Computer Simulation of Condensed Phases, known as CCP5 (www.ccp5.ac.uk). The objective was to develop a comprehensive mesoscale modelling package capable of bridging the gap between atomistic and continuum



MOLECULAR SIMULATION
2021, VOL. 47, NO. 2-3, 228–247
https://doi.org/10.1080/08927022.2018.1524143

Taylor & Francis
Taylor & Francis Group

Check for updates

DL_MESO_DPD: development and use of mesoscale modelling software

Michael A. Seaton

STFC Daresbury Laboratory, Warrington, UK

ABSTRACT
DL_MESO is a highly-scalable general purpose software package for mesoscale modelling. Created and developed at Daresbury Laboratory for the UK Collaborative Computational Project CCP5, it was intended to be a companion package to the flagship molecular dynamics code DL_POLY. One of DL_MESO component codes, DL_MESO_DPD, is based on dissipative particle dynamics, a mesoscale modelling technique with many similarities to classical molecular dynamics. While this code and DL_POLY were created with different applications in mind, they share a significant amount of functionality and development history. This article gives an overview on how DL_MESO_DPD has been developed, including its shared history with DL_POLY and information on its current performance, and a selection of applications for which the code has been used.

ARTICLE HISTORY
Received 19 March 2018
Accepted 11 September 2018

KEYWORDS
DL_MESO; mesoscale modelling; dissipative particle dynamics; software development; DL_POLY

1. Introduction

Mesoscale modelling methods fit into a gap between atomistic and continuum methods of modelling materials, addressing intermediate length and timescales (10–1000 nm, 1 ns–10 ms) by including some vestige of atomistic detail to capture both essential microscopic physics and large-scale structural effects. As well as being of interest to the academic community, mesoscopic simulation methods are of great interest to industry and many of their key developments

The DPD part of the DL_MESO package, DL_MESO_DPD, has a lot in common with DL_POLY. Both programs and their included methodologies (DPD and MD) depend upon calculating interaction forces acting on particles and integrating those forces over time to determine their motion. From the user's point of view, both programs use similar input file formats which are – to at least some degree – interchangeable and compatible. DL_MESO_DPD and DL_POLY_4 also share the same parallelisation strategy of domain decomposition with link-cell

Why do DPD in DL_POLY_5?

- DL_POLY_5 can now do most things DL_MESO_DPD can do, plus a bit more
    - Some DL_POLY_5 functionalities (e.g. rigid body dynamics) potentially useful for DPD calculations, but not widely explored
- Easier to transfer problems from atomistic MD to DPD
    - Can generate atomistic system using DL_FIELD
    - Could systematically coarse-grain atomistic system to DPD scales via e.g. Shapespyer and run DPD calculation immediately
- No need to learn how to use a different code!
    - Inputs for DPD systems similar in form to MD ones
    - Converting between codes not too difficult: similarities in input file formats

DPD in DL_POLY_5

- Intermolecular (van der Waals) interactions include:
  - ‘Standard DPD’ (Groot-Warren)[1]
  - **nDPD (DPD with additional attraction)[2]**
  - **Generalised many-body DPD[3]***
  - **Tabulated interactions supplied in TABVDW file**

* Functionalities added during ARCHER2 eCSE project

Functionality not yet available in DL_MESO_DPD

- DPD thermostat available for NVT ensembles:
  - Simple force integration (MD-VV)*
  - Shardlow splitting[4]: **zeroth***, first and second order
  - Can specify γ for species pairs in FIELD (as additional VDW parameter)

Higher order = more accurate DPD thermostat force integration

- Electrostatic interactions of smeared charges (with SPME)*:
  - Linear[5], Slater-type (exact and truncated[6]), Gaussian[7]

1. Groot and Warren, *J Chem Phys* **107** (11), 4423–4435 (1997)
2. Sokhan *et al.*, *Soft Matter* **19** (30), 5824–5834 (2021)
3. Vanya and Elliot, *PRE* **102** (1), 013312 (2020)
4. Shardlow, *SIAM J Sci Comput* **24** (4), 1267–1282 (2001)
5. Groot, *J Chem Phys* **118** (24), 11265–11277 (2003)
6. González-Melchor *et al.*, *J Chem Phys* **125** (22), 224107 (2009)
7. Warren *et al.*, *J Chem Phys* **138**, 204907 (2013)

DPD units

- New DPD unit scheme added to DL_POLY_5
  - Equivalent to internal DPD units used in DL_MESO_DPD
  - Each unit has fictional equivalent in DL_POLY internal units that does not have to match a 'real' value

DPD unit	Equal to	Unit name	DL_POLY internal equivalent
Mass $[M]$	$[M]$	dpd_m	$m_0 = 1$ Da (u)
Length $[L]$	$[L]$	dpd_l	$\ell_0 = 1$ Å
Energy $[E]$	$[E]$	dpd_e	$E_0 = 10$ J mol^{-1}
Time $[t]$	$[L]\sqrt{[M]/[E]}$	dpd_t	$t_0 = 1$ ps
Temperature $[T]$	$[E]/k_B$	dpd_temp	$1/k_B$ K (ca. 1.2027 K)
Pressure $[P]$	$[E][L]^{-3}$	dpd_p	$\mathcal{P}_0 = 16.61$ MPa (163.9 atm)
Force $[F]$	$[E][L]^{-1} = [M][L][t]^{-2}$	dpd_f	1 Da Å ps^{-2}
Velocity $[V]$	$\sqrt{[M]/[E]} = [L][t]^{-1}$	dpd_v	1 Å ps^{-1}
Charge $[Q]$	$[Q]$	[dpd_q]	$q_0 = \lvert e \rvert \approx 1.6 \times 10^{-19}$ C

DPD units

- Can specify quantities in CONTROL and FIELD in DPD units, including interaction and thermostat parameters
- Need directive in CONTROL file (**io_units_scheme dpd**) to correctly report temperatures, pressures and stress tensors in OUTPUT, STATIS etc.: i.e. printing in DPD units
- Positions, velocities and forces in CONFIG files automatically in DPD units: directly interchangeable with DL_MESO_DPD

```
title DL_POLY DPD lipid bilayer example

io_units_scheme dpd

temperature 1.0 dpd_temp
cutoff 2.0 dpd_l

density_variance 100.0 %

timestep 0.03 dpd_t
time_run 100000 steps
time_equilibration 0 steps
```

```
DL_POLY DPD lipid bilayer example

UNITS dpd

MOLECULES 2
W
nummols 7388
atoms 1  m
W        1.0 0.0 1 0
finish
...
                         A    r_c   γ
VDW 6
H    H    dpd    25.0 1.0 4.5
C    C    dpd    25.0 1.0 4.5
...
```

Science and
Technology
Facilities Council

DPD simulations of molecules

- Can add bonds, angles, dihedrals etc. between beads
  - Not excluding VDW/charge interactions between connected beads
    - Default behaviour in DL_MESO_DPD
    - Bond/angle types starting with ‘-’ in DL_POLY_5
  - DL_POLY_5 can also use fixed-length constraints and rigid bodies (not available in DL_MESO_DPD)

```
DL_POLY DPD lipid bilayer example

...
MOLECULES 2
...
HC6
nummols 700
atoms 7
H       1.0 0.0 1 0
C       1.0 0.0 6 0
bonds 6
-hrm  1 2 128.000 0.500000
-hrm  2 3 128.000 0.500000
-hrm  3 4 128.000 0.500000
-hrm  4 5 128.000 0.500000
-hrm  5 6 128.000 0.500000
-hrm  6 7 128.000 0.500000
angles 5
-cos   1 2 3 20.0 0.0 1.0
-cos   2 3 4 20.0 0.0 1.0
-cos   3 4 5 20.0 0.0 1.0
-cos   4 5 6 20.0 0.0 1.0
-cos   5 6 7 20.0 0.0 1.0
finish
```

Science and
Technology
Facilities Council

File formats: DL_POLY_5 vs DL_MESO_DPD

- CONTROL files
  - DL_MESO_DPD uses similar format to older DL_POLY CONTROL files
  - DL_POLY_5 now uses [keyword] [value] [unit]: must use this format for DPD calculations!
- FIELD files
  - Similar between DL_POLY_5 and DL_MESO_DPD, except DL_MESO_DPD allows definition of non-molecular (solvent) beads and sample configurations of molecules to devise initial configuration without a CONFIG file
- CONFIG files
  - Identical between both codes – can use files generated by either code in both programs
  - Optional for DL_MESO_DPD

CONTROL files: DL_POLY_5 vs DL_MESO_DPD

```
title DL_POLY DPD lipid bilayer example

io_units_scheme dpd

temperature 1.0 dpd_temp
cutoff 2.0 dpd_l

density_variance 100.0 %

timestep 0.03 dpd_t
time_run 100000 steps
time_equilibration 0 steps

traj_calculate ON
traj_start 0 steps
traj_interval 2000 steps
traj_key pos

stats_frequency 2000 steps
stack_size 100 steps

print_frequency 1000 steps
time_job 10800.0 s
time_close 100.0 s

ensemble nvt
ensemble_method dpd
ensemble_dpd_order first
```

```
DL_MESO lipid bilayer example

volume 4096.0

temperature 1.0
cutoff 1.0
boundary halo 2.0

densvar 100.0

timestep 0.03
steps 100000
equilibration steps 0

trajectory 0 2000 0

stats every 2000
stack size 100

print every 1000
job time 10800.0
close time 100.0

ensemble nvt dpds1

finish
```

FIELD files: DL_POLY_5 vs DL_MESO_DPD

```
DL_POLY DPD lipid bilayer example

UNITS dpd

MOLECULES 2
W
nummols 7388
atoms 1
W       1.0 0.0 1 0
finish
HC6
nummols 700
atoms 7
H       1.0 0.0 1 0
C       1.0 0.0 6 0
bonds 6
-hrm  1 2 128.000 0.500000
-hrm  2 3 128.000 0.500000
-hrm  3 4 128.000 0.500000
-hrm  4 5 128.000 0.500000
-hrm  5 6 128.000 0.500000
-hrm  6 7 128.000 0.500000
angles 5
-cos   1 2 3 20.0 0.0 1.0
-cos   2 3 4 20.0 0.0 1.0
-cos   3 4 5 20.0 0.0 1.0
-cos   4 5 6 20.0 0.0 1.0
-cos   5 6 7 20.0 0.0 1.0
finish

VDW 6
H    H    dpd    25.0 1.0 4.5
C    C    dpd    25.0 1.0 4.5
W    W    dpd    25.0 1.0 4.5
H    W    dpd    35.0 1.0 4.5
H    C    dpd    50.0 1.0 9.0
C    W    dpd    75.0 1.0 20.0
close
```

```
DL_MESO lipid bilayer example

SPECIES 3
H        1.0 0.0 0
C        1.0 0.0 0
W        1.0 0.0 7388

MOLECULES 1
HC6
nummols 700
beads 7
H        -0.0172847 0.383451 0.2849
C        -0.016165 0.381111 -0.215093
C        0.280007 0.0703806 0.0412832
C        0.440339 -0.359403 -0.157664
C        0.00469362 -0.136529 -0.260333
C        -0.455453 -0.119133 -0.0654933
C        -0.236136 -0.219878 0.3724
bonds 6
harm   1 2 128.000 0.500000
harm   2 3 128.000 0.500000
harm   3 4 128.000 0.500000
harm   4 5 128.000 0.500000
harm   5 6 128.000 0.500000
harm   6 7 128.000 0.500000
angles 5
cos    1 2 3 20.0 0.0 1.0
cos    2 3 4 20.0 0.0 1.0
cos    3 4 5 20.0 0.0 1.0
cos    4 5 6 20.0 0.0 1.0
cos    5 6 7 20.0 0.0 1.0
Finish

INTERACTIONS 6
H H dpd 25.0 1.0 4.5
C C dpd 25.0 1.0 4.5
W W dpd 25.0 1.0 4.5
H W dpd 35.0 1.0 4.5
H C dpd 50.0 1.0 9.0
C W dpd 75.0 1.0 20.0

close
```

File formats: DL_POLY_5 vs DL_MESO_DPD

- HISTORY files
  - DL_MESO_DPD generates bespoke binary files: need to be converted for visualisation or analysis (utilities to do this supplied with DL_MESO)
  - DL_POLY format text-based and directly readable into VMD and OVITO
- Statistics files
  - DL_MESO_DPD generates CORREL file with flexible number of columns of data (depending on system being modelled): easy to read in and manipulate
  - DL_POLY generates STATIS file with numbers in multi-line blocks: needs to be interpreted to plot or analyse (via e.g. dlpoly-py)

Future DL_POLY_5 DPD functionalities

- Constant pressure (NPT, etc.) ensembles via barostats coupled to DPD thermostat
  - Langevin barostat[1]
- Lees-Edwards boundary conditions for linear shear flows[2]
  - Can be used to calculate viscosities as a function of shear rate (rheology)
- Alternative pairwise thermostats (e.g. Lowe-Andersen[3])
  - Provide larger viscosities than DPD can ordinarily manage
- Dynamic load balancing of particles among processor cores
  - Better parallel scalability for inhomogeneously distributed particle systems, e.g. when using many-body DPD

1. Jakobsen, *J Chem Phys* **122** (12), 124901 (2005)
2. Lees and Edwards, *J Phys C* **47** (2), 145–151 (1999)
3. Lowe, *EPL* **47** (2), 145–151 (1999)

All above functionalities already available in DL_MESO_DPD: use this code if you need them right now!

DPD in DL_POLY_5

- More information about DPD functionalities in DL_POLY_5 available in its documentation (user manual)

https://ccp5.gitlab.io/dl-poly/

  - Specific section on Dissipative Particle Dynamics, including DPD units, thermostats and many-body DPD interactions

https://ccp5.gitlab.io/dl-poly/UserManual/DPD/dpd.html

DPD Tutorial Exercises

- Three available in DL_Software Digital Guide

- Two can work either with DL_MESO_DPD or with DL_POLY_5 (medium difficulty exercises):
  - **Parameterising DPD interactions using Flory-Huggins solution theory**
    - Simulation workflow to find relationship between free energy of mixing (Flory-Huggins χ-parameter) and A
  - **Lipid bilayers and micelles with DPD**
    - Formation of lipid structures from randomised configurations

DPD Tutorial Exercises

- DL_MESO_DPD exclusive exercise:

  - **Transport properties of DPD fluids**
    - Uses Lees-Edwards shear to calculate viscosities, examines DPD and another pairwise thermostat

- All three exercises use DL_POLY_5 or DL_MESO_DPD with input files supplied, some Python scripts provided for simulation workflow (Flory-Huggins exercise) and data visualisation

https://dl-sdg.github.io/RESOURCES/EXERCISES/Exercises.html

Science and
Technology
Facilities Council

Questions?

Science and Technology Facilities Council

Thank you

Science and Technology Facilities Council

@STFC_matters

Science and Technology Facilities Council

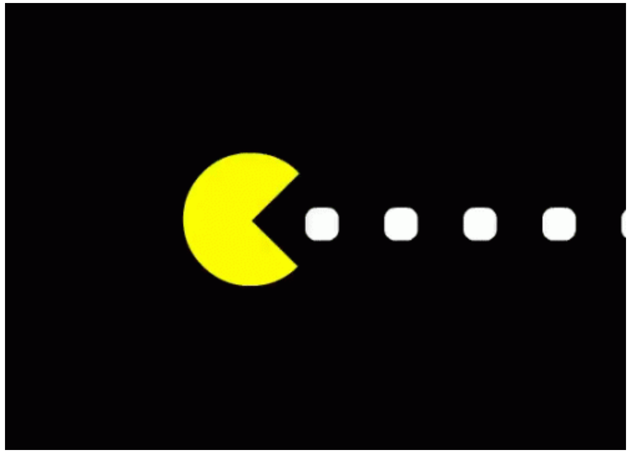
Queen Mary
University of London

DL_SOFTWARE Workshop 2025, Correlations and more

14 May 2025

Jupyter notebook
tinyurl.com/dlp-viscosity

- On-the-fly (also online) algorithms consume a sequence of data, updating immediately on each input
- General on-the-fly correlation module in DL_POLY.

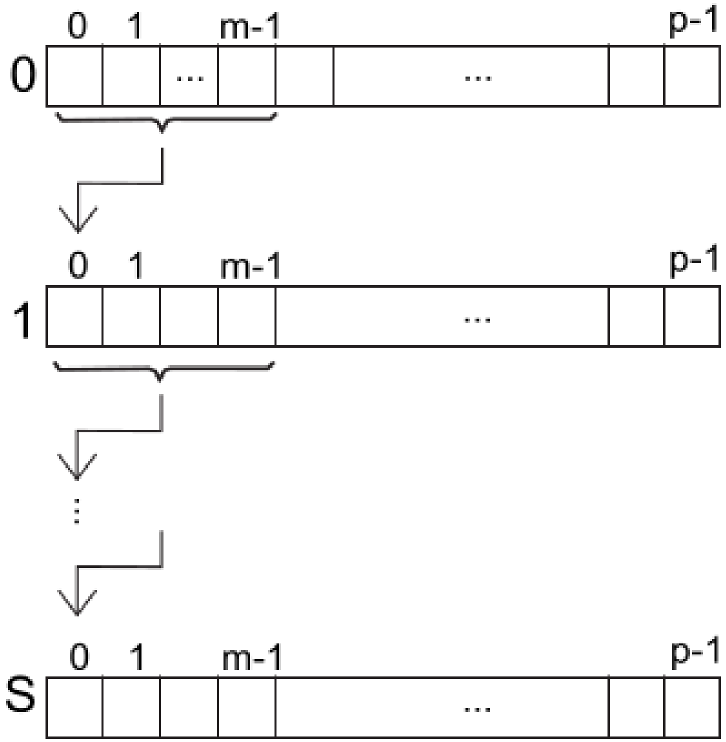


$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Rightarrow \mu_{n+1} = \frac{n\mu_n + x_{n+1}}{n+1}$$

Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.
Karp, R.M., 1992, July. On-line algorithms versus off-line algorithms: How much. In *Algorithms, Software, Architecture: Information Processing 92: Proceedings of the IFIP 12th World Computer Congress* (Vol. 1, p. 416).
Ramírez, J., Sukumaran, S.K., Vorselaars, B. and Likhtman, A.E., 2010. Efficient on the fly calculation of time correlation functions in computer simulations. *The Journal of chemical physics*, *133*(15).

- The multi-tau algorithm calculates on-the-fly correlations [7], following earlier online correlators [8-9].
- Stores data in hierarchical block averages.
- Data is passed down blocks, averaged over m points.



Memory outline of multi tau correlator blocks [7]. Each block stores p data points, and passes down data averaged over m points.

[8] Ramírez, J., Sukumaran, S.K., Vorselaars, B. and Likhtman, A.E., 2010. Efficient on the fly calculation of time correlation functions in computer simulations. *The Journal of chemical physics*, *133*(15).
[9] Frenkel, D. and Smit, B., 2002. *Understanding molecular simulation: from algorithms to applications*. San Diego: Academic Press.
[10] Schätzel, K., Drewel, M. and Stimac, S., 1988. Photon correlation measurements at large lag times: improving statistical accuracy. *Journal of Modern Optics*, *35*(4), pp.711-718.

Correlation	Derived quantities	Atomistic observable
Stress	Viscosity, elastic constants	$\frac{1}{V}\left(\sum_i m_i\lvert\mathbf{r'}_\mathbf{i}^{\alpha}\rvert\lvert\mathbf{r'}_\mathbf{i}^{\beta}\rvert+\sum_{\mathbf{i<j}}\mathbf{r}_\mathbf{ij}^{\alpha}\mathbf{f}_\mathbf{ij}^{\beta}\right)$
Velocity	Transport coefficients, vibrational density of states	$\mathbf{r}'_\mathbf{i}$
Heat flux	Thermal conductivity	$\frac{1}{V}\sum_{i=1}^{N}\left(u_i\mathbf{r}'_i+\sum_{j\neq i}\mathbf{f}_{ij}\cdot\mathbf{r}'_i\mathbf{r}_{ij}\right)$
Current*	Electrical conductivity*	$\sum_i q_i\mathbf{r'}_\mathbf{i}^\mathbf{x}$
Density (user kpoints) $^{\text{dev}}$	Thermal conductivity, intermediate scattering function and dynamic structure factor	$\frac{1}{V}\sum_{i=1}^{N}e^{-i\mathbf{k}\cdot\mathbf{r_i}}$

U. Balucani and M. Zoppi, Dynamics of the liquid state, vol. 10 (Clarendon Press, 1995).
R. Zwanzig and R. D. Mountain, The Journal of Chemical Physics 43, 4464 (1965).
Parrinello, M. and Rahman, A., 1982. Strain fluctuations and elastic constants. *The Journal of Chemical Physics*, *76*(5), pp.2662-2666.
Ray, J.R., 1988. Elastic constants and statistical ensembles in molecular dynamics. *Computer physics reports*, *8*(3), pp.109-151.
Frenkel, D. and Smit, B., 2002. *Understanding molecular simulation: from algorithms to applications*. San Diego: Academic Press.
Hansen, J.P. and McDonald, I.R., 2013. *Theory of simple liquids: with applications to soft matter*. Academic press.
Cheng, B. and Frenkel, D., 2020. Computing the heat conductivity of fluids from density fluctuations. *Physical Review Letters*, *125*(13), p.130602.

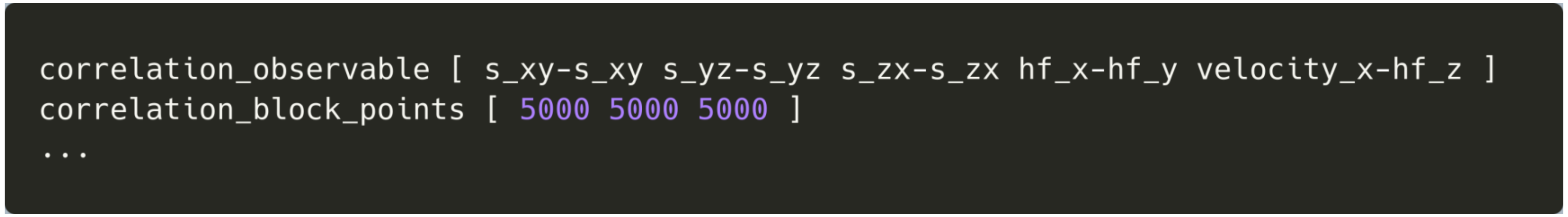
X* not yet supported
X^{dev} In active development

Correlation	Derived quantities	Atomistic observable
Longitudinal momentum current (user kpoints)	Dynamic structure factor [1] liquid spectra [8]	$C_L(\mathbf{k},t) = \langle \mathbf{j}_L^z(\mathbf{k},t)\mathbf{j}_L^z(\mathbf{k},0)\rangle$ $\quad \mathbf{j}_L(\mathbf{k},t) = \sum_i (\mathbf{v}^i(t)\cdot\hat{\mathbf{k}})\hat{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}^i(t)}$
Transverse momentum current (user kpoints)	liquid spectra [8]	$2C_T(\mathbf{k},t) = \langle \mathbf{j}_T^x(\mathbf{k},t)\mathbf{j}_T^x(\mathbf{k},0)\rangle + \langle \mathbf{j}_T^y(\mathbf{k},t)\mathbf{j}_T^y(\mathbf{k},0)\rangle$ $\mathbf{j}_T(\mathbf{k},t) = \sum_i [\mathbf{v}^i(t) - (\mathbf{v}^i(t)\cdot\hat{\mathbf{k}})]\hat{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}^i(t)}$
Energy current (user kpoints)		$\mathbf{q}^a(\mathbf{k}) = \frac{1}{2}\sum_i \left[e_i\mathbf{v}_i^a - \frac{1}{2}\sum_{j\neq i}\sum_b (\mathbf{v}_i^b + \mathbf{v}_j^b)(\mathbf{r}_{ij}^a\mathbf{r}_{ij}^b/\lvert\mathbf{r}_{ij}\rvert^2)\lvert\mathbf{r}_{ij}\rvert\frac{\partial U(\lvert\mathbf{r}_{ij}\rvert)}{\partial r}\frac{1-e^{-i\mathbf{k}\cdot\mathbf{r}_i}}{i\mathbf{k}\cdot\mathbf{r}_i}\right]$
Rigid body: positions, velocities, and orientational velocities		
STATIS values	Bulk modulus, specific heat capacity, … [9]	E.g. temperature, volume, energies, cell angles, …
General molecules?*		Bonds, angles, dihedrals, inversions, positions, velocities, angular velocities?

U. Balucani and M. Zoppi, Dynamics of the liquid state, vol. 10 (Clarendon Press, 1995).
Yang, C., Dove, M.T., Brazhkin, V.V. and Trachenko, K., 2017.
Emergence and evolution of the k gap in spectra of liquid and supercritical states. *Physical review letters*, *118*(21), p.215502.
Allen, M.P. and Tildesley, D.J., 2017. *Computer simulation of liquids*. Oxford university press.

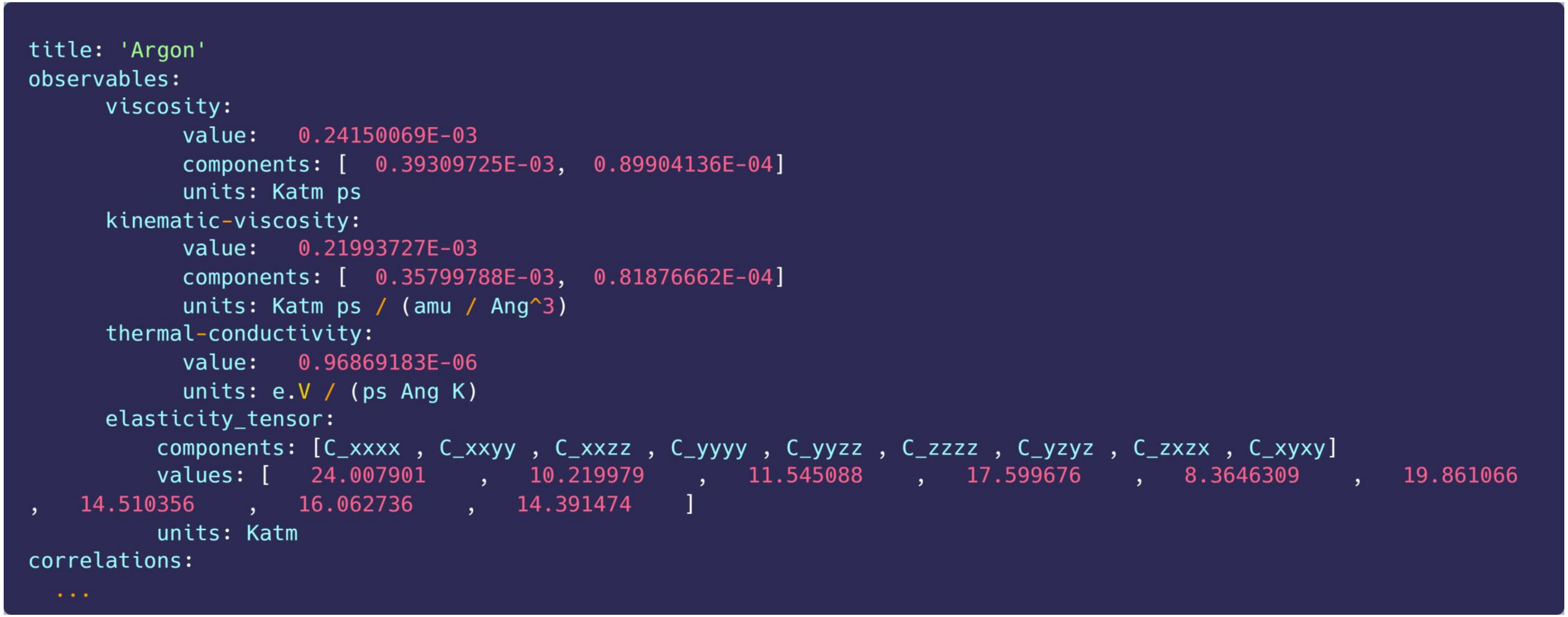
Input

- Request correlations by juxtaposing observables, with specified components
- Averaging of derived quantities is automatic (e.g. stress: xy, yz, zx for viscosity)



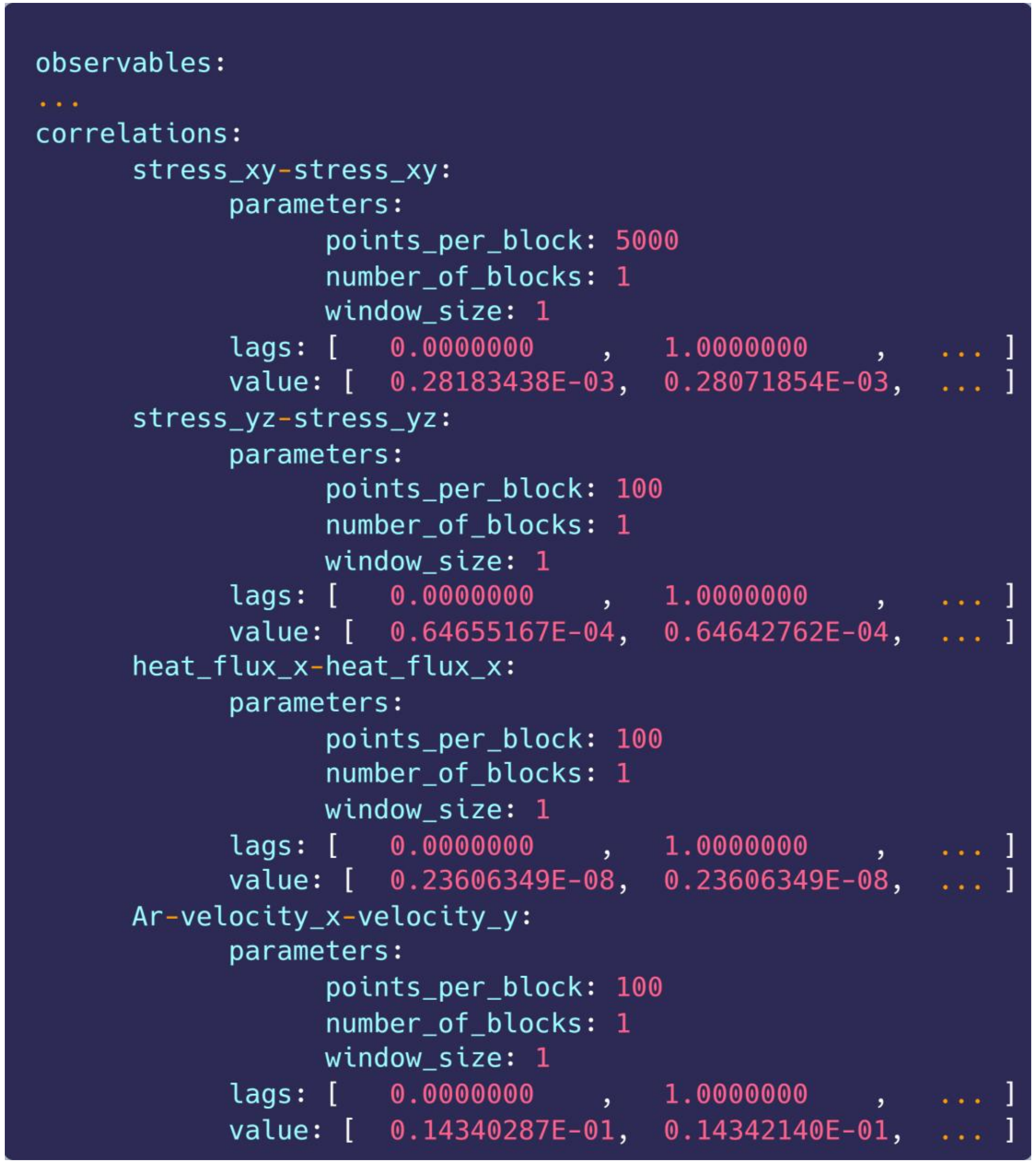
```
correlation_observable [ s_xy-s_xy s_yz-s_yz s_zx-s_zx hf_x-hf_y velocity_x-hf_z ]
correlation_block_points [ 5000 5000 5000 ]
...
```

- New COR file including correlation values, and derived quantities e.g., viscosity, elasticity tensor, or thermal-conductivity.



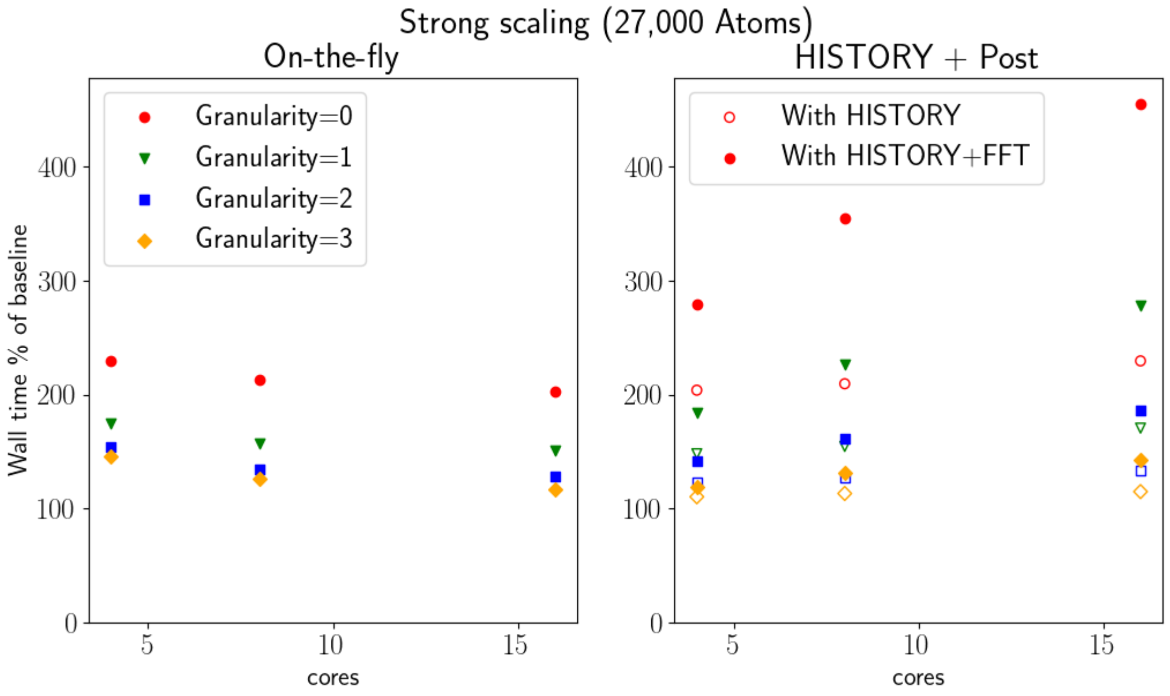
```
title: 'Argon'
observables:
      viscosity:
            value:   0.24150069E-03
            components: [  0.39309725E-03,  0.89904136E-04]
            units: Katm ps
      kinematic-viscosity:
            value:   0.21993727E-03
            components: [  0.35799788E-03,  0.81876662E-04]
            units: Katm ps / (amu / Ang^3)
      thermal-conductivity:
            value:   0.96869183E-06
            units: e.V / (ps Ang K)
      elasticity_tensor:
          components: [C_xxxx , C_xxyy , C_xxzz , C_yyyy , C_yyzz , C_zzzz , C_yzyz , C_zxzx , C_xyxy]
          values: [   24.007901     ,    10.219979     ,    11.545088     ,    17.599676     ,    8.3646309     ,    19.861066
,   14.510356     ,    16.062736     ,    14.391474     ]
          units: Katm
correlations:
  ...
```

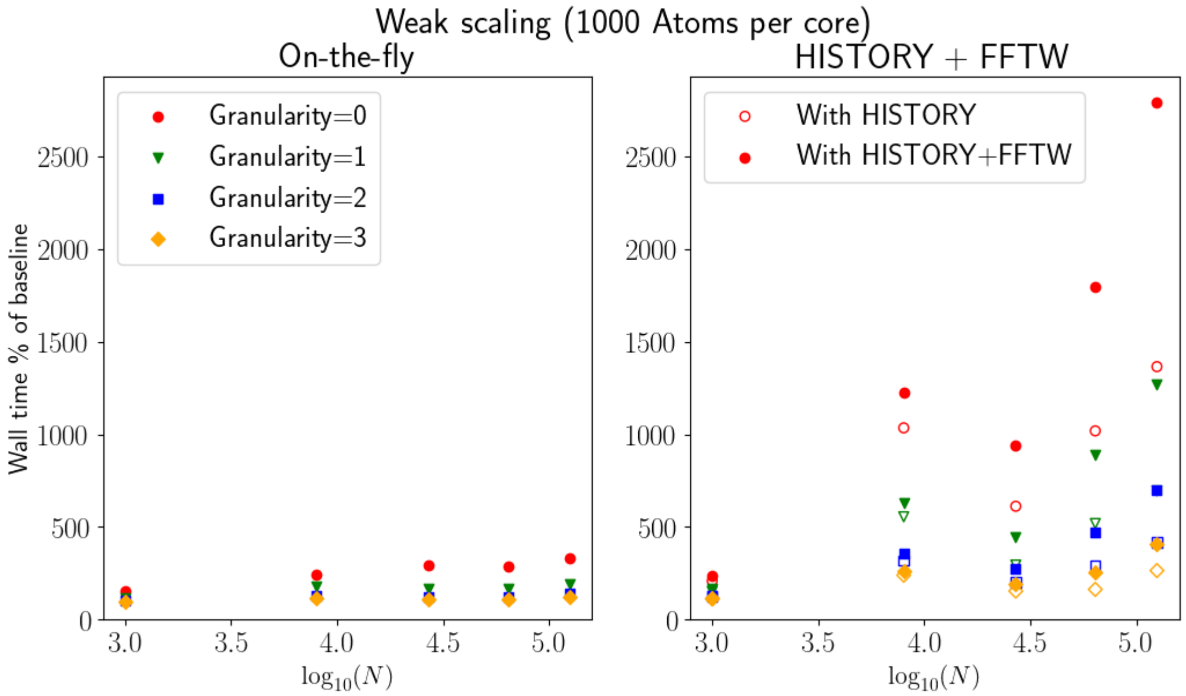
- Values, lags, parameters, and observables all in YAML format
- dlpoly-py reader
- Per-atom or per-molecule quantities are split by species



```
observables:
...
correlations:
      stress_xy-stress_xy:
            parameters:
                  points_per_block: 5000
                  number_of_blocks: 1
                  window_size: 1
            lags: [   0.0000000     ,  1.0000000     ,   ... ]
            value: [  0.28183438E-03,  0.28071854E-03,   ... ]
      stress_yz-stress_yz:
            parameters:
                  points_per_block: 100
                  number_of_blocks: 1
                  window_size: 1
            lags: [   0.0000000     ,  1.0000000     ,   ... ]
            value: [  0.64655167E-04,  0.64642762E-04,   ... ]
      heat_flux_x-heat_flux_x:
            parameters:
                  points_per_block: 100
                  number_of_blocks: 1
                  window_size: 1
            lags: [   0.0000000     ,  1.0000000     ,   ... ]
            value: [  0.23606349E-08,  0.23606349E-08,   ... ]
      Ar-velocity_x-velocity_y:
            parameters:
                  points_per_block: 100
                  number_of_blocks: 1
                  window_size: 1
            lags: [   0.0000000     ,  1.0000000     ,   ... ]
            value: [  0.14340287E-01,  0.14342140E-01,   ... ]
```

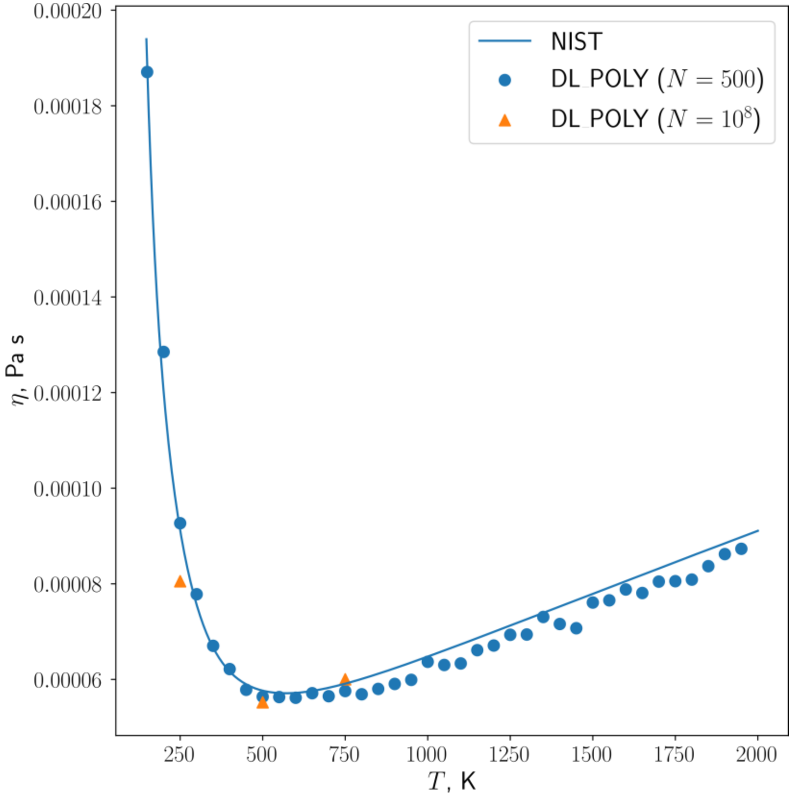
- Scaling (VAF). Granularity is frequency of correlation/ saving trajectory data. FFTW computed in C++

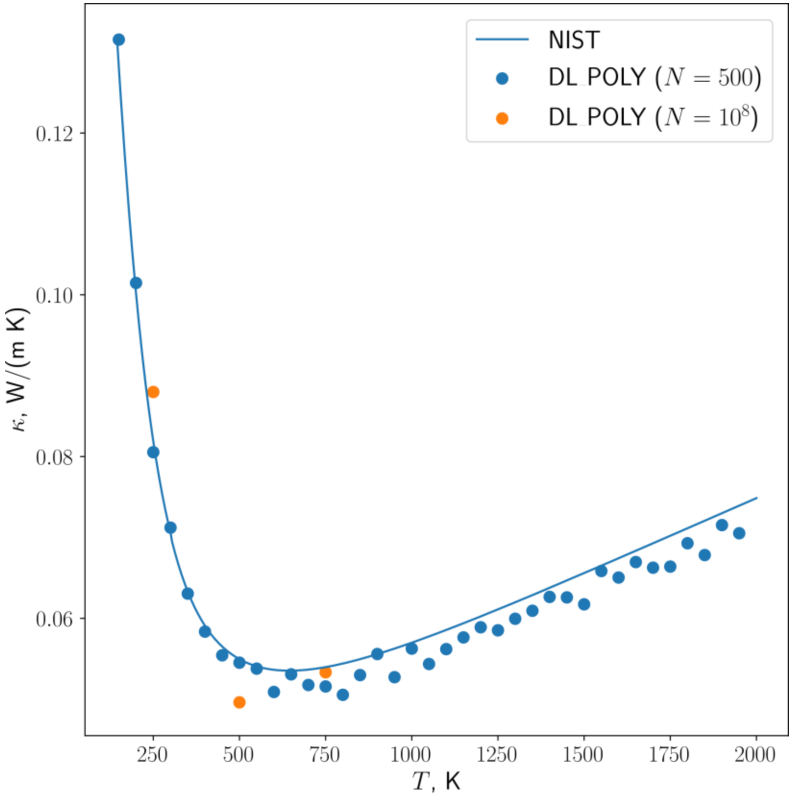




Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.

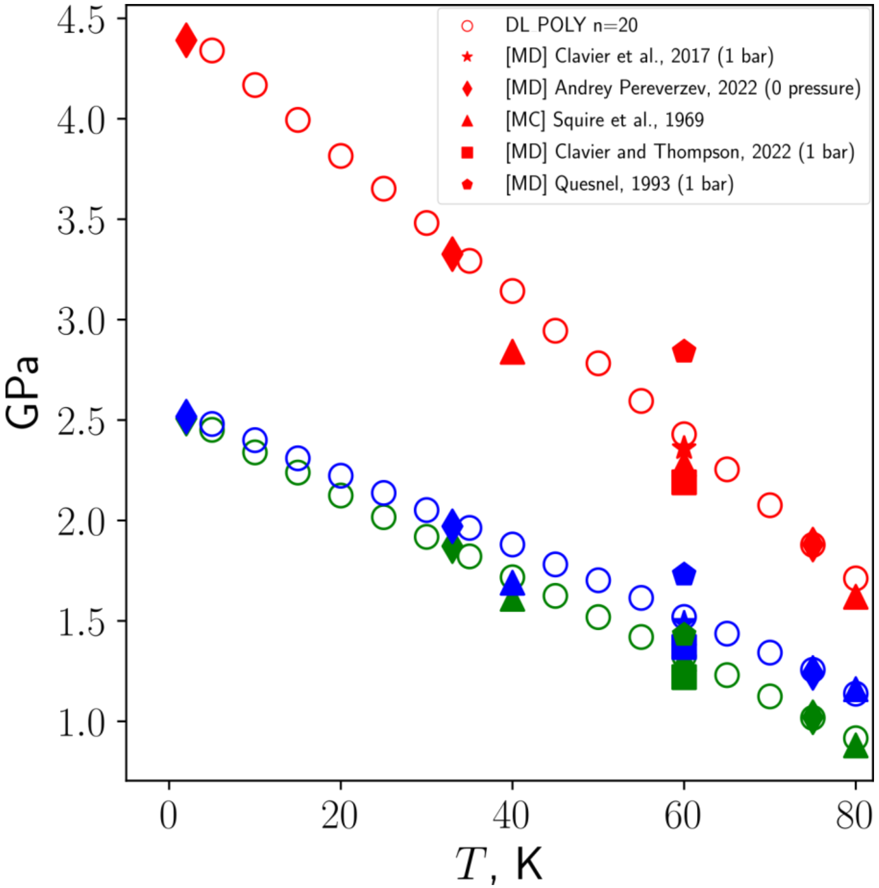
- Viscosity and thermal-conductivity (Argon)

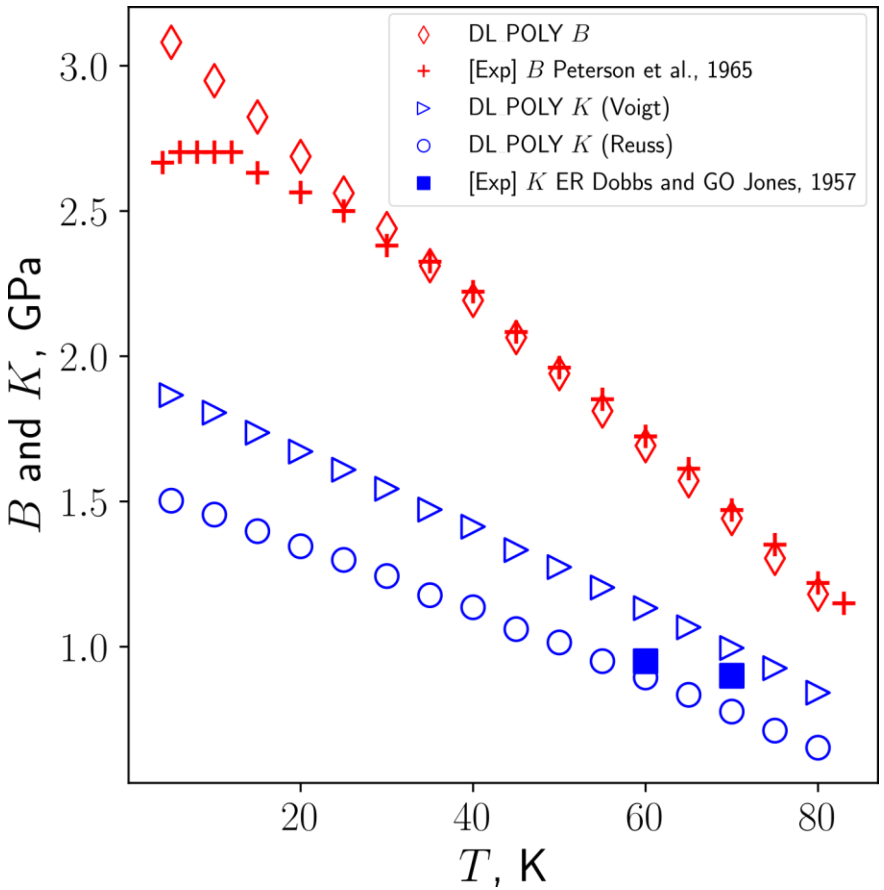




Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.

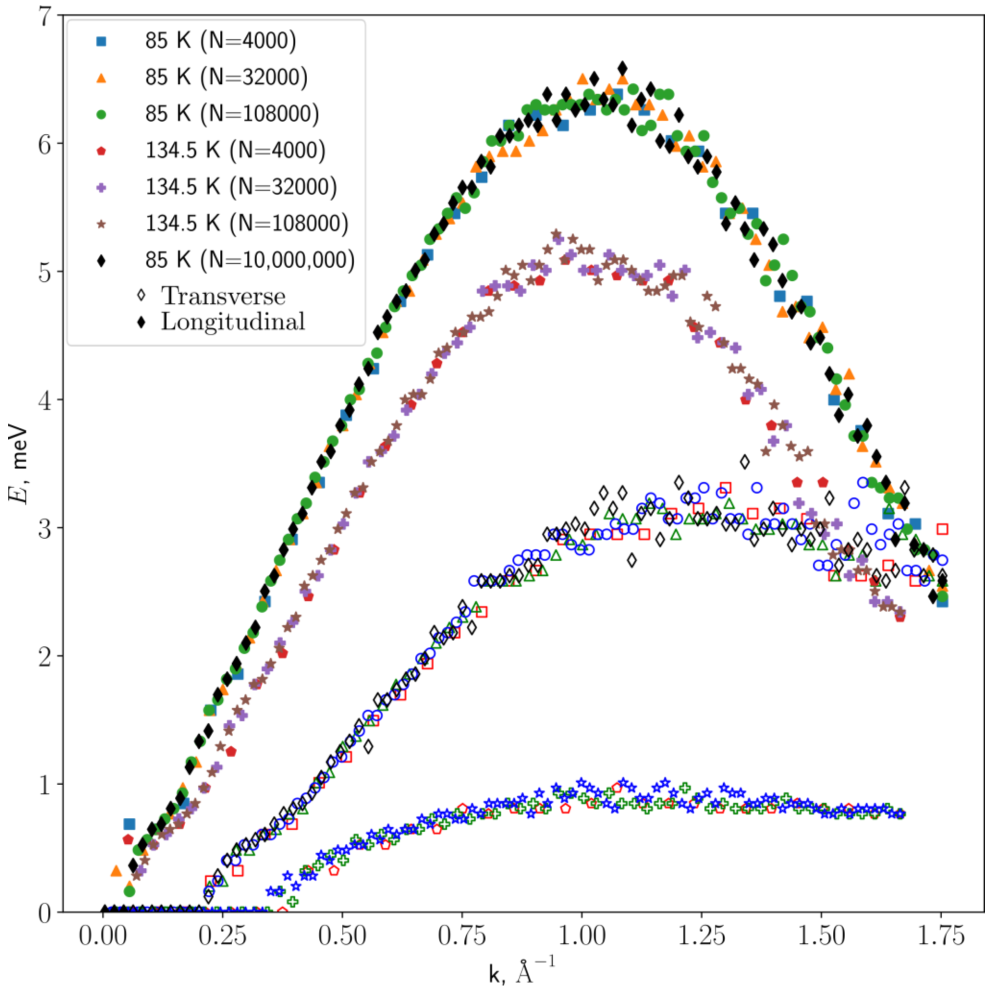
- Elasticity tensor (stress fluctuation method)





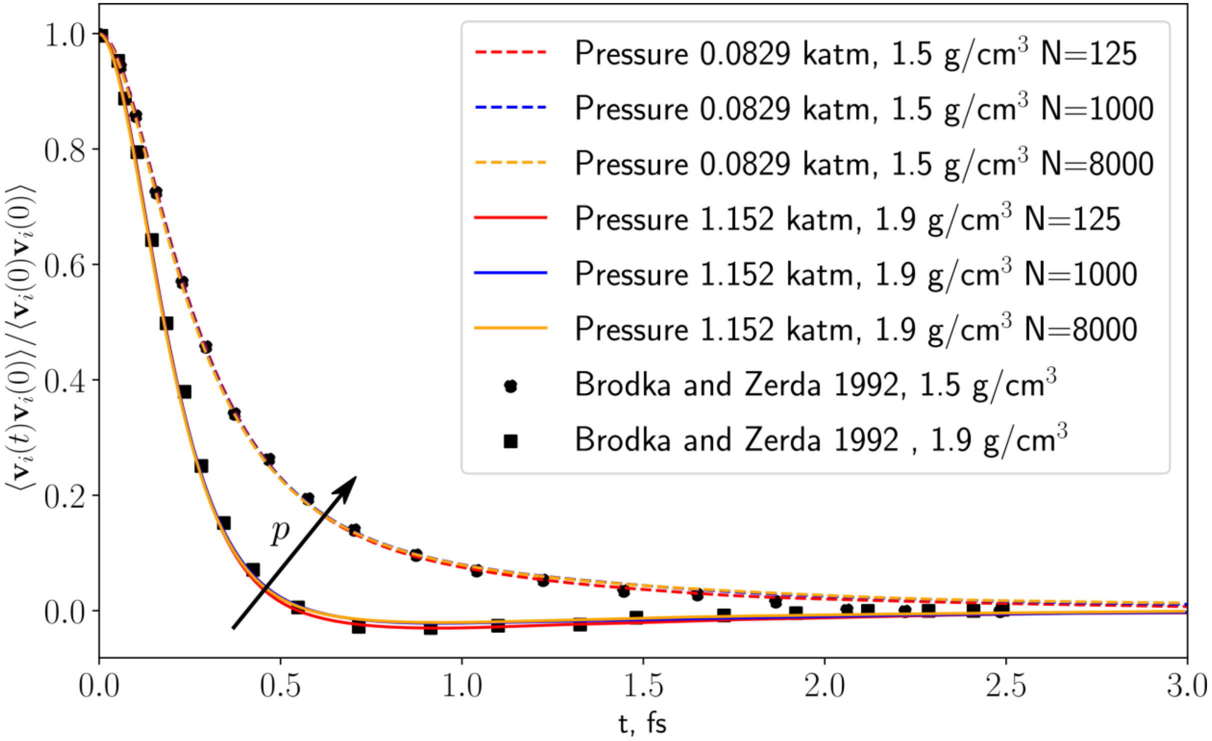
Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.
Clavier, G., Desbiens, N., Bourasseau, E., Lachet, V., Brusselle-Dupend, N. and Rousseau, B., 2017. Computation of elastic constants of solids using molecular simulation: comparison of constant volume and constant pressure ensemble methods. *Molecular Simulation*, *43*(17), pp.1413-1422.

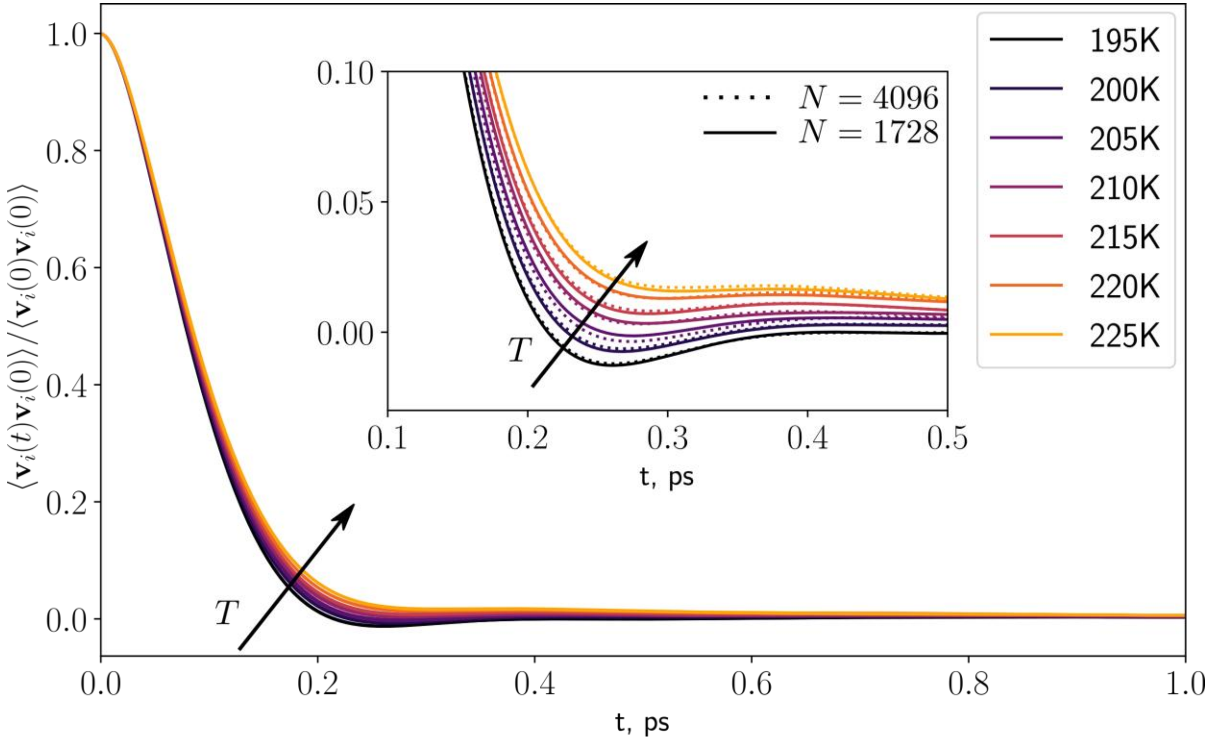
- Currents



Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.

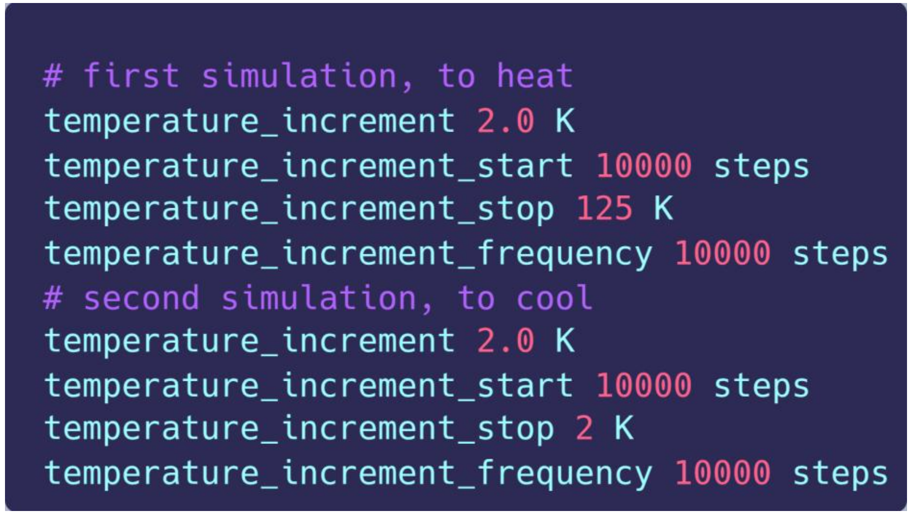
- Rigid bodies (SF6 left and Methane right)



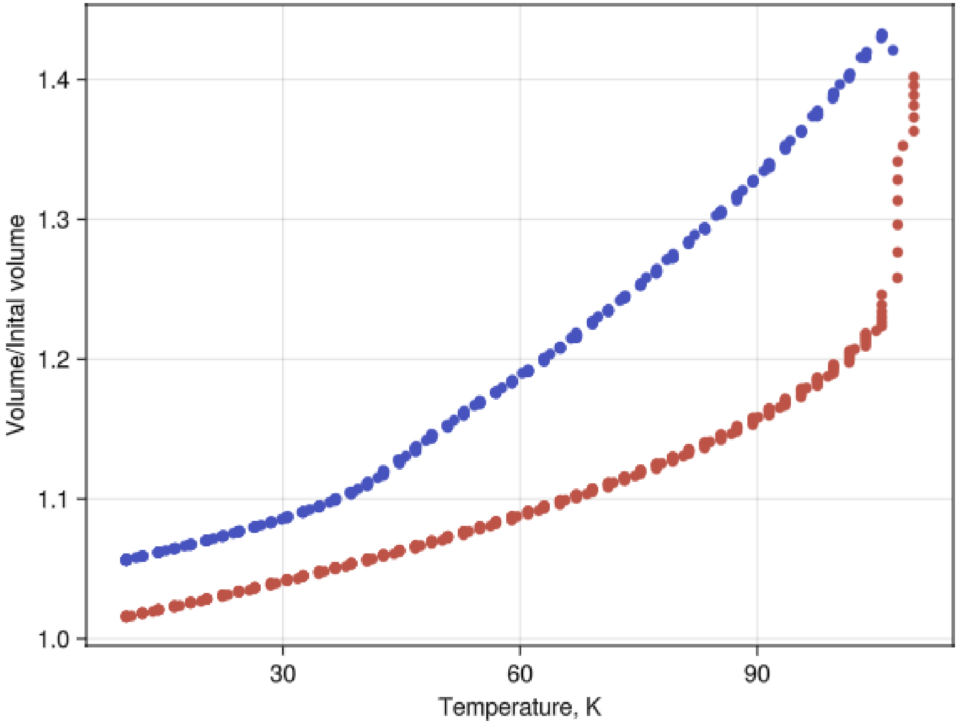


Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.
Yang, C., Brazhkin, V.V., Dove, M.T. and Trachenko, K., 2015. Frenkel line and solubility maximum in supercritical fluids. *Physical Review E*, *91*(1), p.012112.
Brodka, A. and Zerda, T.W., 1992. A molecular dynamics simulation of sulphur hexafluoride. *Molecular Physics*, *76*(1), pp.103-112.

- 2nd derivatives for two-body potentials (e.g., for elastic constants).
- Momentum flux* (for thermal-conductivity corrections) $q = \sum_i m_i \boldsymbol{v}_i$
- Strain tensor (from user supplied reference cell)
- Heating and cooling.

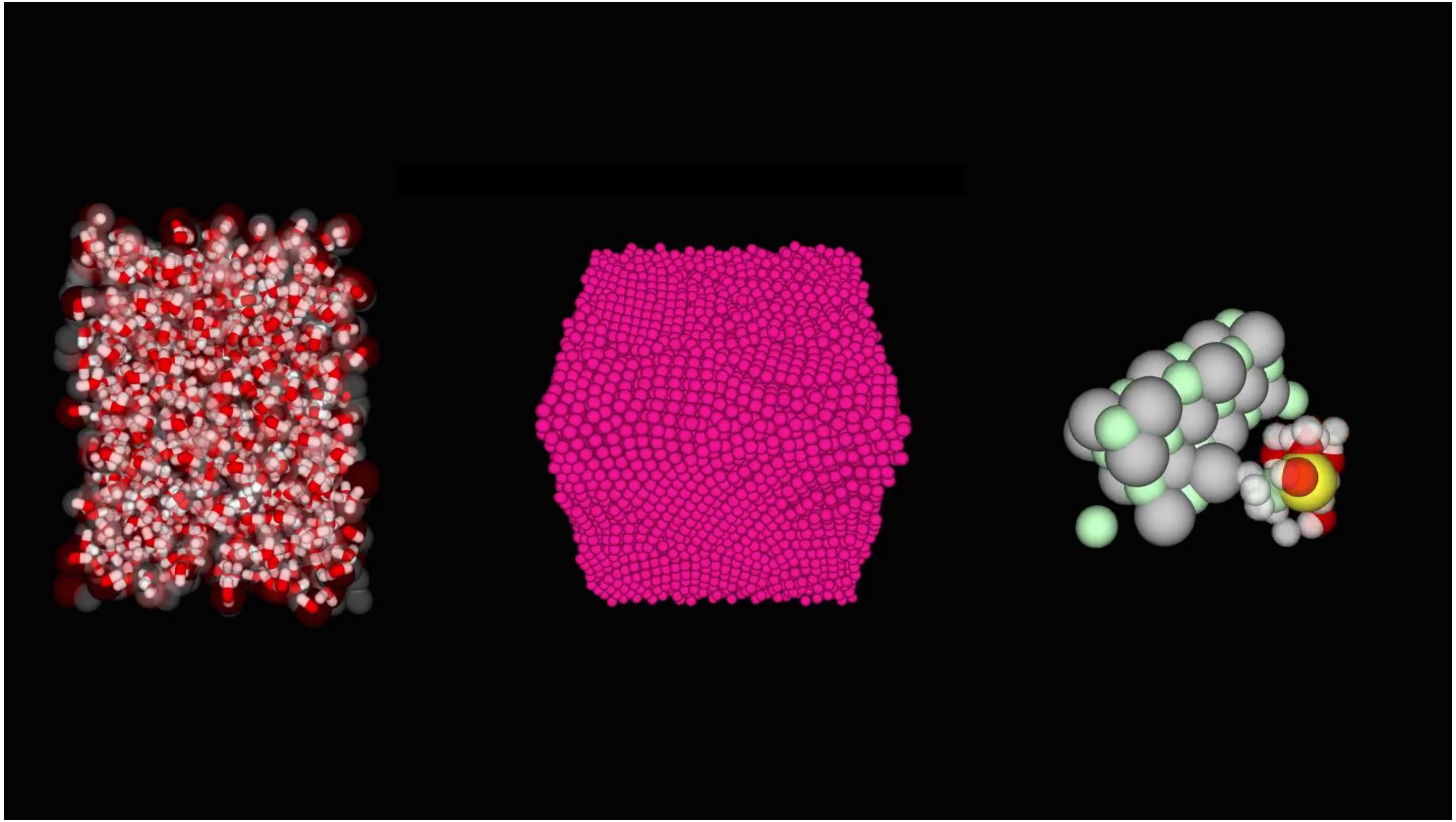


```
# first simulation, to heat
temperature_increment 2.0 K
temperature_increment_start 10000 steps
temperature_increment_stop 125 K
temperature_increment_frequency 10000 steps
# second simulation, to cool
temperature_increment 2.0 K
temperature_increment_start 10000 steps
temperature_increment_stop 2 K
temperature_increment_frequency 10000 steps
```



*Cockrell, C., Withington, M., Devereux, H.L., Elena, A.M., Todorov, I.T., Liu, Z.K., Shang, S.L., McCloy, J.S., Bingham, P.A. and Trachenko, K., 2025. Thermal conductivity and thermal diffusivity of molten salts: insights from molecular dynamics simulations and fundamental bounds. *The Journal of Physical Chemistry B*.

DL_POLY – a brief overview



SPC/Fw model of water. Argon (after melting and freezing at 1 Atm). Hydration of Na (highlighted in yellow) from a NaCl crystal

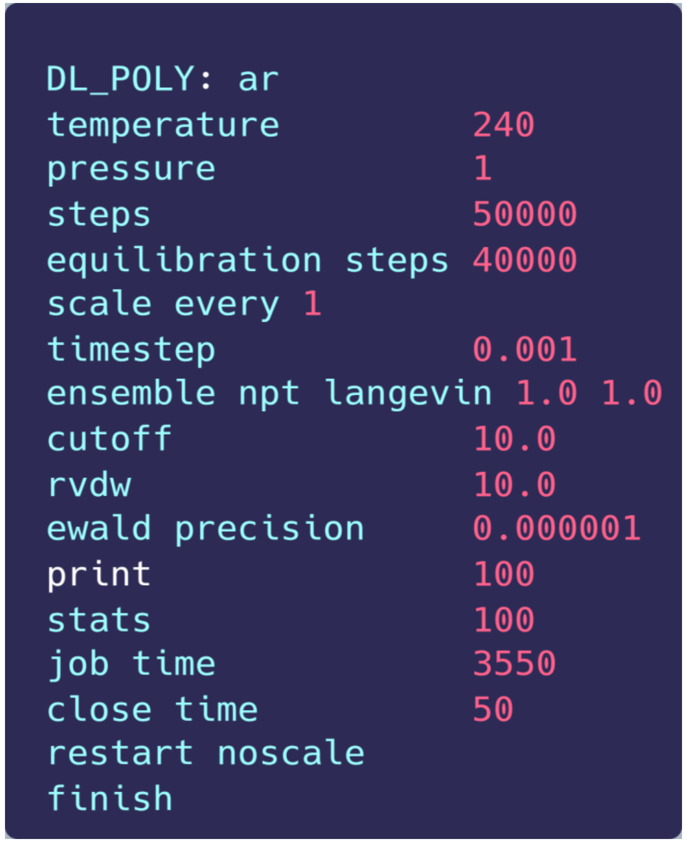
dlpoly-py

DL_POLY

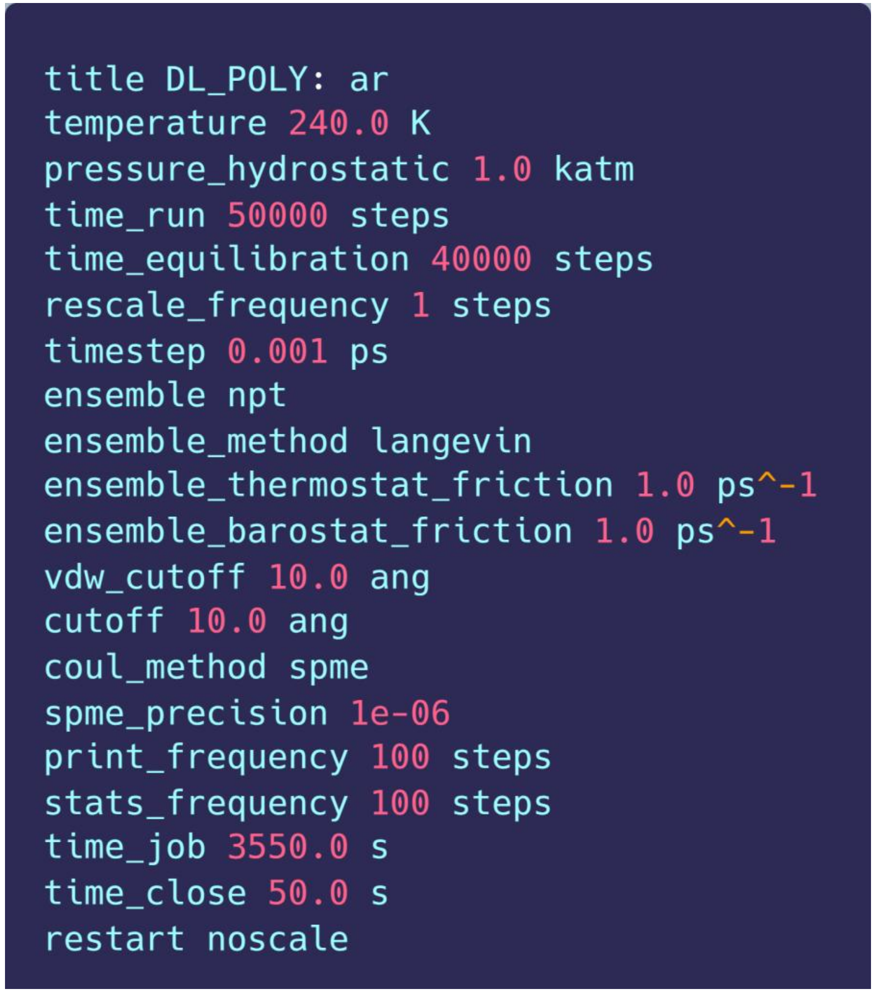
Queen Mary
University of London

From 4 to 5 – New CONTROL

- All control directives now require units where applicable.



```
DL_POLY: ar
temperature           240
pressure              1
steps                 50000
equilibration steps 40000
scale every 1
timestep              0.001
ensemble npt langevin 1.0 1.0
cutoff                10.0
rvdw                  10.0
ewald precision       0.000001
print                 100
stats                 100
job time              3550
close time            50
restart noscale
finish
```



```
title DL_POLY: ar
temperature 240.0 K
pressure_hydrostatic 1.0 katm
time_run 50000 steps
time_equilibration 40000 steps
rescale_frequency 1 steps
timestep 0.001 ps
ensemble npt
ensemble_method langevin
ensemble_thermostat_friction 1.0 ps^-1
ensemble_barostat_friction 1.0 ps^-1
vdw_cutoff 10.0 ang
cutoff 10.0 ang
coul_method spme
spme_precision 1e-06
print_frequency 100 steps
stats_frequency 100 steps
time_job 3550.0 s
time_close 50.0 s
restart noscale
```

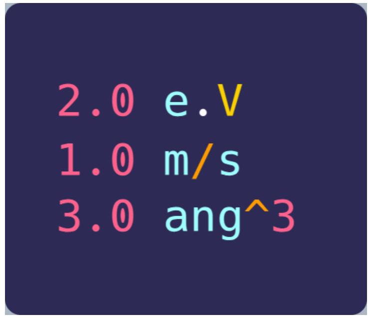
From 4 to 5 – New CONTROL

- All control directives now require units where applicable.

- Various unit schemes are supported with automated conversion.

```
pressure_hydrostatic 0.000986923 katm # 0.99864e-4 GPa
pressure_hydrostatic 1 bar # 1e-4 GPa
pressure_hydrostatic 1 mbar # 1e-7 GPa
pressure_hydrostatic 1 Mbar # 100 GPa
```

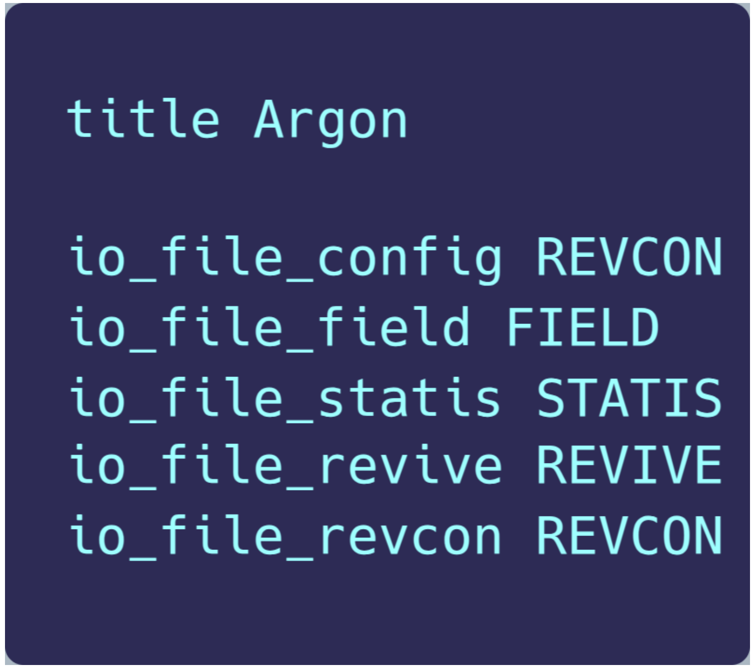
- Including "free-form" units.



```
2.0 e.V
1.0 m/s
3.0 ang^3
```

From 4 to 5 – New CONTROL examples

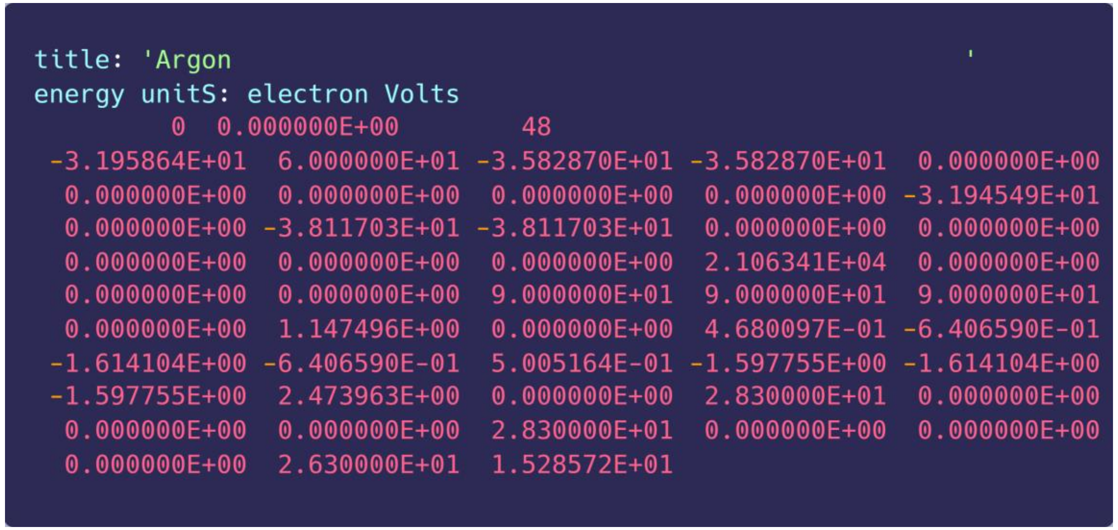
- Control all I/O file names



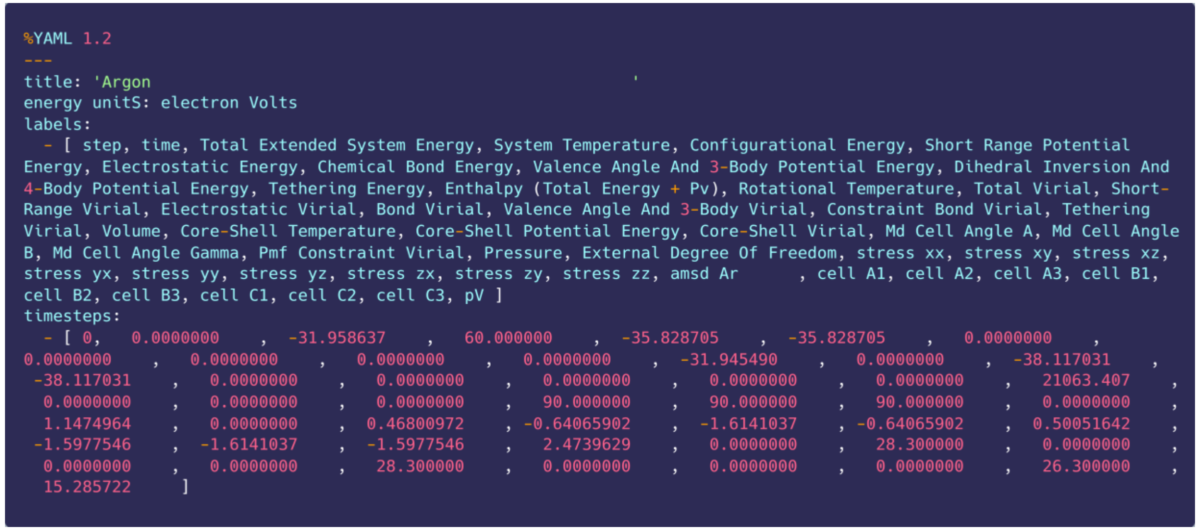
```
title Argon

io_file_config REVCON
io_file_field FIELD
io_file_statis STATIS
io_file_revive REVIVE
io_file_revcon REVCON
```

From 4 to 5 – YAML output

- Can output e.g., STATIS in plain or YAML format.
- Standard format, many languages have a YAML parser package.



```
title: 'Argon                                                  '
energy unitS: electron Volts
         0  0.000000E+00         48
 -3.195864E+01  6.000000E+01 -3.582870E+01 -3.582870E+01  0.000000E+00
  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00 -3.194549E+01
  0.000000E+00 -3.811703E+01 -3.811703E+01  0.000000E+00  0.000000E+00
  0.000000E+00  0.000000E+00  0.000000E+00  2.106341E+04  0.000000E+00
  0.000000E+00  0.000000E+00  9.000000E+01  9.000000E+01  9.000000E+01
  0.000000E+00  1.147496E+00  0.000000E+00  4.680097E-01 -6.406590E-01
 -1.614104E+00 -6.406590E-01  5.005164E-01 -1.597755E+00 -1.614104E+00
 -1.597755E+00  2.473963E+00  0.000000E+00  2.830000E+01  0.000000E+00
  0.000000E+00  0.000000E+00  2.830000E+01  0.000000E+00  0.000000E+00
  0.000000E+00  2.630000E+01  1.528572E+01
```



```
%YAML 1.2
---
title: 'Argon                                                          '
energy unitS: electron Volts
labels:
  - [ step, time, Total Extended System Energy, System Temperature, Configurational Energy, Short Range Potential
Energy, Electrostatic Energy, Chemical Bond Energy, Valence Angle And 3-Body Potential Energy, Dihedral Inversion And
4-Body Potential Energy, Tethering Energy, Enthalpy (Total Energy + Pv), Rotational Temperature, Total Virial, Short-
Range Virial, Electrostatic Virial, Bond Virial, Valence Angle And 3-Body Virial, Constraint Bond Virial, Tethering
Virial, Volume, Core-Shell Temperature, Core-Shell Potential Energy, Core-Shell Virial, Md Cell Angle A, Md Cell Angle
B, Md Cell Angle Gamma, Pmf Constraint Virial, Pressure, External Degree Of Freedom, stress xx, stress xy, stress xz,
stress yx, stress yy, stress yz, stress zx, stress zy, stress zz, amsd Ar      , cell A1, cell A2, cell A3, cell B1,
cell B2, cell B3, cell C1, cell C2, cell C3, pV ]
timesteps:
  - [ 0,   0.0000000    ,  -31.958637    ,   60.000000    ,  -35.828705    ,  -35.828705    ,   0.0000000    ,
0.0000000    ,   0.0000000    ,   0.0000000    ,   0.0000000    ,  -31.945490    ,   0.0000000    ,  -38.117031    ,
 -38.117031    ,   0.0000000    ,   0.0000000    ,   0.0000000    ,   0.0000000    ,   0.0000000    ,   21063.407    ,
  0.0000000    ,   0.0000000    ,   0.0000000    ,   90.000000    ,   90.000000    ,   90.000000    ,   0.0000000    ,
  1.1474964    ,   0.0000000    ,  0.46800972    , -0.64065902    ,  -1.6141037    , -0.64065902    ,  0.50051642    ,
 -1.5977546    ,  -1.6141037    ,  -1.5977546    ,   2.4739629    ,   0.0000000    ,   28.300000    ,   0.0000000    ,
  0.0000000    ,   0.0000000    ,   28.300000    ,   0.0000000    ,   0.0000000    ,   0.0000000    ,   26.300000    ,
  15.285722     ]
```

From 4 to 5 – dlpoly-py

- Python companion package, can organise and manipulate inputs.
- Runs simulations.
- Parses outputs.

```
from dlpoly import DLPoly

dlPoly = DLPoly(control="Ar.control", config="Ar.config",
                field="Ar.field", workdir="argon")
dlPoly.run(numProcs = 4)

# change temperature and rerun, from previous termination
dlPoly = DLPoly(control="Ar.control", config="argon/REVCON", destconfig="Ar.config",
             field="Ar.field", workdir="argon-T310")
dlPoly.control.temp = 310.0
dlPoly.run(numProcs = 4)
```

From 4 to 5 – Currents

- DL_POLY 5 includes a currents module supporting the calculation of user k-point resolved quantities.
- The (k-space) density and longitudinal/transverse momentum currents can be used for calculating thermal conductivity and the intermediate scattering function.

$$n(\mathbf{k},t) = \sum_i e^{i\mathbf{k}\cdot\mathbf{r}^i(t)} \qquad \mathbf{j}_L(\mathbf{k},t) = \sum_i (\mathbf{v}^i(t)\cdot\hat{\mathbf{k}})\hat{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}^i(t)} \qquad \mathbf{j}_T(\mathbf{k},t) = \sum_i [\mathbf{v}^i(t) - (\mathbf{v}^i(t)\cdot\hat{\mathbf{k}})]\hat{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}^i(t)}$$

Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.
Balucani, U. and Zoppi, M., 1995. *Dynamics of the liquid state* (Vol. 10). Clarendon Press.

From 4 to 5 – Currents

- The energy density (using energy per-particle), stress tensor, and energy current all by user supplied k-points.

$$e(\mathbf{k},t) = \frac{1}{2}\sum_i E^i e^{\mathbf{ik}\cdot\mathbf{r}^i(t)}.$$

$$\sigma_{\alpha,\beta}(\mathbf{k}) = \sum_i \left(m v^i_\alpha v^i_\beta - \frac{1}{2}\sum_{j\neq i} \frac{r^{ij}_\alpha r^{ij}_\beta}{|r^{ij}|^2} P(\mathbf{k},\mathbf{r}^{ij}) \right) e^{\mathbf{ik}\cdot\mathbf{r}^i(t)}$$

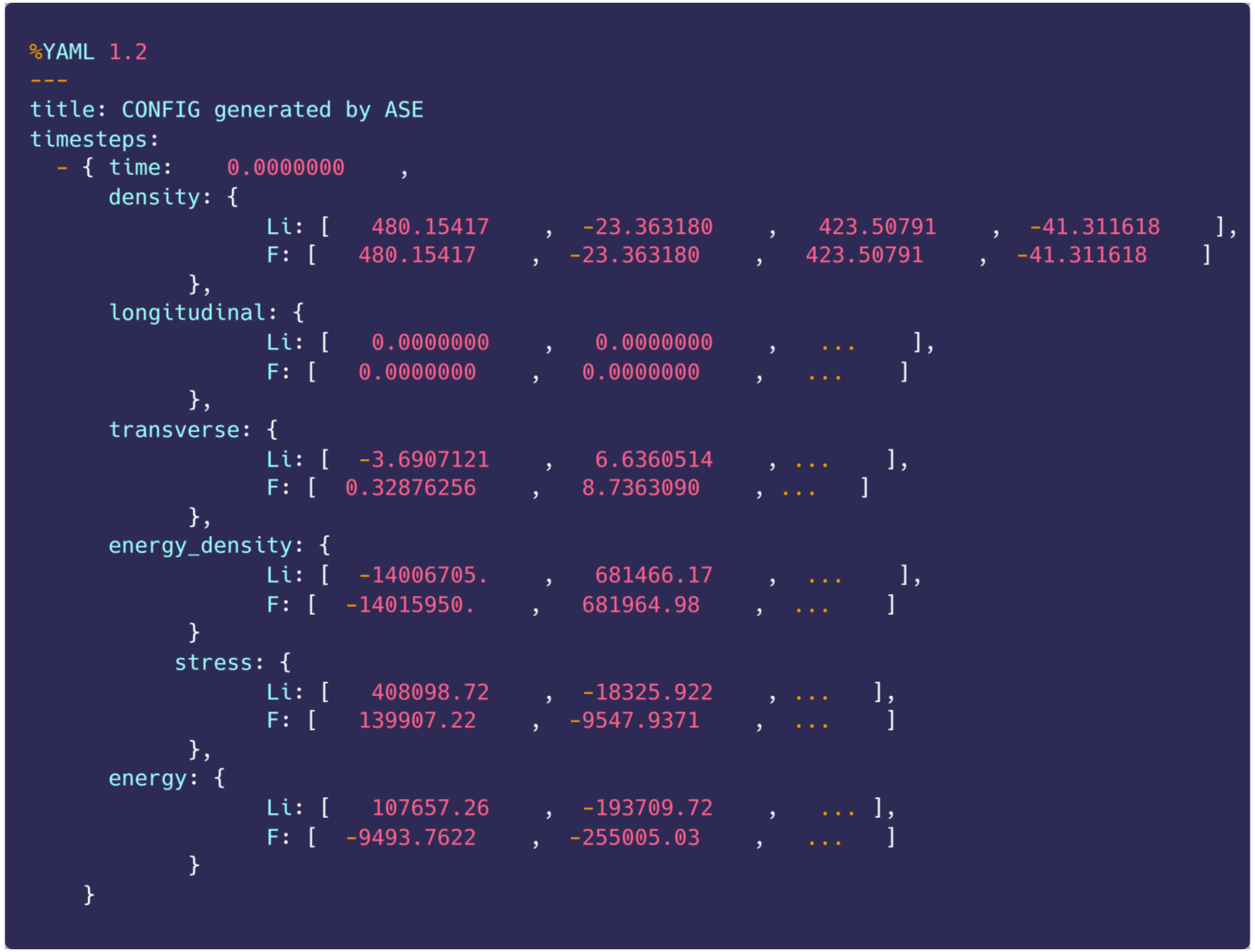
$$\mathbf{q}_a(\mathbf{k}) = \frac{1}{2}\sum_i \left[E^i \mathbf{v}^i_a - \frac{1}{2}\sum_{j\neq i}\sum_b (\mathbf{v}^i_b + \mathbf{v}^j_b)(\mathbf{r}^{ij}_a \mathbf{r}^{ij}_b / |\mathbf{r}^{ij}|^2) P(\mathbf{k},\mathbf{r}^{ij}) \right] e^{\mathbf{ik}\cdot\mathbf{r}^i(t)}$$

$$P_k(\mathbf{k},\mathbf{r}) = |\mathbf{r}| \frac{\partial U(|\mathbf{r}|)}{\partial r} \frac{1 - e^{-i\mathbf{k}\cdot\mathbf{r}}}{i\mathbf{k}\cdot\mathbf{r}}$$

Devereux, H.L., Cockrell, C., Elena, A.M., Bush, I., Chalk, A.B., Madge, J., Scivetti, I., Wilkins, J.S., Todorov, I.T., Smith, W. and Trachenko, K., 2025. DL_POLY 5: Calculation of system properties on the fly for very large systems via massive parallelism. *arXiv preprint arXiv:2503.07526*.
Balucani, U. and Zoppi, M., 1995. *Dynamics of the liquid state* (Vol. 10). Clarendon Press.

From 4 to 5 – *Currents*

- Split by species, with optional YAML output.



```
%YAML 1.2
---
title: CONFIG generated by ASE
timesteps:
  - { time:    0.0000000    ,
      density: {
                  Li: [   480.15417    ,  -23.363180    ,   423.50791    ,  -41.311618    ],
                  F: [   480.15417    ,  -23.363180    ,   423.50791    ,  -41.311618    ]
            },
      longitudinal: {
                  Li: [   0.0000000    ,   0.0000000    ,   ...    ],
                  F: [   0.0000000    ,   0.0000000    ,   ...    ]
            },
      transverse: {
                  Li: [  -3.6907121    ,   6.6360514    , ...    ],
                  F: [  0.32876256    ,   8.7363090    , ...   ]
            },
      energy_density: {
                  Li: [  -14006705.    ,   681466.17    ,  ...    ],
                  F: [  -14015950.    ,   681964.98    ,  ...    ]
            }
           stress: {
                  Li: [   408098.72    ,  -18325.922    , ...   ],
                  F: [   139907.22    ,  -9547.9371    ,  ...    ]
            },
      energy: {
                  Li: [   107657.26    ,  -193709.72    ,   ... ],
                  F: [  -9493.7622    ,  -255005.03    ,   ...   ]
            }
    }
```

Thanks for Listening!

People

Aaron Diver, Aidan Chalk, Alexander Buccheri, Alin Elena, Andres Rojano, Andrey Brukhno, Ben Palmer, Benjamin Speake, Bill Smith, Buket Benek Gursoy, Gavin Khara, Cillian Cockrell, Harvey Devereux, Henry Boateng, Ian Bush, Ilian Todorov, Ivan Scivetti, Jacob Wilkins, Jim Madge, Kiran Jonathan, Kostya Trachenko, Laurence Ellison, Martin Dove, Michael Lysaght, Michael Seaton, Oliver Dicks, Ondřej Čertík, Peicho Petkov, Peter Nash, Ruairi Nestor, Ruslan Davidchack, Ryan Hunt, Samuel Murphy, Szymon Daraszewicz, Thomas Durrant, Vlad Sokhan, Yaser Afshar, You Lu

DL_POLY

dlpoly-py

Silicon (Apocrita@QMUL, Sluis Tier 2, and SCARF@STFC)

- Computing resources provided by STFC Scientific Computing Department's SCARF cluster
- This research utilised Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT. http://doi.org/10.5281/zenodo.438045
- Calculations were performed using the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick. Sulis is funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium.

Daresbury Laboratory

UKRI Engineering and Physical Sciences Research Council

EP/W029006/1

UKRI Science and Technology Facilities Council

CoSeC Computational Science Centre for Research Communities

HEC MCC UK

https://gitlab.com/ccp5/dl-poly

https://gitlab.com/drFaustroll/dlpoly-py