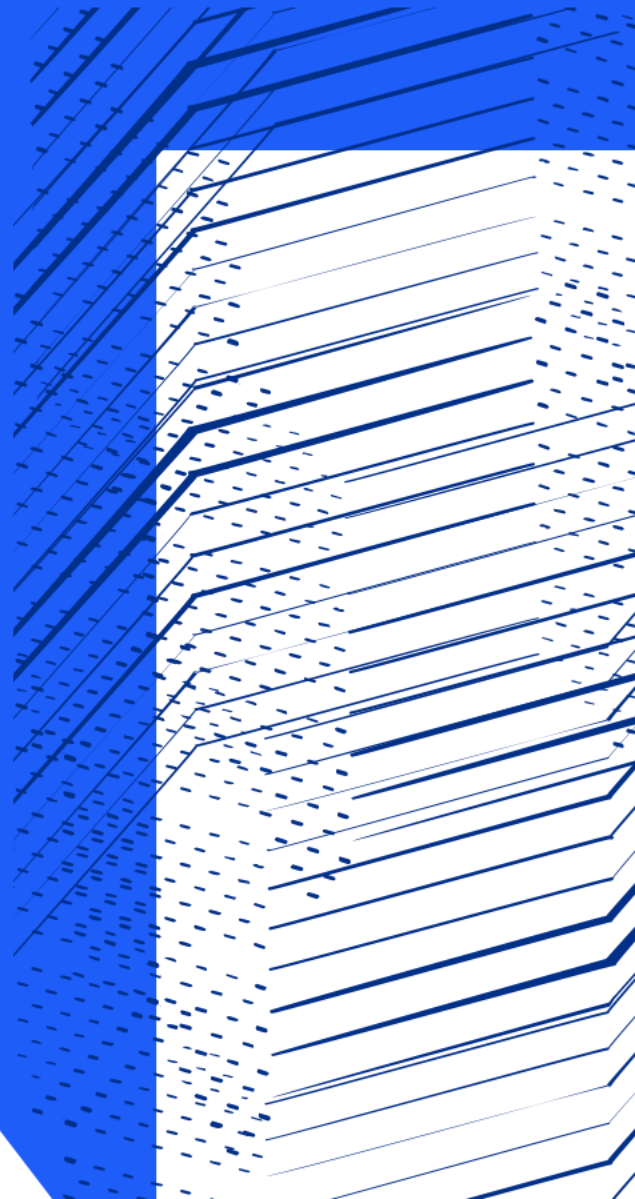
Science and Technology Facilities Council

DL_Software and Lattice Boltzmann Equation (LBE)

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



DL SOFTWARE
Enabling Discovery

CCP5 Collaborative Computational Project
Computer Simulation of Condensed Phases

UK COMES
Consortium on Mesoscale Engineering Sciences

archer2

Science and Technology Facilities Council

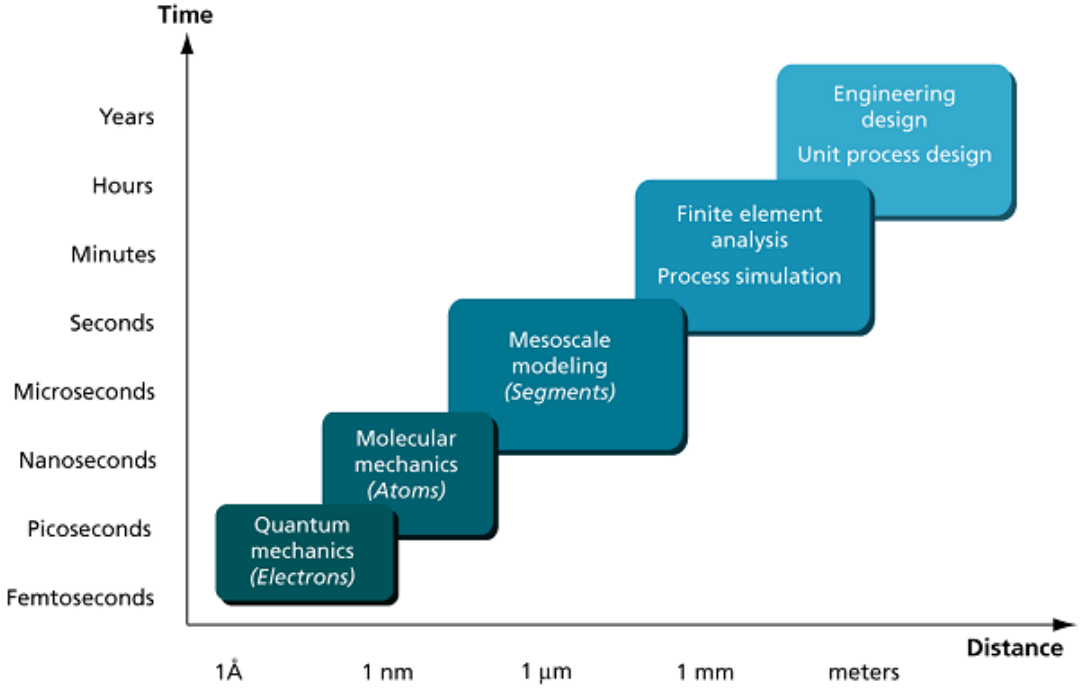
The mesoscale

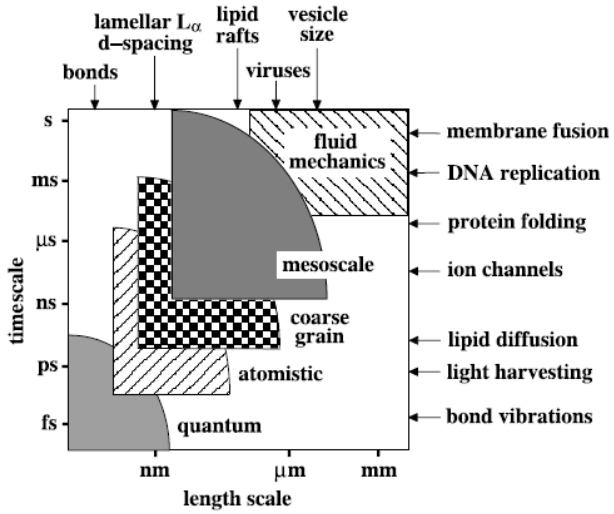
‘If life is going to exist in a Universe of this size, then the one thing it cannot afford to have is a sense of proportion.’
- Douglas Adams, *The Restaurant at the End of the Universe*

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 and hydrodynamics needed
  - Parameterisation may involve bottom-up (microscopic) and top-down (macroscopic)
- Models often involve bigger-than-atom particles (‘beads’)

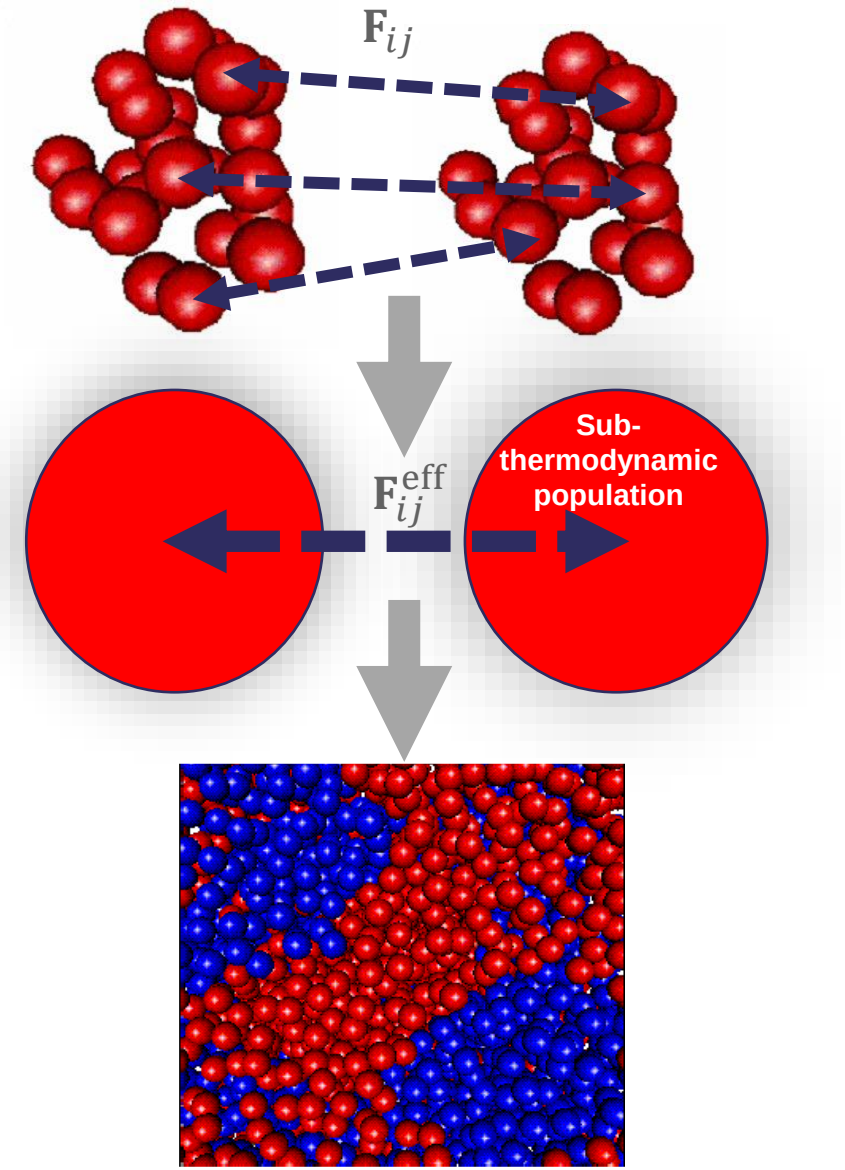
Mesoscale modelling approaches

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



Science and Technology Facilities Council

Science and Technology Facilities Council

Lattice Boltzmann Equation (LBE)

‘He was constantly reminded of how startlingly different a place the world was when viewed from a point only three feet to the left.’
- Douglas Adams, *The Salmon of Doubt*

Lattice Boltzmann Equation (LBE)

Moving from particle dynamics to statistical mechanics

- Still want to consider how particles move around, but at very large scales (cf. atomistic MD) the motion of an individual particle matters very little!
- An approach that can work at mesoscale: using probabilities of finding particles and evolving these over time
  - **Distribution functions**: probabilities of finding particles at given time and space with specific momenta (moments give fluid density, momentum etc.)
  - Distribution functions evolve via **Boltzmann equation**:

Propagation (free-streaming) of particles in system due to velocity field and applied forces

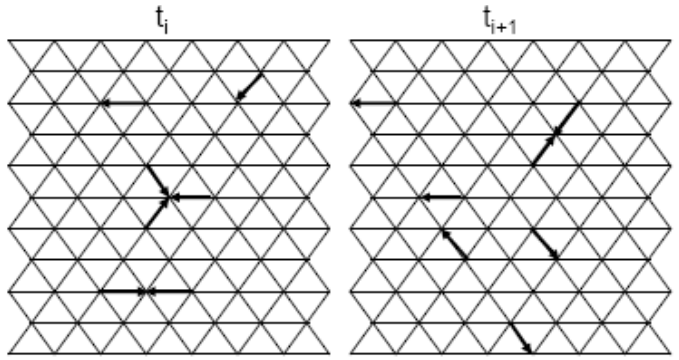
$$\frac{\partial f}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla f + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = \left(\frac{\partial f}{\partial t} \right)_{coll}$$

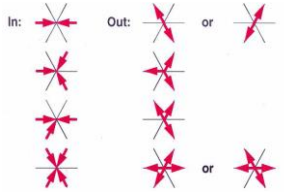
Collisions of particles when encountering each other: governed by e.g. fluid viscosity

- Collisions can be complex, especially if large number of particles involved
- Keeping track of particles and solving Boltzmann equation more difficult if there are too many possible momenta and positions

Lattice gas cellular automata

Antecedent to LBE (and DPD)





- Gas particles stream ('propagate') on a regular lattice along links with set velocities $\mathbf{e}_i$
- Particles collide as they encounter each other at lattice sites: maintain mass/total momentum, related to viscosity

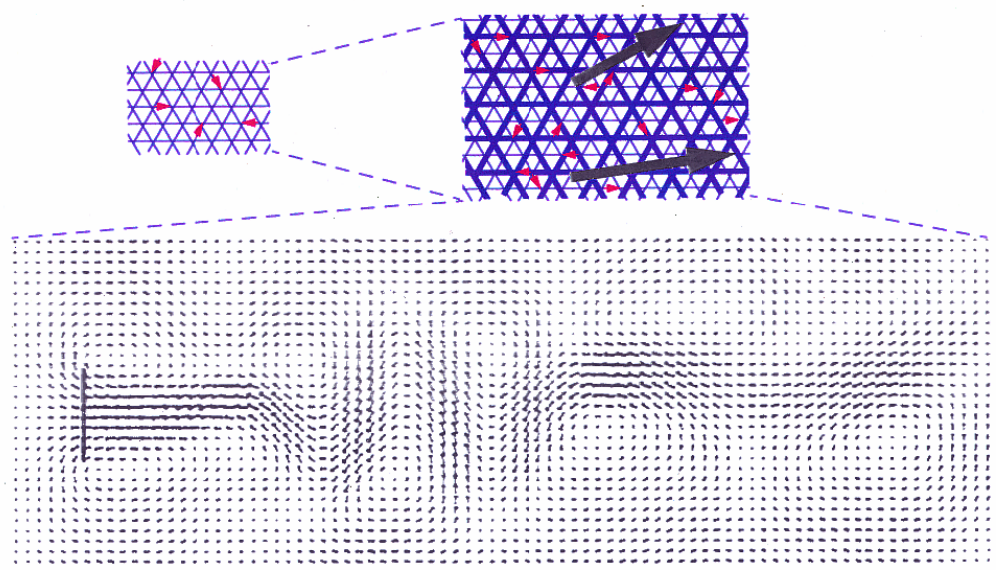
$$n_i(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t) - n_i(\mathbf{x}, t) = \Delta_i(n_0, n_1, \ldots n_N)$$

Pros

- Boolean/integer calculations: unconditional numerical stability
- Easy to parallelise
- Multi-physics possible
- Can show connection to Navier-Stokes fluid flow equations

Cons

- Statistically noisy: limited viscosities available
- Ensemble-averaging required to see results
- No (automatic) Galilean invariance, anisotropic flows
- Large number of possible collision states
- Only small number of lattice schemes will work



LBE: LGCA with statistical mechanics

Getting from LGCA to LBE in three easy(ish!) steps

1. Substitute Boolean particle occupation numbers n_i with density distribution functions f_i[1]
   - Ensemble-averaging LGCA *ab initio*
   - Moments of f_i give macroscopic fluid density, momentum etc.

$$f_i = \langle n_i \rangle$$

$$\rho(\mathbf{x}) = \sum_i f_i(\mathbf{x})$$

$$\rho(\mathbf{x})\mathbf{u}(\mathbf{x}) = \sum_i \mathbf{e}_i f_i(\mathbf{x})$$

Eliminates statistical noise of LGCA and need to ensemble-average results to actually see flows(!), widens available viscosity range

1. McNamara and Zanetti, *PRL* **61** (20), 2332–2335 (1988)

LBE: LGCA with statistical mechanics

Getting to LBE in three easy(ish!) steps

2. Transform collision matrix Δ_i into linearised form based on deviation from local equilibrium f_i^{eq} [1]

   - Matrix Ω_{ij} considerably smaller than before (e.g. from 2^N elements to N^2 for N links per grid point)
   - Can now be considered as a **collision operator**: tries to shift each point towards an equilibrium state

$$\Delta_i(f_0, f_1, \dots f_N) \rightarrow \sum_j \Omega_{ij}\left(f_j - f_j^{eq}\right)$$

$$f_i^{eq} = f(\rho, \mathbf{u})$$

Big improvement in computational speed, Galilean invariance now guaranteed

1. Succi *et al.*, *Physica D* **47** (1–2), 219–230 (1991)

LBE: LGCA with statistical mechanics

Getting to LBE in three easy(ish!) steps

3. Simplify collisions even further by setting Ω_{ij} to unit diagonal with a relaxation parameter τ[1]

   - Old trick used to solve (continuous) Boltzmann equation[2] for low density gases (assuming pairs of particles collide without prior correlation)
   - Most commonly-used form of LBE for simulations

Range of useable lattices expanded[1]: square lattices with multiple speeds will now work!

$$\Omega_{ij} \rightarrow -\frac{\delta_{ij}}{\tau}$$

No matrix operations now necessary, so even more speed-up

$$f_i(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau}\Big(f_i(\mathbf{x}, t) - f_i^{eq}\big(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t)\big)\Big)$$

Science and Technology Facilities Council

1. Qian *et al.*, *EPL* **17** (6), 479–484 (1992)
2. Bhatnagar *et al.*, *Phys Rev* **94** (3), 511–525 (1954)
3. He and Luo, *PRE* **56** (6), 6811–6817 (1997)

Discretising continuous Boltzmann equation also gets us here[3]

LBE solves most issues with LGCA …

- No ensemble-averaging required afterwards
- No statistical noise: wider viscosity range
- Many more lattice schemes possible, including square-grid multispeed ones, with guaranteed isotropy (Galilean invariance)
- Simpler and cheaper collisions
- Multi-physics and parallelisation still possible
- Still a particle-based method (underneath)
- But …
  - Floating point numbers instead of integers: no longer unconditionally stable
  - Lattices still usually isothermal/athermal: often cannot provide sufficient particle speeds to obtain definable temperature* and deal with very compressible flows

$$f_i(\mathbf{x}+\mathbf{e}_i\Delta t, t+\Delta t) - f_i(\mathbf{x},t) = -\frac{1}{\tau}\left(f_i(\mathbf{x},t) - f_i^{eq}\left(\rho(\mathbf{x},t),\mathbf{u}(\mathbf{x},t)\right)\right)$$

Local equilibrium based on Maxwell-Boltzmann distribution:

$$f_i^{eq}(\rho,\mathbf{u}) = \frac{\rho}{(2\pi RT)^{D/2}}\exp\left(-\frac{m(\mathbf{e}_i-\mathbf{u})^2}{2k_BT}\right)$$

but usually approximated for low Mach numbers ($u \ll c_s = \sqrt{3k_BT/m}$) to:

$$f_i^{eq}(\rho,u) = \rho w_i[1+3(\mathbf{e}_i\cdot\boldsymbol{u}) + \tfrac{9}{2}(\mathbf{e}_i\cdot\boldsymbol{u})^2 - \tfrac{3}{2}u^2]$$

where $c_s^2 = \frac{1}{3}\left(\frac{\Delta x}{\Delta t}\right)^2$, $p = \rho c_s^2$ and

$$\nu = \frac{\mu}{\rho} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)\frac{(\Delta x)^2}{\Delta t}$$

Science and Technology Facilities Council

Connecting LBE to Navier-Stokes[1]

Assume small Knudsen number ϵ (molecular mean free path/length scale)

$$f_i = f_i^{eq} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + \dots$$
$$\sum_i f_i^{eq} = \rho$$
$$\sum_i \mathbf{e}_I f_i^{eq} = \rho \mathbf{u}$$
$$\sum_i f_i^{(k>0)} = \sum_i \mathbf{e}_i f_i^{(k>0)} = 0$$

Length and time scales for convection (1) and diffusion (2) (Chapman-Enskog multiscaling)

$$\frac{\partial}{\partial x} = \epsilon \frac{\partial}{\partial x_1} + \cdots$$
$$\frac{\partial}{\partial t} = \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2} + \cdots$$

Taylor expand LBE about ϵ, put in multiscale derivatives, collect terms for ϵ^1 and ϵ^2

Fast modes (ϵ^1):

$$\partial_{t1} f_i^{eq} + \mathbf{e}_i \cdot \nabla_1 f_i^{eq} = \frac{1}{\tau} f_i^{(1)}$$

Slow modes (ϵ^2):

$$\partial_{t2} f_i^{eq} + \left(1 - \frac{1}{2\tau}\right)(\partial_{t1} + \mathbf{e}_i \cdot \nabla_1) f_i^{(1)} = -\frac{1}{\tau} f_i^{(2)}$$

Take moments in $\mathbf{e}_i$, sum over lattice links i and recombine scales

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$
$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla p + \nu \nabla \cdot \left(\nabla(\rho \mathbf{u}) + \left(\nabla(\rho \mathbf{u})\right)^T \right)$$

Definitions for p, c_s and ν emerge from procedure (results on previous slide)

Science and Technology Facilities Council

1. Chen and Doolen, *Annu Rev Fluid Mech* **30** (1), 329–364 (1998)

Using LBE for mesoscale modelling

Can start by treating LBE as mesoscopic CFD solver

- Parameterisation mainly top-down: getting lattice spacing Δx and timestep Δt
- Use expressions for speed of sound c_s and kinematic viscosity ν with known values for given fluid and a choice of τ (relaxation time)
  - τ cannot be too close to $\frac{1}{2}$ (zero viscosity) or calculation will become numerically unstable

Example – water at room temperature (298 K):

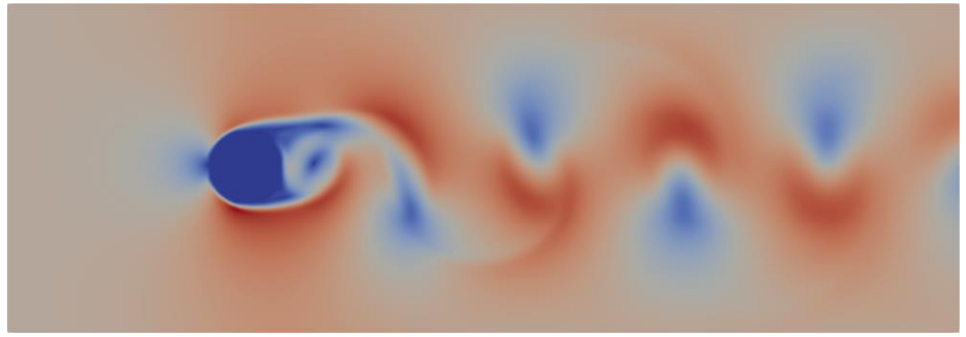
Setting $\tau = 1$
$c_s = 1498$ m s^{-1}
$\nu = 10^{-6}$ m^2 s^{-1}
$\Delta x \approx 2.31 \times 10^{-8}$ m (23.1 nm, 0.0231 μm)
$\Delta t \approx 8.91 \times 10^{-13}$ s (0.891 ps)

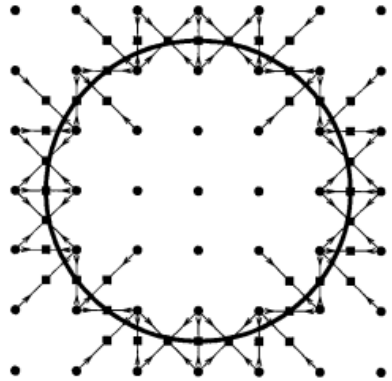
- Density ρ is a relatively free parameter, but usually set to 1 to maximise available calculation precision
- No need to define exactly what particles represent on lattice: we can use LBE for liquids as well as gases

Using LBE for mesoscale modelling

Boundary conditions

- Finding ‘missing’ values of f_i re-entering grid
- Rules can be simple/intuitive and applied at any grid point
  - Bounce-back reflection of f_i to give no-slip boundary[1]
  - ‘Equilibrium forcing’: replacing f_i with $f_i^{eq}(\rho_w, \mathbf{u}_w)$
- More sophisticated boundary conditions available to set ρ_w and/or $\mathbf{u}_w$[2,3,4]
- Can also devise moving boundaries for suspended objects[5,6]





$$f_{-i}(\mathbf{x}_w, t+\Delta t) = f_i(\mathbf{x}_w, t) + 2w_i\rho\mathbf{U}_p \cdot \mathbf{e}_i$$

Can work out force acting on object from f_i at grid points surrounding boundary and determine its motion (translational and rotational)

1. Lavallée *et al.*, *Physica D* **47** (1–2), 233–240 (1991)
2. Inamuro *et al.*, *Phys Fluids* **7** (12), 2928–2930 (1995)
3. Zou and He, *Phys Fluids* **9** (6), 1591–1598 (1997)
4. Halliday *et al.*, *J Phys A* **35** (12), L157–L166 (2002)
5. Ladd, *J Fluid Mech* **271**, 285–309 (1994)
6. Feng and Michaelides, *J Comput Phys* **195** (2), 602–628 (2004)

UKRI Science and Technology Facilities Council

Using LBE for mesoscale modelling

Interactions between multiple fluids and phases

- Can add a force $\mathbf{F}$ acting on a fluid into collisions by:
  - Modifying fluid velocity used in f_i^{eq} [1]
  - Adding a forcing term to the collision[2]
  - A combination of both[3]
- Possible to devise forces based on gradients in densities (or similar properties)
  - Easy to calculate gradients of a property ϕ accurately using a lattice stencil, e.g.

$$c_s^2 \nabla \phi(\mathbf{x}) \approx \sum_i w_i \phi(\mathbf{x} + \mathbf{e}_i \Delta t) \mathbf{e}_i$$

- Multiple fluids modelled on individual lattice grids (occupying same spatial locations) with their own distribution functions

Science and Technology Facilities Council

1. Martys and Chen, *PRE* **53** (1), 743–750 (1996)
2. Kupershtokh and Medvedev, *J Electrostat* **64** (7–9), 581–585 (2006)
3. Guo *et al.*, *PRE* **65** (4), 046308 (2002)

Using LBE for mesoscale modelling

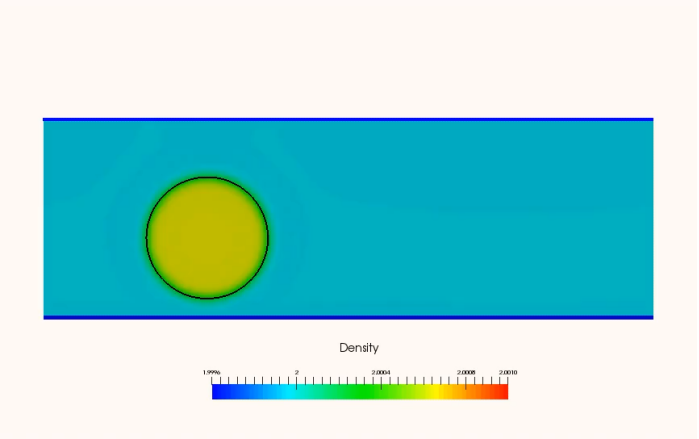
Chromodynamics[1]

- Top-down approach: designed to get correct hydrodynamics and interfacial tension between fluids, originally used for multiple-component lattice gases[2]
- Calculate densities at each lattice point ($\rho_a = \sum_i f_i^a$) and then a 'phase index':

$$\rho^N = \frac{\rho_a - \rho_b}{\rho_a + \rho_b}$$

Pure component a:	$\rho^N = +1$
Pure component b:	$\rho^N = -1$
Interface:	$-1 < \rho^N < +1$

- Forces based on phase index gradients and interfacial tensions
- Fluids collided together 'achromatically' before being re-segregated
  - Combining distribution functions for collision ensures correct hydrodynamics
  - Segregation produces diffuse interfaces, but avoids 'pinning' of drops to lattice and allows larger interfacial tensions



Science and Technology Facilities Council

1. Lishchuk *et al.*, *PRE* **67** (3), 036701 (2003)
2. Gunstensen *et al.*, *PRA* **43** (8), 4320–4327 (1991)

Using LBE for mesoscale modelling

Pseudopotentials[1]

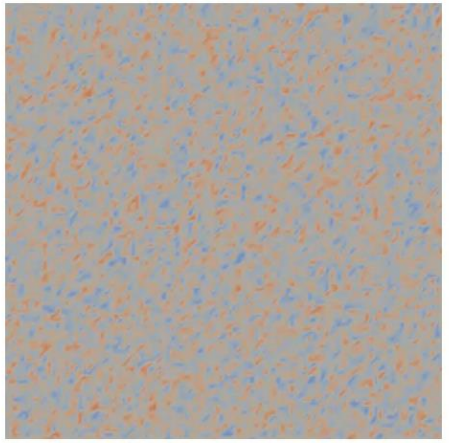
- Makes better connection to microscopic interactions, can be used for single fluid (multiple phases) as well as multiple fluids
- Define 'pseudopotential' $\psi(\rho)$ as function of fluid density for each grid point
- Force on fluid(s) based on gradient(s) of pseudopotential(s):

$$\mathbf{F}^a = -\psi^a(\mathbf{x}) \sum_b g_{ab} \sum_i w_i \psi^b(\mathbf{x} + \mathbf{e}_i \Delta t) \mathbf{e}_i$$

- Form of pseudopotential determines equation of state for fluid(s) …

$$p = \rho c_s^2 + \frac{1}{2} g c_s^2 \psi^2$$

- … or vice versa![2]



Most commonly-used LBE algorithm for multiple phases or fluids, but not without its flaws:

- Spurious flows generated at fluid/phase interfaces
- Often not thermodynamically consistent
- No interfacial tension control

1. Shan and Chen, *PRE* **47** (3), 1815–1819 (1993)
2. Peng and Schaefer, *Phys Fluid* **18**, 042102 (2006)

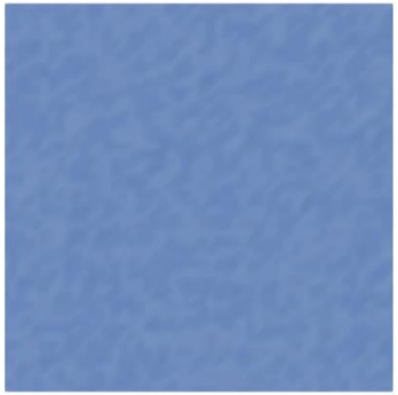
Using LBE for mesoscale modelling

Free-energy LBE[1]

- Modifies f_i^{eq} by adding in bulk pressure and interfacial terms from density gradients:

$$f_i^{eq}(\rho,u) = \rho w_i^{00} + w_i\left[\rho(\mathbf{e}_i\cdot\boldsymbol{u}) + \tfrac{3}{2}\rho(\mathbf{e}_i\cdot\boldsymbol{u})^2 - \tfrac{1}{2}\rho u^2 + f(\lambda,\mathbf{e}_i,\nabla\rho)\right] + w_i^p P_0 - w_i^t\kappa\nabla^2\rho + \kappa\sum_{\alpha,\beta} w_i^{\alpha\beta}\partial_\alpha\rho\partial_\beta\rho$$

- Imposes equation of state via bulk pressure $P_0(\rho,T)$, correct and consistent thermodynamics
- No additional force added to fluid, but need EOS-dependent correction to ensure Galilean invariance
- Density gradients calculated using accurate grid stencils[2]
- Can apply similar (but simpler) local equilibrium distribution function for concentration of two-fluid mixtures, setting chemical potential instead of bulk pressure
  - Harder to apply for more than two fluids



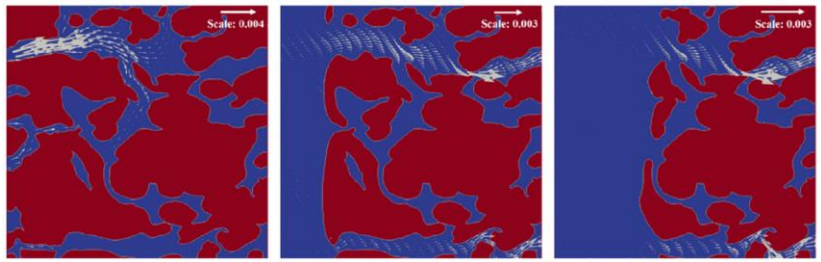
Science and Technology Facilities Council

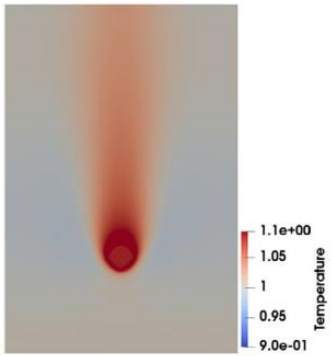
1. Swift *et al.*, *PRL* **75** (5), L157–L166 (2002)
2. Pooley and Furtado, *PRE* **77**, 046702 (2008)

Using LBE for mesoscale modelling

Solute diffusion, reactions and heat transfer

- We can change f_i^{eq} to represent exactly incompressible liquids[1] or truncate to linear velocity terms for low-speed diffusion
- Diffusive solutes can be modelled as additional fluids, but coupled to solvent fluid motion through their own f_i^{eq} [2]
  - Source terms can be added into collisions to represent reaction kinetics[3] and catalytic reactions can be applied at boundaries[4]
- Same approach can be used for heat transfer
  - Simulations of heat convection possible with fluid interactions[2] and combustion with reactions/solid comsumption[5]



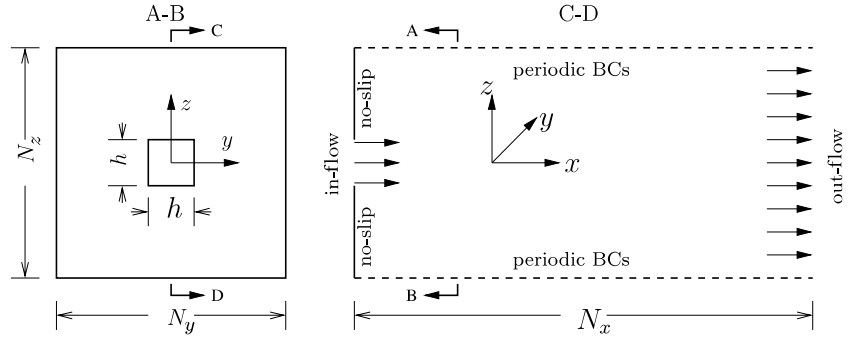


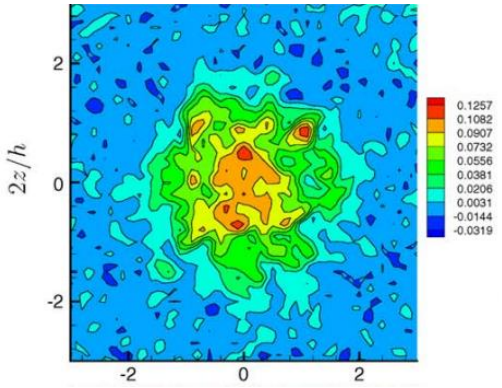
1. He and Luo, *J Stat Phys* **88** (3–4), 927–944 (1997)
2. Inamuro *et al.*, *J Comput Phys* **179** (1), 201–215 (2002)
3. Ponce Dawson *et al.*, *J Chem Phys* **98** (2), 046308 (1993)
4. Arcidiacono *et al.*, *PRE* **78** (4), 046711 (2008)
5. Liu *et al.*, *Combust Flame* **211**, 325–336 (2020)

Using LBE for mesoscale modelling

Rheology, turbulence and viscoelasticity

- To apply a rheological model to a fluid, calculate ν (τ) at each grid point based on its local shear rate
  - Shear rate can be obtained either from velocity gradients or locally from non-equilibrium part of momentum flux tensor (second moment of f_i)[1]
- Turbulence can be applied this way, either with two-equation[2] or Large Eddy Simulation models[3]
- Viscoelasticity also available using similar approach[4]





Science and Technology Facilities Council

1. Boyd *et al.*, *J Phys A* **39** (46), 14241–14247 (2006)
2. Filippova *et al.*, *J Comput Phys* **170** (2), 812–829 (2001)
3. Yu *et al.*, *Comput Fluids* **35** (8), 957–965 (2006)
4. Malaspinas *et al.*, *J Nonnewton Fluid Mech* **165** (23–24), 1637–1653 (2010)

LBE challenges: Collisions

- Simple collisions (BGK) with single relaxation time τ
  - Numerical instabilities for very low viscosities and higher flow speeds
  - No direct control over bulk viscosity

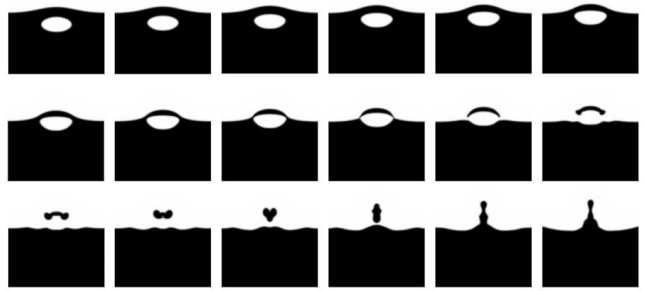
- May need to resort to more complex collisions with additional parameters
  - Two-Relaxation-Time (TRT) collisions[1] work on conjugate f_i values: use additional relaxation time to help reduce numerical instabilities
  - Multiple Relaxation Time (MRT) collisions[2] work on additional f_i moments: bulk viscosity controlled by an additional relaxation time, other relaxation times help eliminate higher-order non-hydrodynamic moments
  - Cascaded LBE (CLBE) collisions[3,4] work on central moments (moments about $\mathbf{e}_i - \mathbf{u}$), also uses higher-order f_i^{eq} – greater stability than other MRT collisions

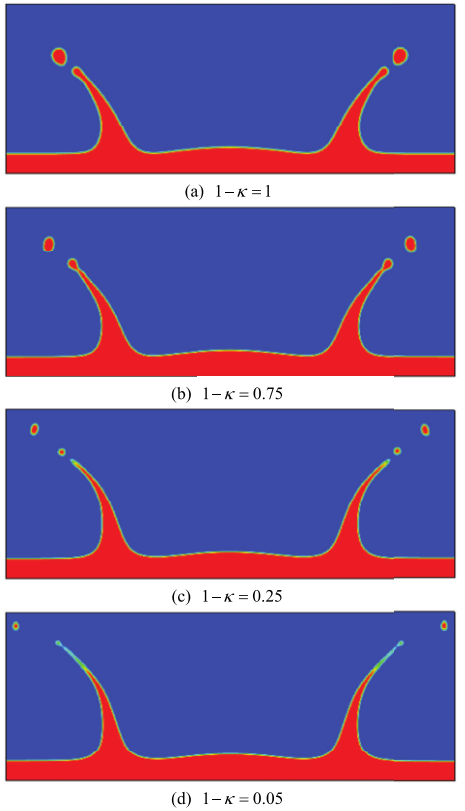
Science and Technology Facilities Council

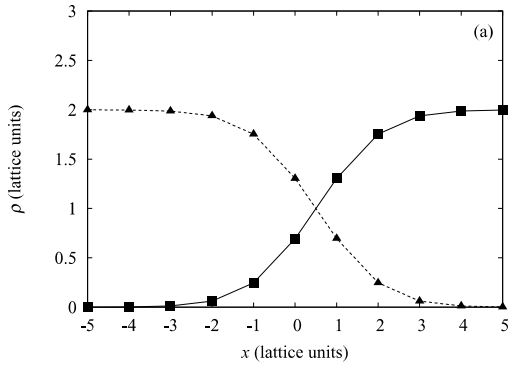
1. Ginzburg *et al.*, *Commun Comput Phys* **3** (2), 427–478 (2008)
2. Lallemand and Luo, *PRE* **61** (6), 6546–6562 (2000)
3. Geier *et al.*, *PRE* **73** (6), 066705 (2006)
4. Lycett-Brown and Luo, *Comput Math Appl* **67** (2), 350–362 (2014)

LBE challenges: Phase/fluid interfaces

- Spurious microcurrents generated at interfaces between fluids or phases
  - Particular problem for large contrasts in density between phases/fluids and/or higher flow speeds
  - Multi-component LBE schemes available to work with larger density contrasts (e.g. gas/liquid)[1]
  - Stencils to calculate gradients more accurately[2]
  - Use of multiple-relaxation-time collisions can help
- Diffuse interfaces: a few lattice points wide[3]
- Interfacial tension not directly controlled in pseudopotential model[4]





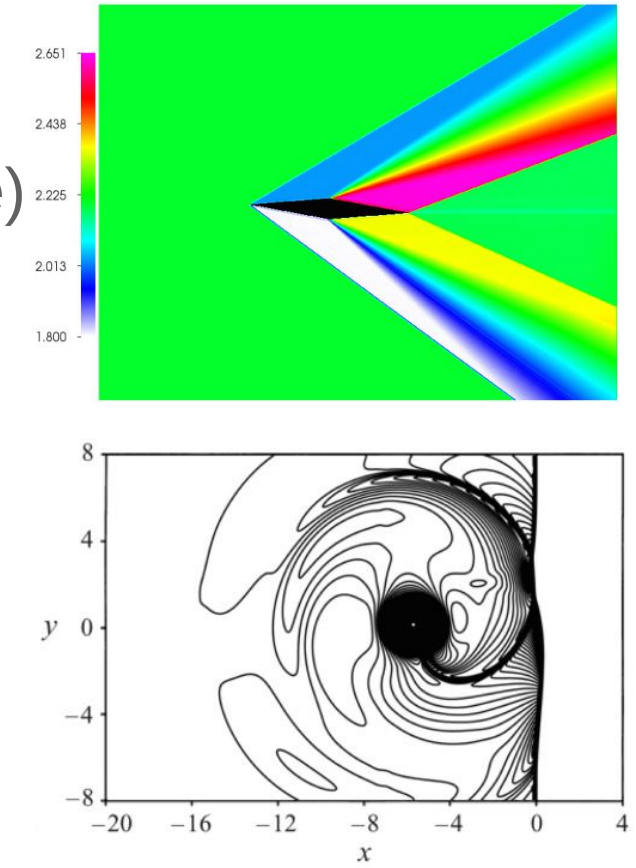


Science and Technology Facilities Council

1. Lishchuk *et al.*, *PRE* **77** (3), 036702 (2008)
2. Pooley and Furtado, *PRE* **77** (4), 046702 (2008)
3. Spencer and Halliday, *PRE* **88** (6), 063305 (2013)
4. Li and Luo, *PRE* **88** (5), 053307 (2013)

LBE challenges: Thermohydrodynamics

- Most LBE lattices use too few links to define temperature
  - Can couple heat transfer effects to fluid flow using separate lattice (but does not solve compressibility issue)
- Low density (rarefied) gas flows with larger ϵ: going beyond Navier-Stokes
  - Slip flows can occur at boundaries even for low Ma[1]
  - Lattice schemes with many more speeds[2]
- Entropic LBE schemes: devise f_i^{eq} to additionally satisfy Boltzmann H theorem[3]
  - Can enable more compressible flows than normally available (including supersonic)



Entropic LBE simulations of supersonic flow around airfoil (top) and vortex interacting with shock wave (bottom)[4]

Science and Technology Facilities Council

1. Zhang *et al.*, *PRE* **71** (4), 047702 (2005)
2. Shan et al., *J Fluid Mech* **550**, 413–441 (2006)
3. Karlin *et al.*, *EPL* **47** (2), 182–188 (1999)
4. Frapolli *et al.*, *Entropy* **22** (3), 370 (2020)

More information about LBE

- Knowledge Centre in DL_Software Digital Guide:

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

  - Includes pages on LBE and further details (how to set scales in LBE calculations, more details on multiple-phase/fluid algorithms, boundary conditions etc.)

Science and Technology Facilities Council

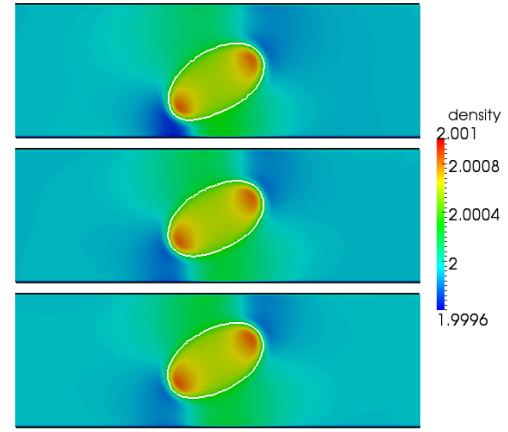
LBE in DL_Software: DL_MESO_LBE

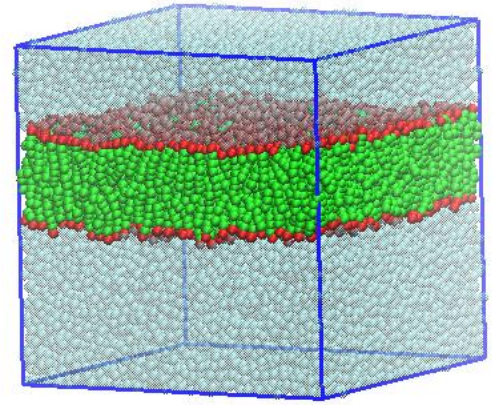
‘Let us think the unthinkable, let us do the undoable, let us prepare to grapple with the ineffable itself, and see if we may not eff it after all.’
- Douglas Adams, *Dirk Gently’s Holistic Detective Agency*

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, 700+ 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

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

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

DL_MESO_LBE

Flexible LBE solver

- Solves discretised Boltzmann equation on regular lattice(s)
- Parallelised mainly by domain decomposition using MPI, but can also use OpenMP multithreading – both share grid points among cores/threads
- Can specify boundary conditions for walls/obstacles, initial conditions and fluid interaction and properties using input files
- Utilities provided to create initial system states, collect up output files created in parallel for visualisation and analysis
    - Also possible to write scripts for simulation setup and post-processing

DL_MESO_LBE: current functionalities

Square lattice grids (two- and three-dimensional)

- D2Q9, D3Q15, D3Q19, D3Q27

Multiple fluids available

- Weakly compressible or fully incompressible fluids
- Can couple in solute mass transfers and/or heat transfers

Collision schemes

- BGK single relaxation time (default)[1]
- Two Relaxation Time (TRT)[2]
- Multiple Relaxation Time (MRT)[3]
- Cascaded Lattice Boltzmann (CLBE)[4]

Different schemes to apply external and interfacial forces available for all collision schemes

Science and Technology Facilities Council

1. Qian *et al.*, *EPL* **17** (6), 479–484 (1992)
2. Ginzburg *et al.*, *Comm Comp Phys* **3** (2), 427–478 (2008)
3. Lallemand and Luo, *PRE* **61** (6), 6546–6562 (2000)
4. Geier et al., *PRE* **73**, 066705 (2006)

DL_MESO_LBE: current functionalities

Boundary conditions

- No-slip bounce-back conditions (simplest)
- Outflows
- Fixed velocity or density (Zou/He[1], Inamuro[2], regularised[3], kinetic[4] BCs)
- Fixed concentration and temperature (Zou/He, Inamuro)
- Surface wetting available

Mesoscopic interactions

- Shan/Chen pseudopotential for any number of fluids
  - Cubic and hard-sphere equations of state
- Lishchuk (chromodynamic) continuum-based for two or more fluids
- Swift free-energy for one/two fluid systems
  - Cubic and hard-sphere equations of state

1. Zou and He, *Phys Fluid* **9** (6), 1591–1598 (1997)
2. Inamuro *et al.*, *Phys Fluid* **7** (12), 2928–2930 (1995)
3. Latt *et al.*, *PRE* **77**, 056703 (2008)
4. Ansumali and Karlin, *PRE* **66**, 026311 (2002)

DL_MESO_LBE: current functionalities

Non-Newtonian rheological models

- Uses momentum flux tensors to calculate shear rates locally, used as inputs for rheological models
- Power law fluids, Bingham/Herschel-Bulkley plastics, Casson and Carreau-Yasuda fluids

Parallel file-writing

- Options to gather data along Cartesian axes to reduce number of output files produced (normally one file per core per simulation snapshot)
- Gathering in all three directions invokes MPI-IO – generates one file per simulation snapshot

Science and Technology Facilities Council

1. Zou and He, *Phys Fluid* **9** (6), 1591–1598 (1997)
2. Inamuro *et al.*, *Phys Fluid* **7** (12), 2928–2930 (1995)
3. Latt *et al.*, *PRE* **77**, 056703 (2008)
4. Ansumali and Karlin, *PRE* **66**, 026311 (2002)

Future DL_MESO_LBE functionalities

- Non-spherical drops (fluid-filled vesicles)[1]
  - Adaptation of chromodynamic (Lishchuk) algorithm to include conservation of membrane area and bending
  - Useful for modelling e.g. red blood cells in haemodynamic flows
- Suspended solid object dynamics[2]
  - Calculating forces on objects and integrating to determine motion
  - Opens up coupling between MD/DPD and LBE
- Adaptive mesh LBE
  - Creating finer grids (smaller Δx) close to interfaces for greater accuracy, e.g. curved boundaries
  - Coupling together fine and coarse grids

Science and Technology Facilities Council

1. Spendlove *et al.*, *Phys Fluid* **33**, 077110 (2021)
2. Ladd, *J Fluid Mech* **271**, 285–309 (1994)

LBE units

Based upon lattice (both set to 1 inside DL_MESO_LBE)

- Grid spacing Δx
- Time step Δt

Relaxation time τ and 'real-life' quantities can be used to find lattice units

- Speed of sound $c_s = \frac{1}{\sqrt{3}} \frac{\Delta x}{\Delta t}$
- Kinematic viscosity $\nu = \frac{\mu}{\rho} = \frac{1}{3}\left(\tau - \frac{1}{2}\right) \frac{(\Delta x)^2}{\Delta t}$

Density in LBE calculation a 'free' parameter

- Multiplier in local equilibrium distribution function f_i^{eq}
- Usually set to (or around) 1 to maximise calculational accuracy

DL_MESO_LBE input/output files

Input files

- **lbin.sys** Simulation parameters/settings
- **lbin.spa** Boundary conditions
- **lbin.init** Initial conditions/configuration

Output files

- Screen System diagnostics (can re-direct to file)
- **lbout*.vts** Plottable results in VTK format (default: other options available)
- **lbout.dump** Simulation restart capability
- **lbout.info** Simulation information for post-processing after parallel run
  lbout.ext

Ibin.sys – Simulation parameters/settings

- Mandatory simulation control and system definition
  - Similar to CONTROL and FIELD in DL_MESO_DPD/DL_POLY
- Each line is a key phrase followed by a number/word
  - First 10 in bold **must** be specified
  - Parameters not required can be omitted
  - No specific line order
- Can specify
  - Initial/boundary velocities, fluid densities, solute concentrations, temperature
  - Mesoscopic interaction and rheological models/parameters

Science and Technology Facilities Council

Chapter 6 in DL_MESO User Manual gives full list of available key phrases

space_dimension	**2**
discrete_speed	**9**
number_of_fluid	**2**
number_of_solute	**0**
temperature_scalar	**0**
phase_field	**0**
grid_number_x	**150**
grid_number_y	**50**
grid_number_z	**1**
domain_boundary_width	**1**
incompressible_fluids	0
collision_type	BGKGuo
interaction_type	Lishchuk
output_format	VTK
total_step	400000
equilibration_step	5000
save_span	500
noise_intensity	0.0
sound_speed	540.0
kinetic_viscosity	0.001
relaxation_fluid_0	1.0
relaxation_fluid_1	1.0
bulk_relaxation_fluid_0	1.0
bulk_relaxation_fluid_1	1.0
interaction_0	0.0
interaction_1	0.0034
interaction_2	0.0034
interaction_3	0.0
segregation	1.4

lbin.spa – Boundary conditions

```
131 49 0 149
132 49 0 149
133 49 0 149
134 49 0 149
135 49 0 149
136 49 0 149
137 49 0 149
138 49 0 149
139 49 0 149
140 49 0 149
141 49 0 149
142 49 0 149
143 49 0 149
144 49 0 149
145 49 0 149
146 49 0 149
147 49 0 149
0 49 0 149
149 49 0 149
1 0 0 13
148 0 0 13
2 0 0 13
3 0 0 13
4 0 0 13
5 0 0 13
6 0 0 13
```

- Mandatory definition of boundary conditions
  - A blank file still needed even if all BCs are periodic
- Four numbers per line representing a grid point

  `x y z [BC code]`
- BC codes listed in chapter 6 of DL_MESO user manual
  - Blank sites, on-grid and mid-grid bounce-back
  - Outflow boundaries
  - Fixed velocity/fluid density, solute concentration, temperature
- BC codes for outflow and fixed velocity/density/concentration/ temperature boundaries are directional, blank sites and bounce-back are not
- Only need to specify BC code for a grid point if its boundary condition is not periodic
- Easiest to create **lbin.spa** using DL_MESO GUI or a script

lbin.init – Initial conditions

```
26 15 0 0.0 0.0 0.0 0.0 2.0
26 16 0 0.0 0.0 0.0 0.0 2.0
26 17 0 0.0 0.0 0.0 0.0 2.0
26 18 0 0.0 0.0 0.0 0.0 2.0
26 19 0 0.0 0.0 0.0 0.0 2.0
26 20 0 0.0 0.0 0.0 0.0 2.0
26 21 0 0.0 0.0 0.0 0.0 2.0
26 22 0 0.0 0.0 0.0 0.0 2.0
26 23 0 0.0 0.0 0.0 0.0 2.0
26 24 0 0.0 0.0 0.0 0.0 2.0
26 25 0 0.0 0.0 0.0 0.0 2.0
27 13 0 0.0 0.0 0.0 0.0 2.0
27 14 0 0.0 0.0 0.0 0.0 2.0
27 15 0 0.0 0.0 0.0 0.0 2.0
27 16 0 0.0 0.0 0.0 0.0 2.0
27 17 0 0.0 0.0 0.0 0.0 2.0
27 18 0 0.0 0.0 0.0 0.0 2.0
27 19 0 0.0 0.0 0.0 0.0 2.0
27 20 0 0.0 0.0 0.0 0.0 2.0
27 21 0 0.0 0.0 0.0 0.0 2.0
27 22 0 0.0 0.0 0.0 0.0 2.0
27 23 0 0.0 0.0 0.0 0.0 2.0
27 24 0 0.0 0.0 0.0 0.0 2.0
27 25 0 0.0 0.0 0.0 0.0 2.0
27 26 0 0.0 0.0 0.0 0.0 2.0
27 27 0 0.0 0.0 0.0 0.0 2.0
28 12 0 0.0 0.0 0.0 0.0 2.0
28 13 0 0.0 0.0 0.0 0.0 2.0
28 14 0 0.0 0.0 0.0 0.0 2.0
```

- **Optional** simulation initialisation
- Each line specifies grid point coordinate and properties for that grid point: velocity, fluid densities, solute concentrations and temperature
  - Similar format to **lbin.spa**:

$$\mathbf{x}\ \mathbf{y}\ \mathbf{z}\ \mathbf{v}_x\ \mathbf{v}_y\ \mathbf{v}_z\ \ \rho_0\ \rho_1\ \cdots$$

  - All properties for system must be specified in each line
- Values for specified points used in place of default system-wide values given in lbin.sys
- Utilities supplied with DL_MESO can be used to create lbin.init
  - **lbeinitcreate.cpp** (**init.exe**) – adds spherical/circular drops for new simulations
  - **lbedumptoinit.cpp** (**dump_to_init.exe**) – creates lbin.init from lbout.dump restart file
- Also possible to devise scripts to write this file

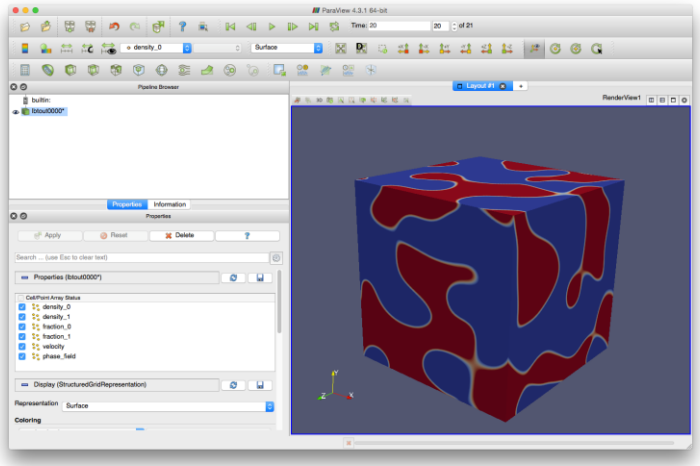
Screen/standard output – Information

- DL_MESO_LBE outputs information about calculation to screen or standard output
  - Can pipe this to a file
  - Similar to DL_MESO_DPD/DL_POLY OUTPUT
- Starts with some information about simulation (e.g. grid size, numbers of fluids)
- Writes out masses of fluids and system momentum at user-specified intervals
  - Useful as diagnostic for LBE calculation, e.g. checking progress towards steady-state
- At end of calculation, prints timing information and efficiency measure: Millions of Lattice Updates Per Second (MLUPS)

```
Welcome to DL_MESO_LBE

UKRI STFC/CCP5 Program Library Package
UKRI STFC Daresbury Laboratory Lattice Boltzmann Equation program

DL_MESO version 2.7 rev 01, January 2019
Authors:       R. S. Qin & M. A. Seaton
Contributors:  J. Meng, L. Grinberg & M. A. Johnston
Copyright (c) 2019 UKRI STFC Daresbury Laboratory, United Kingdom

Any comments or queries, please contact Dr. M. A. Seaton at:
   Email: michael.seaton@stfc.ac.uk
   Tel:   +44 (0)1925 603317

Program started at: Sun Jan  6 07:58:28 2019

armed with D2Q9 LB model
================================================================================
2-dimensional system with grid size: nx=150, ny=50
boundary width = 0
system includes 2 compressible phases, 0 solute and no temperature scalar
D2Q9 lattice Boltzmann model is defined using BGK collisions
with Guo forcing and Lishchuk interactions
--------------------------------------------------------------------------------
Running with 8 threads
--------------------------------------------------------------------------------
fluid 0  :               simple Newtonian fluid
                         relaxation time            = 1
fluid 1  :               simple Newtonian fluid
                         relaxation time            = 1

interaction parameters:  fluid i  fluid j          g_ij        beta_ij
                               0        1        0.0034           1.4

wall interaction parameters:      fluid i          g_wi
                                  1                0
--------------------------------------------------------------------------------
0            MASS: total = 14700, fluid 0 = 13306, fluid 1 = 1394
0.0101249    MOMENTUM: x: -2.04003e-13, y: -2.04003e-13
500          MASS: total = 14700, fluid 0 = 13306, fluid 1 = 1394
0.665448     MOMENTUM: x: 5.97018e-12, y: 0.0313264
1000         MASS: total = 14700, fluid 0 = 13306, fluid 1 = 1394
1.31106      MOMENTUM: x: 5.75585e-12, y: 0.0357875
```

lbout*.vts – Simulation snapshots/trajectory

- Snapshots of simulation at user-specified intervals as trajectory
  - Similar to HISTORY files in DL_MESO_DPD and DL_POLY
  - Includes macroscopic properties for each grid point (velocities, densities, concentrations, temperatures) and boundary condition
- Default format: XML-based Structured Grid VTK
  - Legacy VTK and Plot3D formats also available
  - Binary and text variants available
  - All formats readable in ParaView
- At least one file produced per snapshot: by default, each core writes its own files (to be gathered afterwards)
  - Strongly recommend using data gathering options in **lbin.sys** when running in parallel to reduce number of files



www.paraview.org

lbout.dump – Simulation restart

- Binary file with distribution functions (and local relaxation times when using rheological models) for all lattice points
  - Similar to export/REVCON and REVIVE for DL_MESO_DPD/DL_POLY
  - DL_MESO_LBE can read this file to resume a previous calculation
  - **lbedumpvtk.cpp** (**dump_to_vtk.exe**) utility creates VTK file of simulation state in lbout.dump for visualisation (and simulation diagnostics)
  - **lbedumptoinit.cpp** (**dump_to_init.exe**) utility creates **lbin.init** as initial state for new calculation

More information available in DL_Software Digital Guide about DL_MESO_LBE files and formats:

https://dl-sdg.github.io/RESOURCES/TUTORIALS/dlm_3.html

and utilities:

https://dl-sdg.github.io/RESOURCES/TUTORIALS/dlm_6.html

LBE Tutorial Exercises

- Three available in DL_Software Digital Guide, all designed to work with DL_MESO_LBE

- **Phase separation and equations of state**
  - Setting up Shan/Chen pseudopotential interactions to model vapour-liquid coexistence from a cubic equation of state
- **Drop flows**
  - Using chromodynamics to model drops undergoing linear shear, examining effects of shear rate and interfacial tension

LBE Tutorial Exercises

- **Pressure-driven flows and Kármán vortex streets**
  - Applying channel flows with obstacles

- All three exercises use DL_MESO_LBE with input files supplied, some Python scripts provided for data analysis (Phase separation) and boundary condition building (Pressure-driven flows), and ParaView for visualisation

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

Science and
Technology
Facilities Council

Questions?

UKRI Science and Technology Facilities Council

Thank you

Science and Technology Facilities Council @STFC_matters Science and Technology Facilities Council