Parallel Evaluation of Pair Forces for Molecular Dynamics in Arbitrary Geometries

Graham B. Macpherson and Jason M. Reese

Department of Mechanical Engineering, University of Strathclyde, Glasgow, UK,
www.mecheng.strath.ac.uk/multiscaleflows
graham.macpherson@strath.ac.uk

ABSTRACT

The new Arbitrary Interacting Cells Algorithm (AICA) for calculating intermolecular pair forces for Molecular Dynamics (MD) on a distributed parallel computer is presented. AICA is designed to operate on geometrical domains defined by an unstructured, arbitrary polyhedral mesh, which has been spatially decomposed into irregular portions for parallelisation. It is intended for nano scale fluid mechanics simulation by MD in complex geometries, and to provide the MD component of a hybrid MD/continuum simulation. AICA has been implemented in OpenFOAM and verified against a published MD code.

Keywords: nanofluidics, molecular dynamics, hybrid simulation, pair force evaluation, parallel computing

1 INTRODUCTION AND MOTIVATION

1.1 MD Simulation in Arbitrary Geometries

Simulations of nano scale liquid systems can provide insight into many naturally-occurring phenomena, such as the action of proteins that mediate water transport across biological cell membranes. Nanoflow simulations may also facilitate the design of future nano devices and materials (e.g. high-throughput, highly selective filters or lab-on-a-chip components). The dynamics of these very small systems are dominated by surface interactions, due to their large surface area to volume ratio. However, these surface effects are often too complex and material-dependent to be treated by simple phenomenological parameters [1] or by adding 'equivalent' fluxes at the boundary [2]. Direct simulation of the fluid using molecular dynamics (MD) offers the ability to model these phenomena with minimal simplifying assumptions.

Some MD fluid dynamics simulations have been reported [3, 4], but MD is prohibitively computationally costly for simulations of systems beyond a few tens of nanometers in size. Fortunately, the molecular detail of the full flow-field that MD simulations provide is often unnecessary; in liquids, beyond 5–10 molecular diameters ($\lesssim$ 3nm for water) from a solid surface, the continuum-fluid approximation is valid and the Navier Stokes equations with bulk fluid properties may be used [1, 5, 6]. Hybrid simulations have been proposed [7–10] to simultaneously take advantage of the accuracy and detail provided by MD in the regions that require it, and the computational speed of continuum mechanics in the regions where it is applicable. In order to produce a useful, general simulation tool for hybrid simulations, the MD component must be able to model complex geometrical domains. This capability does not exist in currently available MD codes; domains are simple shapes, usually with periodic boundaries. The most important, computationally demanding, and difficult aspect of any MD simulation is the calculation of intermolecular forces. This paper describes an algorithm that calculates pair force interactions between molecules occupying arbitrary, unstructured mesh geometries that have been parallelised for distributed computing by spatial decomposition.

1.2 Neighbour Lists are Unsuitable

The conventional method of MD force evaluation in distributed parallel computation is to use 'neighbour lists' for interacting molecule pairs with the 'replicated molecule' method of providing interactions across periodic and interprocessor boundaries, where the system has been spatially partitioned [11, 12].

The spatial location of molecules in MD is dynamic, and hence not deducible from the data structure that contains them. A neighbour list defines which pairs of molecules are within a certain distance of each other, and so need to interact via intermolecular forces.

Neighbour lists are unsuitable when considering systems of arbitrary geometry, that may have been divided into irregular and complex mesh segments using standard mesh partitioning techniques (see, for example, figure 1) for two reasons:

1. **Interprocessor molecule transfers:** A molecule may cross an interprocessor boundary at any point in time (even part of the way through a timestep), at which point it should be deleted from the processor it was on and an equivalent molecule created on the processor on the other side of the

NSTI-Nanotech 2007, www.nsti.org, ISBN 1420061828 Vol. 1, 2007

boundary. Given that neighbour lists are constructed as lists of array indices, references or pointers to the molecule's location in a data structure, deleting a molecule would invalidate this location and require seaching to remove all mentions of it. Likewise, creating a molecule would require the appropriate new pair interactions to be identified. Clearly this is not practical. It is conventional to allow molecules to 'stray' outside of the domain controlled by a processor and carry out interprocessor transfers (deletions and creations) during the next neighbour list rebuild. This is only possible when the spatial region associated with a processor can be simply defined by a function relating a position in space to a particular cell on a particular processor (i.e. a uniform, structured mesh, representing a simple domain). In a geometry where the space in question is defined by a collection of individual cells of arbitrary shape, this is not possible. For example the location the molecule has 'strayed' to may be on the other side of a solid wall on the neighbour processor, or across another interprocessor boundary. For the molecule to end up unambiguously in the correct place, the interprocessor transfer must happen as the molecule crosses the boundary.

2. **Spatially resolved flow properties:** MD simulations used for flow studies must be able to spatially resolve fluid mechanic and thermodynamic fields. This is achieved by accumulating and averaging measurements of the properties of molecules in individual cells of the mesh that defines the geometry. If a molecule is allowed to stray outside of the domain controlled by a processor, as above, then it would not be unambiguous and automatic which cell's measurement the molecule should contribute to.

Our new algorithm is of comparable computational cost to neighbour lists, but designed to be powerful and generic for simulation in arbitrary geometries [13].

2 ARBITRARY INTERACTING CELLS ALGORITHM (AICA)

2.1 Replicated Molecule Periodicity and Parallelisation

When parallelising an MD simulation, the spatial domain is decomposed and each processor is given responsibility for an individual region [11]. Processors must communicate to carry out intermolecular force calculations, where molecules close to processor boundaries need to be replicated on their neighbours to provide interactions. Periodic boundaries also require information about molecules that are not adjacent physically

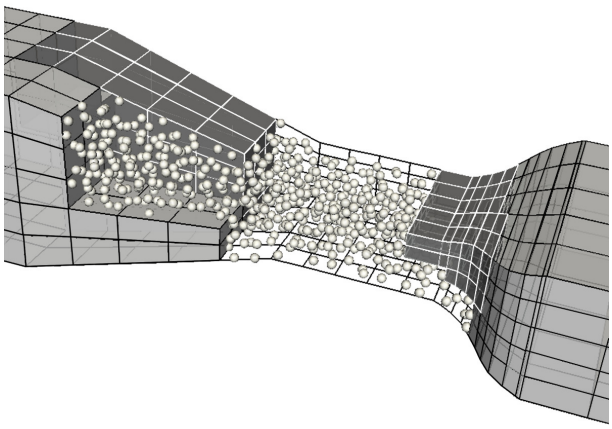

Figure 1: Example of a molecular dynamics flow simulation in a nano channel. The mesh is decomposed into five irregular portions, four of which are shown as shaded blocks, the molecules contained in the fifth portion are shown explicitly.

in the domain; these required interactions can also be constructed by creating copies of molecules outside the boundary. It is possible to handle processor and periodic boundaries in exactly the same way, since they have the same underlying objective: molecules close to the edge of a region need to be copied either between processors or to other locations on the same processor at every timestep to provide interactions. This is a useful feature because decomposing a mesh for parallelisation will often turn a periodic boundary into a processor boundary. The issue is how to efficiently identify which molecules need to be copied, and to which location, because this set continually changes as the molecules move.

2.2 Interacting Cell Identification

Our new AICA algorithm is an extension of the Conventional Cells Algorithm (CCA) [11]. In the CCA, a simple (usually cuboid) simulation domain is subdivided into equally sized cells. For computational and theoretical reasons [14], intermolecular potentials do not extend to infinity, they are assigned a cut-off radius, r_{cut}, beyond which they are set to zero. A molecule must calculate intermolecular force contributions from all other molecules within a distance of r_{cut}. The minimum dimension of the CCA cells must be greater than r_{cut}, so that all molecules in a particular cell interact with all other molecules in their own cell and with those in their nearest neighbour cells (i.e. those they share a face, edge or vertex with — 26 in 3D). AICA uses the same type of mesh as would be used in Computational Fluid Dynamics (CFD) to define the geometry of a region. *There are no restrictions on cell size, shape or connectivity.* Each cell has a unique list of other cells that it is to interact with, this list is known as the Direct Interaction List

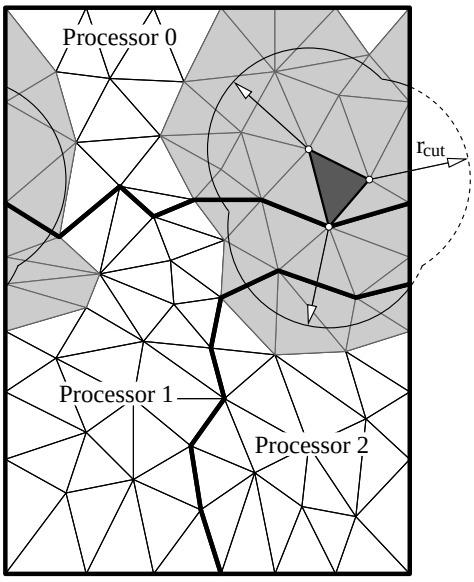

Figure 2: Interacting cell identification using unstructured triangular cells to demonstrate insensitivity to mesh topology. The spatial domain is periodic top-bottom and left-right. Real cells within r_{cut} that interact with the CIQ (dark) are shaded in grey. Some of these may lie on the opposite side of a periodic boundary, or on a different (possibly non-adjacent) processor when the domain has been decomposed. Realistic systems would be significantly larger compared to r_{cut} than shown here.

(DIL) for the Cell In Question (CIQ). It is constructed by searching the mesh to create a set of cells that have at least one vertex within a distance of r_{cut} from any of the vertices of the CIQ, see figure 2.

The DILs are established prior to the start of simulation and are valid throughout because the spatial relationship of the cells is fixed, whereas the set of molecules they contain is dynamic. In a similar way to the CCA, at every timestep a molecule in a particular cell calculates its interactions with the other molecules in that cell and consults the cell's DIL to find which other cells contain molecules it should interact with. Information is required to be maintained stating which cell a molecule is in — this is straightforward and computationally cheap.

Replicated molecule parallelisation and periodic boundaries are handled in the same way using *referred molecules* and *referred cells*. Referred molecules are copies of real molecules that have been placed in a region outside a periodic or processor boundary in order to provide the correct intermolecular interaction with molecules inside the domain. Referred cells are created once, at the start of the simulation [13], see figure 3. They define a region of space and hold a collection of referred molecules. The forces between real molecules in the mesh and the referred molecules provides the intermolecular force link between processors and across

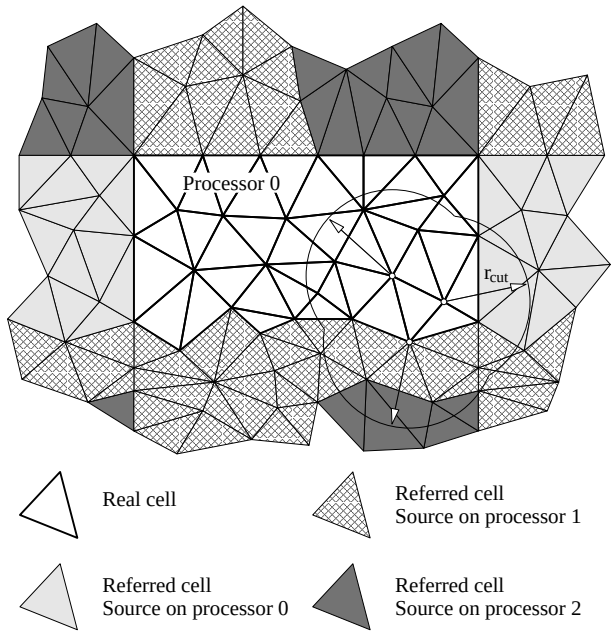

Figure 3: The configuration of referred cells around the portion of mesh on processor 0. A circle of radius r_{cut} drawn from any vertex of a real cell would be fully encompassed by other real or referred cells, thereby providing all molecules in that cell with the appropriate intermolecular interactions.

periodic boundaries. Each referred cell knows

- which real cell in the mesh (on which processor) is its source;
- the required transformation to refer the positions and orientations of the real molecules in the source cell to the referred location;
- which real cells are in range of this particular referred cell and hence require intermolecular interactions to be calculated. This is constructed once at the start of the simulation in the same way as the DIL for real cell interactions — the vertices of the real cells in the portion of mesh on the same processor as the referred cell are searched, those with at least one vertex in range of any vertex of the referred cell are found.

At each timestep, the referred cells are populated with referred molecules, copied from their (real) source cells. These are discarded after the force calculation.

3 VERIFICATION

The AICA algorithm, implemented in OpenFOAM [15] is run on a test case in serial on a PC and decomposed into irregular shaped portions and run as a distributed parallel calculation on 24 processors of a cluster. These results are compared to an in-house code which has been validated against those supplied in [11].

NSTI-Nanotech 2007, www.nsti.org, ISBN 1420061828 Vol. 1, 2007

Figure 4 shows the average Total Energy (TE) per molecule, for a cubic domain containing 27000 Lennard Jones molecules, starting in a simple cubic lattice at a temperature of 120K. TE for the AICA simulations shows a very similar trend to the in-house code. The in-house code and AICA simulations do not follow exactly the same trajectory because the velocities in the systems are initialised by random numbers, which are different between the two codes. This is likely to account for the slight (0.0025%) difference in average TE. The serial and parallel tests use the same initial velocity configuration and each value for TE is the same to better than 1 part in 10^{12}. Momentum is conserved in both serial and parallel tests to approximately 1 part in 10^{16}.

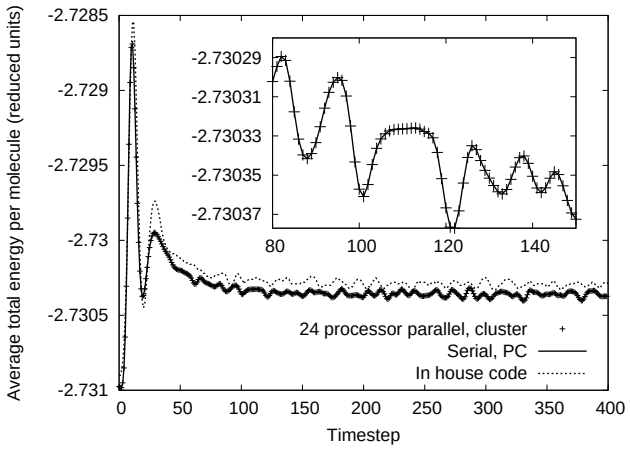

Figure 4: Verification of total energy results. Serial and parallel results are identical, see inset.

4 CONCLUSIONS

Our Arbitrary Interacting Cell Algorithm (AICA) has been described, which is a new way of calculating pair forces in a molecular dynamics simulation, carried out on arbitrary unstructured meshes that have been spatially decomposed to run in parallel. The algorithm is expected to achieve a similar computational speed to the neighbour list method, but does not suffer from the degradations in performance that neighbour lists experience in large systems and at high temperatures [13]. AICA has been implemented in the OpenFOAM code, and produces the same results when a system is simulated in serial, or in parallel on a mesh decomposed into irregular shapes on 24 processors. The exchange of intermolecular force information across interprocessor and boundaries must therefore be correct.

ACKNOWLEDGEMENTS

The authors would like to thank Chris Greenshields and Matthew Borg of Strathclyde University (UK), and Henry Weller and Mattijs Janssens of OpenCFD Ltd. (UK) for useful discussions. This work is funded in the UK by Strathclyde University and the Miller Foundation, and through a Philip Leverhulme Prize for JMR from the Leverhulme Trust.

REFERENCES

[1] J. Koplik and J. R. Banavar. *Annual Review of Fluid Mechanics*, 27, 257–292, 1995.

[2] H. Brenner and V. Ganesan. *Physical Review E*, 61, 6879–6897, 2000.

[3] D. Hirshfeld and D. Rapaport. *Physical Review Letters*, 80, 5337–5340, 1998.

[4] K. Travis, B. Todd and D. Evans. *Physical Review E*, 55, 4288–95, 1997.

[5] T. Becker and F. Mugele. *Physical Review Letters*, 91, 166104, 2003.

[6] H. Okumura and D. M. Heyes. *Physical Review E*, 70, 061206, 2004.

[7] R. Delgado-Buscalioni and P. V. Coveney. *Philosophical Transactions of the Royal Society London*, 362, 1639–1654, 2004.

[8] G. Wagner and E. G. Flekkøy. *Philosophical Transactions of the Royal Society London*, 362, 1655–1665, 2004.

[9] X. B. Nie, S. Y. Chen, W. E and M. O. Robbins. *Journal of Fluid Mechanics*, 500, 55–64, 2004.

[10] T. Werder, J. H. Walther and P. Koumoutsakos. *Journal of Computational Physics*, 205, 373–390, 2005.

[11] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2nd edition, 2004.

[12] D. Rapaport. *Computer Physics Communications*, 62, 217–228, 1991.

[13] G. B. Macpherson and J. M. Reese. *Submitted to Journal of Computational Physics*, 2007.

[14] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon Press, 1994.

[15] "OpenFOAM, www.openfoam.org."