Scalable coupling of Molecular Dynamics (MD) and Direct Numerical Simulation (DNS) of Multi-scale Flows — Part 2

Edward Smith, David Trevelyan and Tamer A. Zaki

Mechanical Engineering, Imperial College London
London SW7 2AZ, United Kingdom

May 31, 2013

Abstract

The objective of this project was the development and performance optimisation of a coupling application. The coupler library is intended for interfacing massively-parallel algorithms for multi-physics simulations. The development of the coupler library adopted the same philosophy of the Message Passing Interface (MPI) library: The coupler was engineered as a set of libraries that are accessible from various applications, in order to interface their operation. However, the applications maintain independent data and scope, and only exchange information via calls to the coupler library. The development, validation, verification and optimisation of the coupling library were performed in the context of interfacing two massively parallel algorithms: a continuum Navier-Stokes solver ($\mathcal{T}ransFlow$) and a molecular dynamics algorithm ($\mathcal{S}treamMD$). However, the development of the coupling library has maintained generality in order to facilitate coupling other application softwares in the future.

Contents

1 Introduction

1.1 Background

The continued development of large-scale scientific computing has enabled unprecedented simulations of complex physics. These simulations rely on advanced computational techniques, particularly developed for accuracy, efficiency and scalability on thousands of computational cores. In addition, the mathematical details, computational algorithm, software design, implementation and optimisation of these codes are often subject specific. Therefore, interfacing these algorithms with the objective of exploring multi-physics problems can be a significant algorithmic and computational challenge. In this project, a coupler library is developed which is used to interface massively-parallel algorithms. The development vehicle for this library was multiscale simulations of continuum-molecular coupled flows.

Many current and emerging technologies rely on flow phenomena which span a wide range of length scales – from the atomistic to the macroscopic level. In such multi-scale flows, the motion of individual atoms near the surface of a wall can determine the bulk motion of the fluid, and involve dynamics which cover more than than six orders of magnitude. A variety of applications, such as electro-kinetic flow in microfluidic channels or the wetting of superhydrophobic surfaces, depend on this complex interaction between the nanoscale and macroscale physics. Thus, the ability to model and understand such multiscale phenomena can lead to significant advances in microfluidic control, and drag reducing surface coatings, among a host of other multiscale technologies.

Current computational approaches have focused on either simulating the dynamics of large groups of atoms, or the macroscopic flow behaviour independently. At the small end of the spectrum, Molecular Dynamics (MD) simulations track the motion of individual atoms based on their mutual interactions inside a confined domain. This approach has been successful in determining the transport and shear properties of liquids [5, 6]. On the other hand, Direct Numerical Simulations (DNS) employ continuum-level assumptions to solve the non-linear, partial differential equations for the fluid velocity at all points within a large computational domain. For instance, the DNS code $\mathcal{T}rans\mathcal{F}low$ by Zaki and co-workers [13, 14, 15] has been used in investigating purely hydrodynamic phenomena such as the interaction of vortical modes with turbulence. However, to date, few attempts have been made to combine both methodologies within a single framework, in particular in the context of large-scale, massively-parallel scientific computing.

The goal of the current project is to develop, validate, and optimise a state-of-the-art computational framework to efficiently couple simulations of various physical phenomena. The particular development within the project was focused on coupling MD simulations with high-fidelity DNS for realistic simulations. The combined MD-DNS code provides new capabilities and insight into how phenomena at the molecular scale, such as microscopic surface textures and coatings, can influence macroscopic hydrodynamic motions. The ability to resolve the entire range of scales will also obviate the need for simplified, and often inaccurate, constitutive models for molecular scale effects.

1.2 The continuum algorithm: $\mathcal{T}rans\mathcal{F}low$

The continuum algorithm solves the full non-linear, incompressible Navier-Stokes equations in curvilinear coordinates using a control volume formulation. While the details of the numerical method are immaterial to the current project, it is important to summarise a few of the algorithmic features of the solver that are relevant to coupling to the MD code. For example, use of the control volume approach ensure conservation properties are maintained by the solver, and facilitates accurate coupling. The algorithm has been used in prior high-fidelity investigations of stability of shear flows, where accuracy is paramount [13, 14]. Also, due to the computational cost, efficient algorithms for the solution of the elliptic pressure equations are required.

This code has been benchmarked on the US Department of Energy (DoE) supercomputers,

the supercomputing centre in Germany (HLRB) and, most importantly, on HECToR. As part of the UK-Turbulence Consortium (UKTC), $\mathcal{T}ransFlow$ was verified by Daresbury laboratory for performance and scalability on HECToR. It was also adopted for the most recent massively parallel simulations of turbulent flows on HECToR [7], and demonstrated scalability up to 10^3 cores.

The parallel implementation of the numerical scheme is via Message Passing Interface (MPI). Domain decomposition is used to divide the solution domain among MPI processes. The choice of the domain size per MPI process is guided by minimising surface area to volume ratio, in order to maximise computations relative to data exchanges. Performance testing was carried out using a Cartesian mesh for turbulent flow over a flat surface. The flow was simulated using 10^6 grid points per MPI-task. At 1024 cores, the algorithm performed at approximately 90% parallel efficiency relative to the performance on 64 cores.

1.3 The MD algorithm: $\mathcal{S}treamMD$

The molecular dynamics algorithm solves Newton's equation of motion for a set of discrete molecules. The interaction potential in the simplest case is the Lennard-Jones potential, but the solver can also model long-chain molecules. Algorithmic acceleration is achieved via updated cell lists, which maintain a record of molecules within physical proximity. These cells are optimised in order to maximise the number of steps where interacting molecules are within the same or neighbouring cells, before the need to reconstruct the cell list. Further acceleration is achieved via neighbour lists which maintain a record of neighbouring molecules. Parallelism is implemented via MPI domain decomposition. The physical domain is divided among MPI processes, in a similar manner to the continuum solver. Neighbouring cells, and hence neighbouring molecules, are therefore hosted on the same MPI process. Halos, or duplicate copies of cells, at the boundaries are employed in order to maximise the number of computations prior to rebuilding of the cell and neighbour lists. It should be noted that the domain decomposition parallelism in $\mathcal{S}treamMD$ is similar to the approach adopted for $\mathcal{T}ransFlow$. This facilitates the task of interfacing the two applications since a logical connectivity can be readily established between boundary processes at the continuum-MD boundary.

The performance of $\mathcal{S}treamMD$ was compared to LAMMPS which was developed by Sandia National Laboratories and was optimised and extensively tested for parallel performance. The parallel scaling efficiency of $\mathcal{S}treamMD$ was shown to be similar to LAMMPS. In addition, similar to $\mathcal{T}ransFlow$, the scalability of $\mathcal{S}treamMD$ has been verified extensively with target operation at 10^3 CPU cores. Using 3.3×10^6 molecules, $\mathcal{S}treamMD$shows 94% parallel efficiency on 1024 cores relative to the 8 cores performance on HECToR.

1.4 Coupling CFD and MD

The prohibitive computational cost of MD simulations at the macroscale has motivated the development of hybrid techniques. In these methods, the MD simulation is limited to a small spatial region, for example interfaces. The rest of the flow domain is computed using a continuum approach. An overlap between the MD and continuum regions allows the exchange of the two states, and thus couples the two descriptions.

Independently, the MD and continuum descriptions are well established. The difficulty arises in the overlap region where they must be coupled. The coupling must satisfy conservation laws and ensure continuity of density, momentum, and energy and their fluxes. The upscaling problem (MD to continuum) is straightforward: microscopic properties are integrated locally over space and in time to provide boundary conditions for the continuum region. The inverse problem, however, is non-trivial and in fact non-unique. The continuum-molecular coupling adopted in this work is based on flux exchange, and was pioneered by Flekkøy and coworkers. It naturally ensures that fluxes from one configuration "flow" to the other, but special care must be taken in order to ensure continuity of the state variables.

The above coupling methodology has been tested in canonical problems, for example laminar Couette and Poiseuille flow were simulated over a molecular representation of flat surfaces and two-dimensional, organised roughness. A significant leap in modelling and computational ability is required in order to translate these canonical studies to simulations of engineering flows, where the outer flow is multiscale and surface textures and coatings are modelled more accurately using MD. Coupling of $\mathcal{T}ransFlow$and $\mathcal{S}treamMD$for massively parallel simulations on HECToR is intended to serve this purpose. It should be noted that the coupler module, however, is designed to be a general purpose library, or a multi-purpose computational coupling utility, which is available to the wider research and HECToR community.

2 Coupler design and implementation

This second generation of the coupler design has been re-engineered in order to incorporate a few desirable elements. Most importantly, the coupler is now designed along the philosophy of the Message Passing Interface (MPI) library: It is a collection of library calls that can be accessed from any application and can handle any data format. The main elements of the coupler are described below, along with the motivation and benefits of its features.

2.1 Client scope

It was also essential that the client applicants maintain their scope and data integrity. Therefore the coupler was designed to interface independent applications. This was in part possible via the design of the coupler which is discussed below, but also by making use of the Multiple Program Multiple Data capabilities of MPI. For example, the following call can run both `app1.exe` and `app2.exe` separately,

```
>> mpirun -n nproc1 ./app1.exe :  -n nproc2 ./app2.exe
```

The applications can therefore retain their independent instructions, data and scope for independent operation. However, the same command can be used to run the two applications conjointly, if they include calls to the coupler library. Any exchange of data is via the coupler interface.

Since the coupler is a set of library routines, any access to these routines is via functional calls within the client applications. In order to ensure minimal change to the client application, these calls can all be grouped in a "socket", or "portal" module within the clients. The terminology is intended to describe how the client is plugged into the coupler via the socket module.

2.2 Coupler modules

The coupler is composed of three main categories of modules which manage (i) the setup of the interface between client applications, (ii) data exchanges and (iii) access to parameters.

(i) The setup library calls are intended to establish the communicators with overlapping processors from the client applications. Therefore, these library calls must follow any internal calls within the client applications to initialise MPI and to setup a process topology. The setup routines then create the `CPL_INTER_COMM` which is the logical process topology which governs any data exchanges, or overlaps, between the client appellations. In addition, it initialises the coupling parameters which are prescribed by the user. It should be noted that the initialisation step involves many error checks and logical expressions in order to ensure that the topology of the interface between the client applications is accurate. This utility is therefore advantageous in coupling any set of client applications, since the extensive checks minimise user error.

(ii) The exchange class is a variant of the MPI, non-blocking send and receive operations, and other global operations (within the scope of the coupler communicators) such as scatter and gather. One important point to note is that the coupler data transfer operations are designed to handle any data structure, dimensionality and size being exchanged. Therefore, the coupler

retains generality and remains a general-purpose interfacing utility independent of the client applications and data structures.

(iii) The final category of library routines is a utility for retrieving parameters from the coupler. It can provide information on process topology, logical overlaps between topologies...etc. It is used throughout the setup process, but is also accessible to client applications that require direct knowledge of coupler entities.

3 Summary of work

During this second part of the project, a number of work packages were originally proposed and are summarised below. It should be noted that the new coupler design facilitated many of these tasks. An explanation of the new coupler design and how it addresses the deliverables is also included below.

Work Package 1.1: Load balancing of the coupled application, by appropriate division of compute resources between the various algorithms.
The new coupler design (MPMD) ensures that all the scalability characteristics of the client applications are retained. As discussed earlier, the scalability of $\mathcal{T}ransFlow$ has been demonstrated up to 10^3 cores on HECToR phase 3 with 90% parallel efficiency. In order to achieve this scalability, effort was dedicated to synchronising the communication pattern in $\mathcal{T}ransFlow$. In addition, effort was dedicated to improving the parallel performance of $\mathcal{S}treamMD$, and ensuring scalability figures commensurate with established packages such as LAMMPS. The parallelism was also extended to include the latest features of the code (originally only available in the serial release) such as long-chain molecules.

Work Package 1.2: Implementation of non-blocking MPI communication in the coupler and performance assessment.
As the case with any MPI-based development, the initial effort used blocking communication in order to ensure synchronous execution and for ease of debugging. However, in the final release of the coupler, all the MPI calls are based on non-blocking communication. Extensive verification of the coupler module ensured accuracy of the non-blocking calls. Further optimisation of this feature is dependent on the client application: Analysis of the client applications is required to identify suitable tasks that can be performed during asynchronous data exchanges.

Work Package 2.1: Implementation of mixed-mode operation (OpenMP + MPI) for massively parallel computations
The coupler library is an MPI-type utility. Therefore, the use of mixed-mode operation was intended for the client applications. $\mathcal{T}ransFlow$ has been adapted to fully exploit mixed mode parallelism with MPI as the main, or default, mode. Thread-safe, shared-memory parallelism was extensively tested and validated agains MPI-only benchmarks. No major change in performance was observed since our simulations are compute intensive, without significant communication bottlenecks. It is expected, however, that performance benefits would be observed in massively-parallel computations on future architectures with large number of cores operating in shared-memory mode.

Work Package 2.2: Implementation of "MD farming", where MD instances are generated at different locations in the continuum.
The current design of the coupler was specifically engineered to be a multi-purpose, or general-purpose, coupling utility. This choice is a major benefit to the research and HECToR community. The case of farming is a straightforward example. Since the coupler operates in Multiple Program Multiple Data mode, any external programs can interface via the coupler. In that respect, any number of MD instances can interface to the continuum solver. At this stage, a farming example has not been verified since priority was dedicated to re-engineering the coupler into a general multi-purpose utility, performing extensive performance and accuracy check and implementing error reporting and documentation. In the near term, effort will be dedicated to exploiting the coupler for massively parallel simulations of physical problems of interest in order

to maximise the scientific impact from this software development.

4 Impact

The software development performed in this project has impact in two main categories: (i) application areas and (ii) future computing requirement. In application areas, a general coupling library is essential for interfacing specialised application softwares from various strands of research, in order to simulate multi-physics problems. With the ever increasing necessity to simulate complex physics using large-scale scientific computing, it is impractical to redevelop or merge algorithms which simulate various physical phenomena. Such approach is inefficient, and requires exhaustive validation and verification of the new codes. Adopting the coupler as a basis for interfacing existing and established algorithms is a much more robust and efficient approach. Some example applications where the current coupler is expected to be applied are simulations of electro-kinetic flows in microchannels and simulations of drag reduction using surface coatings. The same coupling library can be applied by other researchers, e.g. in coupling Reynolds Averaged Navier-Stokes solvers with higher fidelity methods such as direct numerical simulations, or performing coupled radiation transport and flow simulations in reactor physics. Many of these coupled applications are actively pursued at Imperial College and widely by the UK research community, e..g. members of the UK Turbulence Consortium. In addition, the coupler will create many new opportunities for the scientific community across a broad spectrum of disciplines.

It is important to consider the impact of the coupling library on the HECToR AU footprint. Using the coupler reduces this footprint in two ways: First, the coupler accelerates the solution procedure of multi-physics problems by concurrently running the various software, relative to iterative sequential coupling of applications which is at times adopted in multi-physics problems. Second, interfacing established applications using the coupler library circumvents the need to re-validate and load-balance multi-physics simulations that require more than one application software. This advantage reduced the number of man-months dedicated to this effort, and the HECToR AU requirements for such validation and scalability studies.

5 Conclusions and future work

Coupling of various massively-parallel computational algorithms is a challenging task, but a necessity in multi-physics simulations. The coupler developed during this project is a powerful resource. It has the format of a library, which can be invoked by various applications, which are executed independently. It is compartmentalised such that it does not affect the execution of any of the applications, and only manages data exchange between codes. In that respect, it follows the model of the MPI library and interfaces applications, as opposed to processes within one application.

The use of the coupling library does not require any modification of the client applications, except for a class of "socket" routines that manage the communication via the coupler library. This approach is least intrusive for interfacing independent, massively parallel applications. It also facilitates the developer's task since it is independent of the individual applications data structures and management, parallel implementation and scope.

The coupler library has been applied in simulations of coupled continuum-molecular flows. The continuum algorithm is $\mathcal{T}rans\mathcal{F}low$, and the molecular dynamics were performed using $\mathcal{S}treamMD$. Both codes have been extensively validated and optimised for parallel scalability. Both Couette and Poiseuille flows were simulated using continuum-MD coupling for validation. Further simulations will be performed during the coming year of various physical phenomena that require MD-level detail near interfaces, coupled to large-scale continuum solution away from the interface. Some examples include the effect of outer macroscopic shear on polymeric surface coatings and the onset of cavitation in microscopic textures.

Finally, the coupler library will be hosted online for the benefit of the wider research community, in order to facilitate multi-physics simulations. In addition, the work will be disseminated at various conferences and consortia in an effort to maximise the uptake of the library.

Acknowledgements

This project was funded under the HECToR Distributed Computational Science and Engineering (CSE) Service operated by NAG Ltd. HECToR – A Research Councils UK High End Computing Service – is the UK's national supercomputing service, managed by EPSRC on behalf of the participating Research Councils. Its mission is to support capability science and engineering in UK academia. The HECToR supercomputers are managed by UoE HPCx Ltd and the CSE Support Service is provided by NAG Ltd. `http://www.hector.ac.uk`.

References

[1] M. K. Borg, G. B. Macpherson, and J. M. Reese. Controllers for imposing continuum-to-molecular boundary conditions in arbitrary fluid flow geometries. *Molec. Sims.*, 36:745, 2010.

[2] R. Delgado-Buscalioni and P.V. Coveney. Continuum-particle hybrid coupling for mass, momentum, and energy transfers in unsteady fluid flow. *Phys. Rev. E*, 67:046704, 2003.

[3] E. G. Flekkøy, G. Wagner, and J. Feder. Hybrid model for combined particle and continuum dynamics. *Europhys. Lett.*, 52:271, 2000.

[4] N. G. Hadjiconstantinou. Hybrid atomistic-continuum formulations and the moving contact-line problem. *J. Comp. Phys.*, 154:245, 1999.

[5] D. M. Heyes. Shear thinning and thickening of the Lennard-Jones liquid. a molecular dynamics study. *Journal of the Chemical Society*, Faraday Transactions 2: Molecular and Chemical Physics 82, 13651383.

[6] D. M. Heyes. Transport coefficients of Lennard-Jones fluids: A molecular-dynamics and effective- hard-sphere treatment. *Phys. Rev. B* 37 (10), 56775696.

[7] J. Lee, S. Y. Jung, H. J. Sung and T. .A. Zaki Effect of wall heating on turbulent boundary layers with temperature-dependent viscosity. *Journal of Fluid Mechanics* 726, 196-225.

[8] J. Li, D. Liao, and S. Yip. Coupling continuum to molecular-dynamics simulation: Reflecting particle method and the field estimator. *Phys. Rev. E*, 57:7259, 1997.

[9] X. B. Nie, S.Y. Chen, W. N. E, and M.O. Robbins. A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow. *J. of Fluid Mech.*, 500:55, 2004.

[10] S. T. O'Connell and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flow. *Phys. Rev.*, E 52:R5792, 1995.

[11] W. Ren. Analytical and numerical study of coupled atomistic-continuum methods for fluids. *J. of Comp. Phys.*, 227:1353, 2007.

[12] T. Werder, J. H. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flows. *J. of Comp. Phys.*, 205:373, 2005.

[13] T. .A. Zaki and P. A. Durbin Mode interaction and the bypass route to transition. *Journal of Fluid Mechanics* 531, 85111.

[14] T. .A. Zaki and P. A. Durbin Continuous mode transition and the effects of pressure gradient. *Journal of Fluid Mechanics* 563, 357388.

[15] T. .A. Zaki, J. Wissink, P. A. Durbin and W. Rodi DNS of transition to turbulence in a linear compressor cascade. *In High Performance Computing in Science and Engineering*. Springer.