Unlocking First Principle Simulation of Liquid Injection using Blue Waters' Heterogeneous XK Nodes

1 Project Information

- Name and institution of principal investigator:

Asst. Prof. Vincent LE CHENADEC
Department of Aerospace Engineering
University of Illinois at Urbana-Champaign

- Corresponding author name and contact information (email)

Asst. Prof. Vincent LE CHENADEC
319F Talbot Laboratory
104 S. Wright Street
Urbana, IL 61801
+1 (217) 300-0347
vlc@illinois.edu

2 Executive Summary

Over the past decade, Direct Numerical Simulations of interfacial flows in the intermediate Reynolds and Weber regime have been performed with ever increasing fidelity levels. In particular, one-fluid interface-capturing techniques have played a major role in exploring and modeling complex and turbulent two-phase flows. Enable large scale applications however requires a number of advances in improving the efficiency and the load-balancing of existing computational infrastructures. The objective of the proposed work is to achieve this goal by leveraging the Blue Waters facility, to ultimately enable the development of new models, and ultimately the optimization and control of technological applications.

3 Description of Research Activities and Results

The reliable prediction of primary atomization and the resultant spray population statistics is vital for the design of low-emissions and stable gas turbine combustors. Atomization of a liquid jet is controlled by unique small-scale physics that rely on the interaction between vorticity generation at the small scales and the interface motions controlled by surface tension forces. Although a critical sub-component of a liquid fueled combustor, the liquid spray breakup resulting from complex fuel injectors is poorly understood. Yet it has a significant influence on the engine performance in terms of emissions, fuel consumption, thermo-acoustic instabilities and durability [4]. The modeling of liquid fuel injectors represents a formidable modeling challenge. The main quantity of interest is the spray penetration, characterized by the droplet number density function that includes the distribution of droplet sizes and velocities. This information is subsequently used to design the primary reaction zone. For instance, insufficient atomization could result in fuel-rich pockets that promote soot formation. Similarly, the nature of atomization will also control ignition tendencies in high-altitude relight conditions. Hence, reliable computational models that predict the final spray properties are indispensable for the successful design of aircraft combustors, and progress is required in two areas.

The first is Direct Numerical Simulations. In DNS, the equations that govern the mixing of the liquid and gas are solved *model free*, without simplifying assumptions. Hence they represent the highest achievable level of fidelity (*first principle*). This comes at a price, however, in that the computational techniques currently available face two limitations. First, liquid fuel injection involves complex interfaces topologies, characterized by discontinuous thermo-dynamic, kinematic, and mechanical properties. The discretization (*i.e.* the approximation of the original infinite dimensional partial differential equation with a finite dimensional algebraic system) involves a different technology altogether from the traditional *Taylor expansion-based* Computational Fluid Dynamics. Second, in the regime of interest (large yet finite Reynolds and Weber number), the governing equations (two-phase Navier-Stokes) are *chaotic*, and feature a wide range of scales, which requires the use of extremely large computational meshes. For the foreseeable future, it is therefore undeniable that DNS of full scale industrial configurations will remain out of reach. As a consequence, their major contribution will consist in the simulation of canonical flows, that will provide the quantitative information required to develop and valide the future generations of models.

The second area is Reynolds-Averaged Navier-Stokes and Large Eddy Simulation, characterized by their reliance on the modeling of the small scales of the flow. They, of course, come with a significant reduction in computational cost, the trade-off being their lower fidelity. An additional (often overlooked) advantage is their improved *controllability* over the local instantaneous formulation used in DNS, which is chaotic and therefore harder to optimize or control. Multi-phase modeling, however, suffers from an unfortunate difference with its single-phase counterpart: in single-phase turbulence, simple ideas such as the mixing length hypothesis already give useful results. Simple approaches to multi-phase modeling, on the other hand, have had a much more limited success. This is due to the considerable difficulties that plague the modeling of the many terms that arise from the averaging of the first principle equations. DNS, which can provide *a priori* insight into these terms, therefore complements modeling in a very clear way.

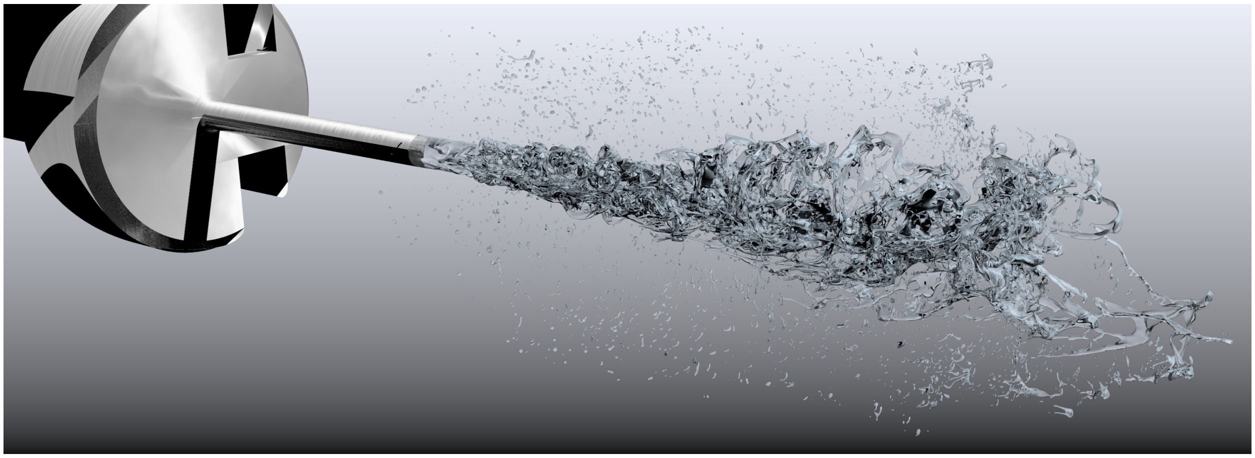

Figure 1: DNS of a scaled-up injector at high-density ratio (800), low Reynolds number (5 000) and moderate Weber number (60 000) conditions.

Even in canonical configurations, such as a simplex atomizer (Fig. 1), the unsteadiness, the multi-scale and multi-regime nature of these flows, require extremely large datasets to achieve statistical convergence. This can only be achieved with highly optimized file systems, and the high storage bandwidth found in Blue Waters. Additionally, numerical algorithms for DNS have been the subject of much research efforts, and among the techniques that have been developed, interface capturing methods hold most promises [7]. Recent developments, in the context of

Volume-of-Fluid methods in particular [3, 2], have significantly improved their predictivity. The scaling of these methods to the large scale application that are necessary to capture the physics relevant to most technologies however faces a major challenges, namely a far greater computational cost over alternative strategies. Acceleration, in particular by leveraging the heterogeneous `XK` nodes on Blue Waters, hold the key to unlocking the physics of primary atomization.

4 Accomplishments

This first year accessing Blue Waters has seen a complete rewrite of our *in-house* code, a `Fortran 95` structured finite volume code. The original code was parallelized using blocking `MPI` communication only, and one of the priorities was therefore to leverage computation and communication overlap. Additionally, performance analysis of the original code on a representative configuration revealed that the main share of the computational time was spent in the computational geometry (43%), and the linear algebra (iterative solvers, 41%) modules. These represent very different workload, as explained below.

The linear algebra component is currently delegated to the `PETSc` library, which interfaces our application to a number of solvers and preconditioners in a very flexible manner, hence enabling the testing of different combination to find an optimum with minimal efforts on our end. Since the degrees of freedom are uniformly partitioned across processes, the workload is also uniformly distributed. The workload attributed to the computational geometry component, on the other hand, is proportional to the interface surface area present on each process which, as seen in Fig. 1, is extremely heterogeneous. The focus was therefore set on improving the performances and the load-balancing of this module.

4.1 Improving the Performances at the Computational Fluid Dynamics Level

Computational geometry algorithms, as opposed simpler algebraic ones, is required by the reliance of the equations governing the evolution of interfacial flows *distributions*, as opposed to *smooth functions*. Incorporating this distinctive feature is the backbone of sharp interface methods. An example of such method is the *Level Set* method [5], which leverages an auxiliary function, typically chosen to be the signed distance to the interface, denoted hereafter ϕ. The characteristic function χ is then simply defined as $H \circ \phi$, where H denotes the Heaviside distribution. An alternative method, that has gained renewed interest over the past few years, is the geometric *Volume-of-Fluid* method, which has proved extremely valuable in overcoming the limitations of Level Set (*e.g.* lack of local conservation properties) and Front Tracking (*e.g.* arbitrary topology changes) methods. Interestingly, the benefits are not restricted to the interface representation, but they also extend to discretization of the flow governing equations. Consistent interface and momentum transport in particular has proved effective in simulating realistic injectors at ambient conditions.

Geometric Volume-of-Fluid methods are most easily interpreted as a *Reconstruct-Evolve-Average* method (REA), typical of Finite Volume update schemes. In the *Reconstruct* step, the *Piecewise Linear Interface Calculation* algorithm leverages the cell-centered volume fraction field to construct a local approximation of the interface (a *planar polygon*). In the *Evolve* step, the faces adjacent to each control volume are transported along the flow characteristics (a single contact wave in the incompressible case), leading to a signed (and potentially self-intersecting) *polyhedron* that is intersected against one side (say, the liquid) of the reconstructed piecewise planar interface. Finally, in the *Average* step, the resulting fluxes are summed to update the (liquid) volume fraction values.

In effect, the geometric Volume-of-Fluid method therefore tailor the bases used to approximate the characteristic distribution χ, as opposed to relying on the traditional use of polynomials which perform poorly near discontinuities (Gibbs phenomenon). Although this tailoring is extremely beneficial to prevent numerical artifacts and/or numerical dissipation, its implementation introduces a profusion of conditional statements (*e.g.* `if`), which makes the computational geometry algorithms very expensive. This module's work is split in two stages: the first (27%) involves the solution of a non-linear constrained least-squares problem for every interfacial cell (Piecewise Linear Interface

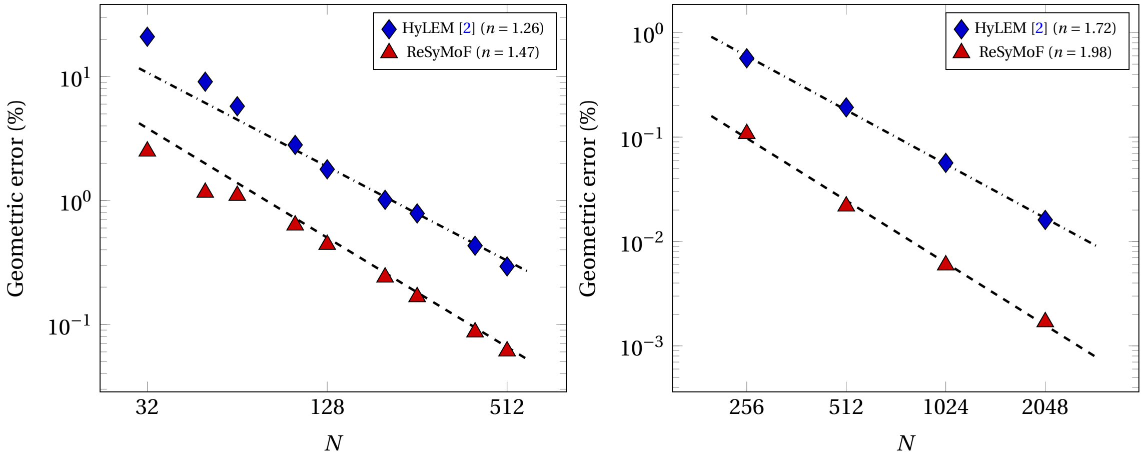

Figure 2: Improved accuracy of newly developed reconstruction algorithm (ReSyMoF) in two standard benchmarks: Zalesak's disk (*left*), and 2D deformation (*right*).

Calculation). The minimization algorithm is a standard algorithm (Levenberg-Marquardt). Being derivative-free, the cost of each minimization was concentrated on the function evaluations, called for each cell involved in the stencil. The original PLIC algorithm we employed (LVIRA [6]) involved a 3^3 stencil; an alternative method, the Moment-of-Fluid [1] method, is *local* in the sense that it does not require access to any neighboring information. We found the new algorithm to be not only faster (3 times speed-up), but also superior in accuracy (Fig. 2).

The second stage (73%) is associated with the interface *transport*, which can be formulated in both *Flux-Based* and *Semi-Lagrangian* forms. This component involves a large number of recursive calls to a *clipping-and-capping* algorithm. Each operation consists forming a new three-dimensional polyhedron by truncating an existing one with a half-space. In order to assess the potential for speed-up, we explored two different implementation of our in-house computational library: the first was based on a general representation of complex polyhedra using collision tables, the second was based on the use of recursive tessalations, performed by means of a low dimensional look-up table. The interesting conclusion was both implementation resulted in very similar performances, hence we proceeded to the next step.

4.2 Improving the Performances by Uniformly Balancing the Load

Regardless of the underlying mesh connectivity (structured or unstructured), the balancing of the workload found in two-phase flow solvers is organized into

- a uniformly balanced component, that corresponds to all linear algebra related tasks, where the communication load is determined by the density of the sub-domain boundaries (update of the *guard* or *ghost* cells),
- a poorly balanced component, that corresponds to the reconstruction and transport tasks detailed above.

Because of these different workload distributions, adaptive strategies (*e.g.* migrating the `MPI` processes across physical nodes until a optimal, yet non-uniform, balance is achieved) will not improve the performances significantly. Although both components (reconstruction and transport) represent complex operations, they are completely defined by a small number of input and output parameters. For example, the input for the interfacial reconstruction requires the cell size (3 double precision numbers) and the lowest moments of the geometric distribution a reference phase (volume and center of mass coordinates, that is 4 numbers). The output simply consists of the equation of a plane, that requires two angles and one intercept, hence 3 numbers. As a consequence, it is reasonable to attempt to distribute the total N reconstruction (resp., transport) tasks across the P processes uniformly, that is allocate to process $p \in [\![0, P-1]\!]$

$$N_p = \left\lfloor \frac{N(p+1)}{P} \right\rfloor - \left\lfloor \frac{Np}{P} \right\rfloor \tag{1}$$

tasks [8].

For each module (reconstruction and transport), the load-balancing is therefore performed as follows:

1. the workload is determined (this involves one `MPI_Iallgather` call),
2. the uniform workload is defined according to Eq. 1,
3. the migration of the input is performed (this involves one `MPI_Ialltoallv`,
4. the tasks are solved,
5. the output are collected (this involves one final `MPI_Ialltoallv` call).

Interweaving the computation and the non-blocking communication for both modules was also beneficial. Small scale tests (*i.e.*, 1024^3 degrees-of-freedom) have shown up to ten-fold speed-ups. The avenue we are currently exploring (see Sec. 6) is to leverage the NVIDIA GK110 "Kepler" accelerator to speed up the individual tasks. This involves a rewrite of our computational geometry library using `OpenCL`.

5 List of Publications Associated with this Work

The work delineated below has directly enabled the following contributions

1. "Realizable and Symmetric Transport of Geometric Moments and Application to the Moment-of-Fluid Method", submitted to the *Journal of Computational Physics*
2. "A Survey of Momentum-Conserving Methods for High-Reynolds Incompressible Two-Phase Flows", submitted to ILASS-Americas proceedings
3. "Load balancing and Acceleration for Volume-of-Fluid Methods", in preparation

6 Plan for Next Year

2015 mainly consisted in deploying our in-house code on Blue Waters. This involved a large number of developments, which will continue during the first two quarters of 2016 (Q1 and Q2) on

- a new in-house `OpenCL` computational geometry library,
- the optimization of the IO currently performed with the `HDF5` library,
- the use of `OpenACC` to speed-up the pre-conditioned Krylov subspace solvers currently in use in the code.

These first two quarters will also see the first preliminary runs (on a canonical configuratin, Taylor's pressure swirl atomizer). As seen from Fig. 3), the conical liquid sheet leads to highly inhomogeneous interface distribution, which will therefore put our implementation to the test. The second half of 2016 (Q3 and Q4) will lead to our first large scale computations in a configuration of interest to scram-jet technology, namely a liquid jet in cross-flow.

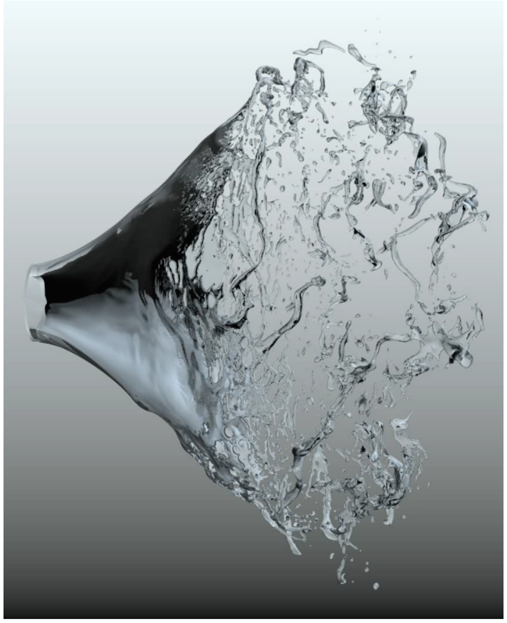

Figure 3: A pressure-swirl atomizer.

A concurrent project, led in collaboration with Asst. Prof. Taraneh Sayadi (UIUC), is also planned to use the Blue Waters platform end of 2016. The same code architecture, but a set of Computational Fluid Dynamics algorithms, will be used. The project, entitled "Adjoint-based Optimization and Control of Separated Interface Flows". The transition from model-based numerical simulations to model-based design and optimal control requires additional technology that enables access to *inverse* information. To date, this information has only been extracted from simulations of simplified configurations with additional unrealistic assumptions. In related fields (aero-dynamics, aero-acoustics), inverse optimization and control have improved airfoil shapes and reduced noise levels. Multiphase flows however constitute a far larger step in complexity, and require advanced techniques such as *adjoint-based optimization* we are currently extending to two-phase flows.

The 2016 timeline for all corresponding sub-projects can be found in Tab. 1. We request an allocation of 245,000 node-hours, and the usage is scheduled as follows: Q1 — 5%, Q2 — 15%, Q3 — 45% and Q4 — 35%. The number of nodes will not exceed 500, and the memory usage 10 GB per node.

		Q1	**Q2**	**Q3**	**Q4**
Computational infrastructure	Computational geometry on GPU using OpenCL	■	■		
	Linear algebra acceleration using OpenACC		■	■	
	I/O optimization		■		
First principle simulations	Pressure-swirl atomizer		■	■	
	Liquid jet in cross-flow			■	■
Adjoint-based minimization	Motion planning			■	
	Linear stability analysis				■

Table 1: Timeline for 2016.

References

[1] V. Dyadechko and M. J. Shashkov. “Reconstruction of multi-material interfaces from moment data”. In: *Journal of Computational Physics* 227.11 (2008), pp. 5361–5384.

[2] V. Le Chenadec and H. Pitsch. “A 3D Unsplit Forward/Backward Volume-of-Fluid Approach and Coupling to the Level Set Method”. In: *Journal of Computational Physics* 0 (2013), pp. 10–33.

[3] V. Le Chenadec and H. Pitsch. “A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows”. In: *Journal of Computational Physics* 249.0 (2013), pp. 185–203.

[4] A. H. Lefebvre and D. R. Ballal. *Gas Turbine Combustion: Alternative Fuels and Emissions, Third Edition*. CRC Press, 2010, p. 557.

[5] S. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces*. New York: Springer, 2003.

[6] J. E. Pilliod and E. G. Puckett. “Second-order accurate volume-of-fluid algorithms for tracking material interfaces”. In: *Journal of Computational Physics* 199.2 (2004), pp. 465–502.

[7] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct numerical simulations of gas-liquid multiphase flows*. Cambridge; New York: Cambridge University Press, 2011.

[8] T. Tu, D.R. O’Hallaron, and O. Ghattas. “Scalable Parallel Octree Meshing for TeraScale Applications”. In: *ACM/IEEE SC 2005 Conference (SC’05)*. IEEE, 2005, pp. 4–4.