

Discrete velocity methods for helicopter multi-block CFD solver: towards real-time wake simulations

PI: G.Barakos

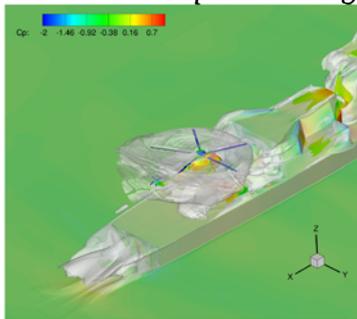
Technical staff member: M.Woodgate

Project summary

This project is a step towards the development of real-time simulations using detailed flow physics.

Such simulations are necessary to enhance the physics included in the models used for flight simulators. An example is the simulation of a helicopter operation near a building, ship, or wind turbine where due to wind, the resulting wakes are affecting the performance and the operation of the helicopter. The figure shows an example of this situation.

IMAGE Helicopter landing in the wake of ship.



At present, wakes are treated like databases that are pre-computed and stored. This is unsatisfactory since there are situations where the dynamics of the wake will change with the weather conditions or the presence of the helicopter. It is therefore advisable to look at how the wake can be computed in real-time using a small cluster of computers attached and interfaced with the flight simulator. Most of the times the fluid flow is simulated using mesh-based solvers of the Navier-Stokes equations like the HMB3 method of Glasgow. These methods are very robust and accurate, but cannot be easily deliver real-time results. For this reason, this project takes advantage of the Lattice-Boltzmann method that is a special way of solving Boltzmann's equation on a lattice of points. The method is restricted to incompressible flows on this occasion, which is fine due to the nature of the wakes that are to be resolved. In addition, the method is essentially using Cartesian grids. Since we are not using the method for computing detailed aerodynamic forces on the helicopter, this limitation is not important for this project.

The LBM or DVM was improved in this project to allow for real-time computations of wakes for cases where the wind speed is of the order of 10 m/s. This is adequate for most cases and a first step in this direction. An additional step that had to be taken was related to the coupling between the flow methods that can be used to solve the flow around a helicopter and the LMB/DVM

method. This was resolved by using a control software that used the MPI library and controlled the start and end of each of the solvers.

The most challenging part of the work was, however, the load-balancing between the HMB3 and the LMB/DVM methods. This was a difficult step due to the need to maintain real-time execution of the LMB method while keeping at least a loosely coupled simulation in place with the main solver HMB3. The two pieces of software have different speedup characteristics when used in a parallel computer, and different I/O requirements. Working with EPCC we were able to establish a strategy for re-balancing the tools with minimal overhead and with good potential for exploitation in a flight simulator.

Achievement of objectives

The original proposal had the following objectives:

- 1) Port the HMB2/DVM method on ARCHER and measure the performance of the coupled HMB2/DVM method as it stands.
- 2) Improve the performance of the HMB2/DVM method by implementing implicit coupling between the two codes HMB2 and DVM instead of the currently-employed explicit and symplectic schemes.
- 3) Improve the load balancing of the HMB2/DVM method by implementing a simple strategy for re-balancing the work load between processors as the solution proceeds in time.

Demonstrate the benefits of the method for the problems of ship and wind turbine wake predictions

The objectives 1-3 were fully met while the last objective was only met partially since the wind turbine wake demonstrations did not materialise due to lack of time. Early in the project it was established that the performance of HMB2 was adequate but the DVM needed further work. By the end of the project our DVM was as scalable as the most established DVM method in the open literature.

The coupling between the two methods was demonstrated for the case of a floating wind turbine where the DVM was not used since scaling efforts were still taking place. The coupling took place via a control programme that spawns the two solvers. The third objective involved computations with the SFS2 (simple frigate shape 2) body that were proved successful and DVM could maintain real-time executions without I/O. Given that this is the first time real-time computations are reported with DVM for these wakes, it is justified to state that the objectives were met at least by 90%.

Regarding the performance of the DVM, at the beginning of the project we could only perform 2.67 Million Updates per second on a 100x100x100 lattice, while at the end of the project this was increased between 15.6 Million updates per second for a 96x96x96 lattice to a maximum of 19.93 Million updates a second for a smaller 16x16x16 lattice . Figure 2 above shows the obtained speedup on ARCHER. Apart from the DVM, the coupling strategy between the two solvers was streamlined and improved and a method for dynamic load balancing is now in place. This last method was not available at the beginning of the project and represents new capability.

Summary of the software

At present the tools are installed on ARCHER but we did not try to create a software repository entry. This can of course be addressed, now that the tools are in almost final form.

