11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

DEVELOPING A SOFTWARE FRAMEWORK FOR FLUID-STRUCTRE INTERACTION INCOMPRESSIBLE FLOW PROBLEMS

Marcos Vanella[1], Elias Balaras[1] and Anshu Dubey[2]

[1] The George Washington University, 801 22ND Street, NW Washington, DC 20052, USA.
[2]Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA.

Key words: *Fluid-Structure Interaction Problems, Incompressible Flows, Petascale Applications.*

High fidelity computations of fluid-structure interactions (FSI) are amongst the most challenging problems in computational mechanics requiring large computational resources. Because they include moving and deforming boundaries, many of these applications have different resolution requirements in different parts of the domain and therefore requirement adaptive mesh refinement (AMR) for computational efficiency. Vanella & Balaras [4] proposed a robust and efficient scheme applicable to multi-body fluid-structure interactions. It utilizes an Eulerian mesh to model the fluid and uses Lagrangian markers to couple the structure with the mesh by means of an immersed boundary method. Here we describe the adaptation of this scheme to a parallel AMR environment in FLASH [1], a massively parallel multiphysics code, which was originally developed for Astrophysics applications. FLASH has a Lagrangian framework that co-exists with the Eulerian mesh [3, 2], thereby providing a framework on which the FSI capabilities with AMR could be built with minimal investment in new infrastructure development.

FLASH discretizes the physical domain into an Eulerian mesh consisting of *cells*, which are grouped to form blocks. A union of blocks spans the entire computational domain. Each block, surrounded by appropriate layers of ghost cells, presents a self-contained computational domain to the explicit Eulerian solvers in the code. FLASH also has a built-in Lagrangian framework that uses a simplified two dimensional array as its primary data structure. The first dimension encodes the attributes of the Lagrangian entities and the second dimension is the list of entities [2]. The Lagrangian entities have an associated block number from the Eulerian mesh with which they physically overlap. For the vast majority of applications, the Lagrangian entities live on the same processor as their associated block for efficient mapping of quantities to and from the mesh. The framework provides implementations for the mappings, and for migrating data to appropriate blocks when the corresponding Lagrangian entities physically move between blocks.

To accommodate the implementation of the immersed boundary FSI formulation within the existing framework we have introduced a series of conceptually simple, therefore elegant, algorithmic modifications to the way Lagrangian particles are handled. More remarkably, we have succeeded in devising the method atop an already mature application code that was originally designed for vastly different application domains. With a few well placed augmentations to the existing frameworks, and design of a couple of new parallel algorithms, we were able to achieve our goal of having FSI capabilities with AMR without having to build all the capabilities from scratch. After a few iterations in algorithm design, we finally have an implemented method which has demonstrated good scaling, both in number of solid objects and the size of the computing platform. There is further scope for improving performance by optimizing the master processor selection algorithm. It is expensive because global `AllReduce` operations are used to check overlap of all processors with the object in order to select the master processor. Future work would involve removing these `AllReduce` operations by identifying nearest neighbors to send data to. A markers position can advance at most from one block to an adjacent one within one timestep. Therefore, we can simplify the routine by checking the overlap of only the nearest processors, thereby reducing cost.

The work was supported by the NSF (awards OCI-0904920, and CBET-0932613), and used computational resources from XSEDE (award TG-CTS110028).

REFERENCES

[1] A. Dubey, K. Antypas, M.K. Ganapathy, L.B. Reid, K. Riley, D. Sheeler, A. Siegel, and K. Weide. Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code. *Parallel Computing*, 35(10-11):512–522, 2009.

[2] A. Dubey, C. Daley, J. ZuHone, P. M. Ricker, K. Weide, and C. Graziani. Imposing a Lagrangian particle framework on an Eulerian hydrodynamics infrastructure in FLASH. *ApJ Supplement*, 201:27, Aug. 2012.

[3] A. Dubey, K. Antypas, and C. Daley. Parallel algorithms for moving lagrangian data on block structured Eulerian meshes. *Parallel Computing*, 37(2):101 – 113, 2011.

[4] Marcos Vanella and Elias Balaras. Short note: A moving-least-squares reconstruction for embedded-boundary formulations. *J. Comput. Phys.*, 228(18):6617–6628, October 2009.