Exascale Computing of Multiphase Flow

BERKELEY LAB

Erik Palmer[1], Michele Rosso[2], Andrew Myers[2], Andy Nonaka[2], Jordan Musser[3] and Ann S. Almgren[2]
[1]University of South Carolina, [2]Lawrence Berkeley National Laboratory, [3]National Energy Technology Laboratory

Introduction

MFiX ECP EXASCALE COMPUTING PROJECT

MFiX is a software suite developed by the Department of Energy to model dense or dilute fluid-solid and fluid-fluid flows, such as those that occur in energy conversion and chemical processing reactors. An example application of this software can be seen in the cyclone in figure 1 below. MFiX or **M**ulitphase **F**low with **I**nterphase e**X**changes fulfills a vital need as it is difficult to predict the behavior of full scale reactors from expensive small scale prototypes. The code itself consists of about 135,000 lines of Fortran 90. In 2016, the MFiX code won seed funding as part of the Exascale Computing Project to be upgraded to run on the next generation of supercomputers. [1]

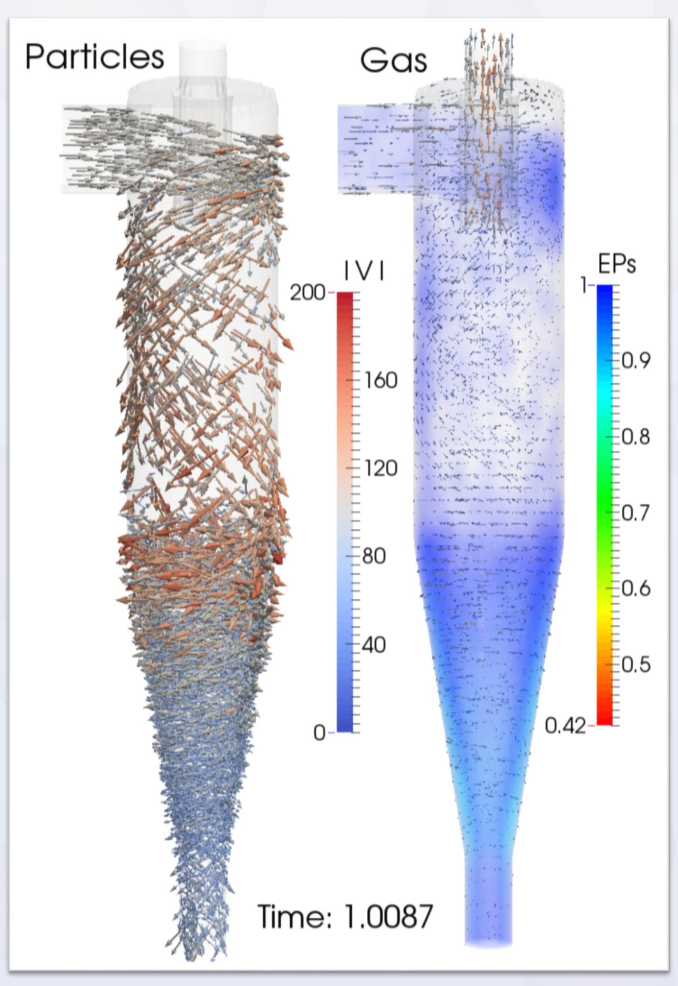


Figure 1. Simulation of a cyclone, showing both particle and gas phases. Image taken from MFiX Applications at the NETL Multiphase Flow Science website: mfix.netl.doe.gov.

Particle Modelling

MFiX-DEM couples particle and fluid motion using the discrete element method (DEM) to model particles. Within DEM there are a number of techniques for simulating particle collisions, including both hard and soft-sphere. In addition, within the soft-sphere approach, there are numerous models of varying complexity. Of these, even the simplest models need to account for a variety of forces involved in particle collisions. For example, in addition to an inelastic collision force in the normal direction, there are also effects in the tangent directions and on the angular momentum which is subject to the frictional forces of each particle. Equations 1 and 2 below, show the simplest form of the normal force with the desired properties and a more complex form. For simulations where the volume fraction of particles is high when compared to the fluid, multibody collisions should also be considered common, see figure 2.

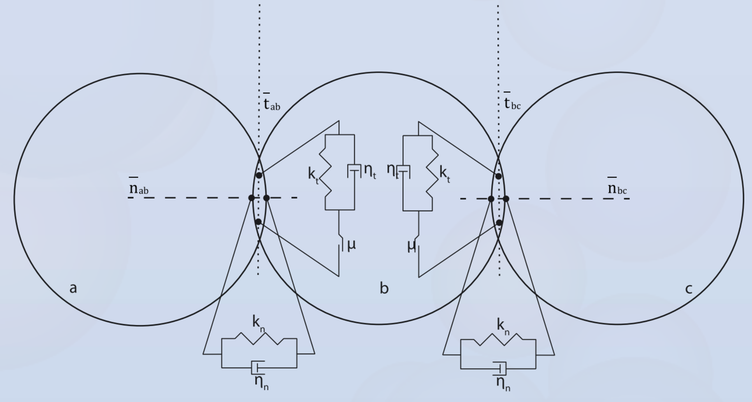
$$F_n = -k_n \xi - \eta_n \dot{\xi} \qquad (1)$$

$$F_n = kF_{nd} + F_{Hertz} \qquad (2.1)$$

$$F_{Hertz} = \frac{4}{3}\frac{E}{1-\nu^2}\sqrt{R_e}\,\delta_n^{3/2} \qquad (2.2)$$

$$F_{nd} = 2\frac{(1-2\nu)(1+\nu)}{E}\left(2\eta_2 + \frac{\eta_1}{3}\right)\frac{E}{1-\nu^2}\sqrt{R_e\delta_n}\,\dot{\delta}_n \qquad (2.3)$$

$$k = \left(0.8 + 26\nu^3\right)\sqrt{\frac{\eta_1}{\eta_2}}\left(\frac{\delta_n}{R}\right)^{0.04} \qquad (2.4)$$



Equations 1 and 2. (Top) Describe the collision force of particles in the normal direction. The linear spring dashpot model (eq.1) is considerably simpler than the equations for the second model (eq. 2) from Zheng et al. [2] Figure 2. (Bottom) Diagram of a multibody particle collision where both normal and tangential forces are modelled as a linear spring dashpot combination. [3]

The Exascale Challenge

Although computing particle collisions can become quite complex, it is not the limiting factor in large particle-fluid simulations. The most time consuming step is checking for collisions. This is because collision forces are orders of magnitude more sensitive to particle positions than the movement of the particles in the fluid themselves and therefore require very small time steps to resolve. In addition, checking all possible particle collisions at these small time steps results in a N^2 computation. These two effects snowball and make large particle simulations extremely demanding.

Adaptive Mesh Refinement

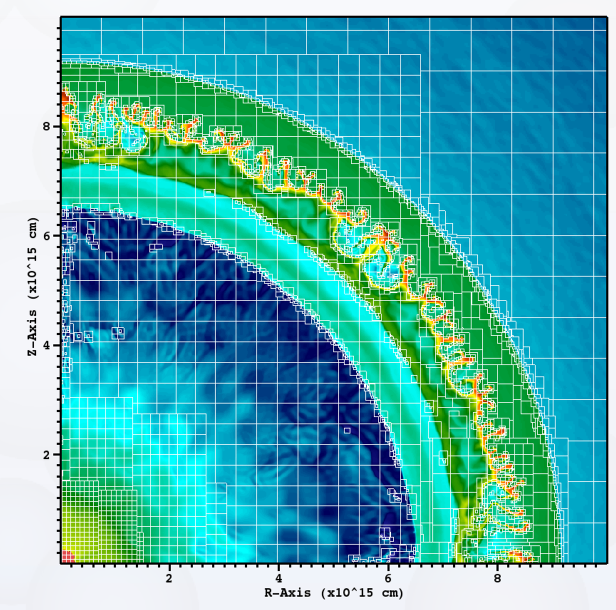
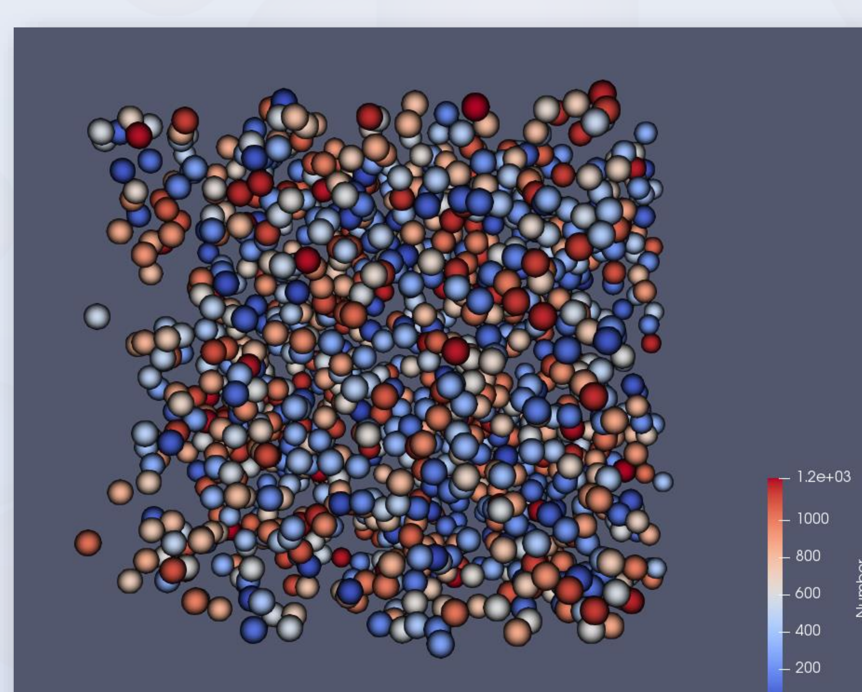
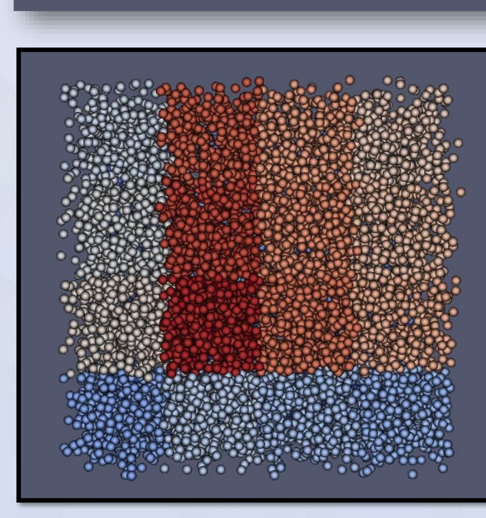


Figure 3. A simulation performed with the Castro Adaptive Mesh Refinement code that uses the AMReX framework. The larger boxes represent areas of lower accuracy and lower computational cost. Smaller boxes indicate a concentration of resources to resolve areas of greater interest. Image taken from "Berkeley Lab to Lead AMR Co-Design Center for DOE's Exascale Computing Project", Nov. 11, 2016, at web: crd.lbl.gov.

The Lawrence Berkeley Lab's Center for Computational Science and Engineering (CCSE) is home to the AMReX adaptive mesh refinement (AMR) code framework. Using this freely available framework researchers can automatically allocate computational resources to the most demanding regions of their partial differential equations simulations. AMR works by shrinking and blending cells in the computational domain where a value or region of particular interest lies. This affords the simulation code more accuracy in certain places, while allowing more general estimates in others. This ability to divide and rebalance makes simulations with AMR computationally more efficient than ones without it.

AMReX provides the necessary framework to adapt MFiX-DEM to exascale computations. By dividing the problem domain, each region can be concerned with only particles in their vicinity and ignore others. This means that only nearby particles need to be checked for collisions, dramatically reducing the N^2 cost mentioned earlier. This property combined with other time reducing techniques makes billion particle simulations possible.





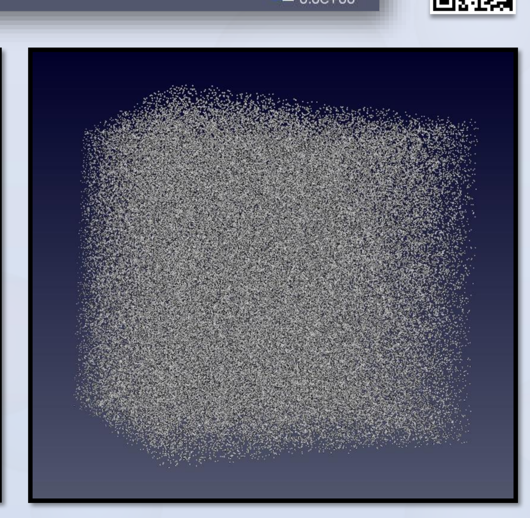


Figure 4. (Top) A simulation of 1200 particles moving freely in a stationary fluid. Figure 5. (Bottom Left) A 10,000 particle simulation involving multiple processes. Regions divided by process are color coded. Figure 6. (Bottom Right) A 200,000 particle simulation using 64 Intel Knight's Landing cores on a single Cori node at NERSC. Use the QR code (Top Right) to see animations on the CCSE website.

Modern HPC

Using AMReX also provides additional benefits for high performance computing (HPC). Built in to the AMReX code is the ability to incorporate both multicore processes with the use of MPI and multithreading with the use of OpenMP. As modern HPC moves towards lower powered multi-core CPUs, incorporating these technologies will be essential for achieving maximum utilization of large machines.

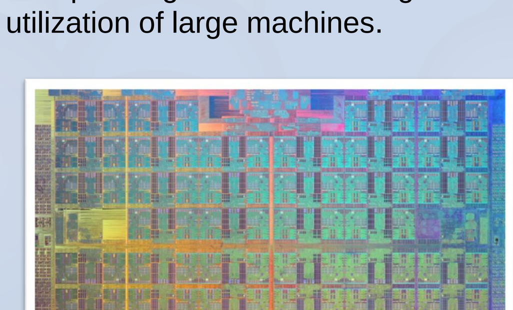
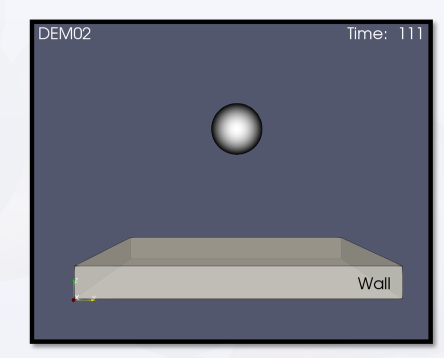


Figure 7. (Left). Intel's Knight's Landing central processing unit (CPU). In the image you can see the 68 cores that make up a single Knight's Landing chip. Utilizing such a large number of cores in an efficient way is a common challenge for HPC codes.

Figure 8. (Right). The Cori super computer at the National Energy Research Scientific Computing Center (NERSC). Cori houses 2,388 Intel Xeon "Haswell" and 9,688 Intel Xeon Phi "Knight's Landing" nodes, making it the world's 6[th] fastest machine.

Benchmark Problems

As the MFiX-DEM code is reworked to incorporate ARMeX and other improvements it is necessary to have a series of problems that ensure correctness and numerical accuracy of the code. The regression tests below were selected based on the ability to verify the answers analytically as well as the amount of computational resources they require to be completed [4]. After each code change is pushed to the remote repository, an automated script runs the tests and reports the results.



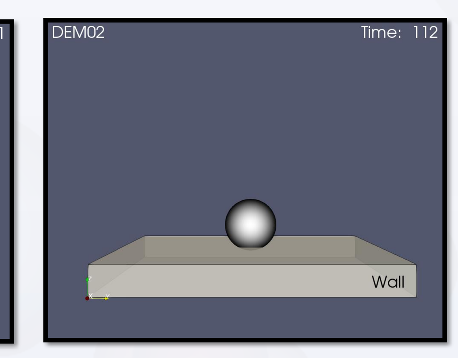


Figure 9. (Left) A single particle is allowed to freefall and impact a wall. The particle rebounds and strikes the wall several times. Heights are recorded and compared with previous results to verify particle collision behavior. Figure 10. (Right) A single particle striking the wall after freefall. Notice that the particle appears to penetrate the wall. This is because the visualization does not include deformations of the particle or wall.

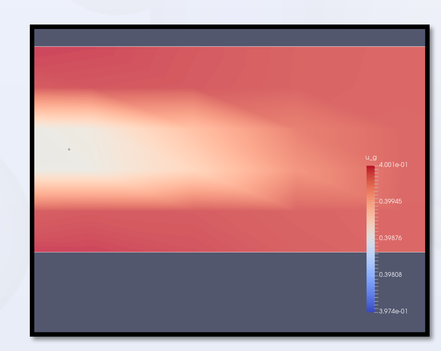
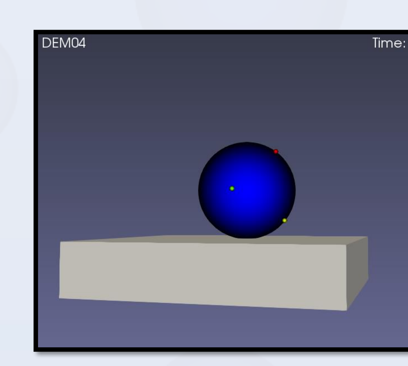


Figure 11. (Left). The image to the left shows a small particle in fluid flow. The velocity of the fluid is represented by the background heat map. In this simulation the particle accelerates with the fluid flow until it reaches a terminal velocity. This velocity is then compared with theoretical results.

Figure 12. (Right). The particle and wall shown right, are used to verify the rolling friction aspect of the particle behavior model. In this simulation the particle rolls along a wall. In the actual simulation output, only point data at the particle center and angular velocities are reported. In order to develop an animation which displays the rolling behavior, the three points on the surface were extrapolated. They are colored red, yellow and green for each axis they track.



Weak Scaling

Another important type of software testing for HPC codes is weak scaling. Weak scaling involves increasing the problem size in tandem with increasing the number of processors involved in running the code. Since large codes will be run across thousands of processors, its important that run times increase modestly with the problem scale. Weak scaling tests in MFiX-DEM are currently being done on the simulation of a fluidized bed.

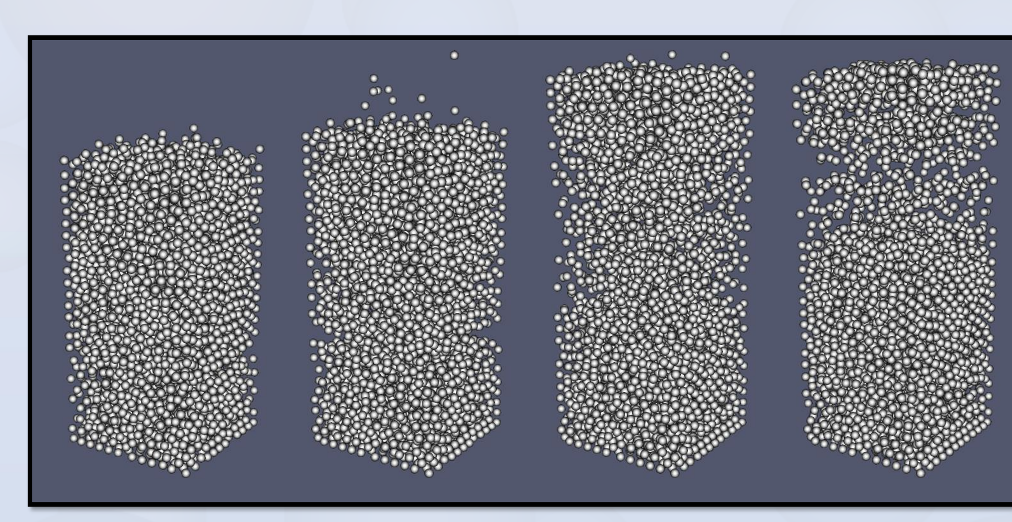


Figure 13. Four frames from the simulation of a fluidized bed where particles and gases interact. In a fluidized bed, gas rises from beneath, creating pockets of gas that push the particles upward in a bubbling fashion. In this image you can see the particles rise with the gas and then begin to fall after the gas rises through them.

References

[1] "MFiX-Exa project selected as Department of Energy's Exascale Computing Project", NETL Multiphase Flow Science, Sept. 14, 2016, Web.
[2] Q. J. Zheng, H.P. Zhu, A.B. Yu, "Finite element analysis of the contact forces between a viscoelastic sphere and rigid plane," in *Powder Technology*, no. 226 (2012): 130-142
[3] K.A. Buist, L.J.H. Seelen, N.G. Deen, J.T. Padding, J.A.M. Kuipers, "On an efficient hybrid soft and hard sphere collision integration scheme for DEM" in *Chemical Engineering Science* no.153 (2016): 363-373
[4] J. Musser and A. Choudhary, "MFiX Documentation Volume 3: Verification and Validation Manual," NETL Multiphase Flow Science, Nov. 9, 2015, Web.

Acknowledgements

This research was supported in part by an appointment with the NSF Mathematical Sciences Summer Internship Program sponsored by the National Science Foundation, Division of Mathematical Sciences (DMS). This program is administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and NSF. ORISE is managed by ORAU under DOE contract number DE-SC0014664.

U.S. DEPARTMENT OF ENERGY NSF OAK RIDGE INSTITUTE FOR SCIENCE AND EDUCATION