Open-Source Development of a Gas/Particle Flow Problem Solving Environment

Thomas J. O'Brien, National Energy Technology Laboratory, Morgantown, WV
Sreekanth Pannala, Oak Ridge National Laboratory, Oak Ridge, TN
Madhava Syamlal, Fluent, Inc, Morgantown, WV
Michael Prinkey, Fluent, Inc, Morgantown, WV
Aytekin Gel, Aeolus Research, Morgantown, WV
Philip Nicoletti, Parsons, Inc, Morgantown, WV
Sofiane Benyahia, Fluent, Inc, Morgantown, WV

Over the past decade, the National Energy Technology Center (NETL) has developed an open-source code for the simulation of heavily-loaded, reactive gas/particle flows, MFIX (Multiphase Flow with Interphase eXchange). The programmatic purpose of this development has been to model multiphase flow processes in power and process industry (*e.g.*, coal gasifiers) so as to increase the use of modeling for design and control. MFIX has been used to develop and validate multiphase flow theory (transport equations and constitutive relations) and to develop numerical techniques for solving these equations efficiently and accurately. It has been downloaded from the web site www.mfix.org by over 350 scientists from around the world. Features of the code include:

Physical features

- the solution of mass, momentum, energy and species balance equations for gas and multiple solids phases including a user defined scalar transport equation;
- granular stress equations based on kinetic theory and frictional flow theory;
- impermeable and semi-permeable internal surfaces;
- flexible chemical reactions and kinetics, defined with the input data file or with a user-defined subroutine;
- multiphase turbulence models;
- Discrete Element Method coupled to Eulerian-Eulerian formulation;
- alternative gas-particle drag laws
- polydispersity, including Direct Quadrature Method of Moments

Numerical features

- three-dimensional, nonuniform meshes with Cartesian or cylindrical coordinate systems;
- error checking of user input;
- multiple, single-precision, binary, direct-access output files that reduces disk space and increases data retrieval speed;
- customized post-processing tools for the animation and retrieval of output data;
- Fortran 90 code base that enables dynamic memory allocation depending on the size of the problem;
- compilation scripts that can generate serial, shared-memory parallel (SMP) or distributed-memory parallel (DMP) executables from the same code base on a variety of supported hardware platforms.

Currently the MFIX group is assessing plans for the development of the "next generation" version of this code. The intended application domain of the code is for dilute to dense gas-solid flows with chemical reactions and heat transfer, including radiation. It will incorporate both Eulerian-Eulerian and Eulerian-Lagrangian representations of particle motion and operate within a framework to achieve scalable performance on large scale parallel

computing platforms. This version would enable scientists to focus on model and algorithm development and validation, rather than on code development and debugging, thereby providing a substantially new capability that is useful for research and not available elsewhere. Future MFIX development would continue to: be complementary and contributory to commercial software; be a vehicle of technology transfer for theoretical multiphase flow; facilitate dissemination of information through shared software; and enhance collaboration with other labs, academia, commercial companies and other agencies. The intended user of the code is researchers who would like to change the theory and/or numerical method and thus need to have access to the details of the simulation.

The software design goals are to: use the best-in-class software components from different sources (academia, national labs, open source and commercial); provide a fully functional, minimum set of features that is based completely on public domain software components; allow the use of software components in different languages (Fortran 90, C, C++, *etc.*); use modern software tools and practices such as object oriented design, automated building, testing and generation of documentation; conform to well established interface standards (CCA, CORBA, and COM); use a scripting front end to enable customization. It will use well-defined, immutable interfaces to link components and be portable, scalable and extensible. The overall goal of this project is to facilitate harnessing the power of next generation computers to understand, improve and control chemical reactors. This presentation will provide details on the current status of this development project.

It is intended that this program will be able to utilize software made available as Open-Source software by other government organizations (*e.g.*, DOE, NASA, NFS) or university groups (*e.g.*, University of Oregon - Program Database Toolkit, Rutgers University - Grid Adaptive Computational Engine).

One package of particular interest is *Overture*, under development at Lawrence Livermore National Laboratory (http://www.llnl.gov/casc/Overture/). Overture is an object-oriented code framework for solving partial differential equations. It provides a portable, flexible software development environment for applications that involve the simulation of physical processes in complex moving geometries. It is implemented as a collection of C++ libraries that enable the use of finite difference and finite volume methods at a level that hides the details of the associated data structures.