lowmachSolver: a low Mach number solver for spray simulation with OpenFOAM

Rodrigo B. Piccinini

December 24, 2011

Abstract

lowmachSolver is an OpenFOAM solver that deals with two-phase flows composed of a continuous gaseous phase and a liquid dispersed phase.

1 Introduction

lowmachSolver is an OpenFOAM solver that deals with two-phase flows composed of a continuous gaseous phase and a liquid dispersed phase. It is mainly based on *dieselFOAM* and it was written only for purpose of my master thesis.

1.1 OpenFOAM Code

By OpenCFD own words:

> The OpenFOAM (Open Field Operation and Manipulation) CFD Toolbox is a free, open source CFD software package produced by OpenCFD Ltd. It has a large user base across most areas of engineering and science, from both commercial and academic organisations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics. It includes tools for meshing, notably snappyHexMesh, a parallelised mesher for complex CAD geometries, and for pre- and post-processing. Almost everything (including meshing, and pre- and post-processing) runs in parallel as standard, enabling users to take full advantage of computer hardware at their disposal.
>
> By being open, OpenFOAM offers users complete freedom to customise and extend its existing functionality, either by themselves or through support from OpenCFD. It follows a highly modular code design in which collections of functionality (e.g. numerical methods, meshing, physical models, ...) are each compiled into their own shared library. Executable applications are then created that are simply linked to the library functionality. OpenFOAM includes over 80 solver applications that simulate specific problems in engineering mechanics and over 170 utility applications that perform pre- and post-processing tasks, e.g. meshing, data visualisation, etc.

Details of OpenFOAM formulation are explained in [3] and [6]. From all the set of available solvers and libraries in OpenFOAM, the dieselFOAM solver and dieselSpray class (as implemented in version 1.7.1 of OpenCFD release) were the major pieces of code used in this work. To my knowledge, both were written by Niklas Nordin, [4].

The dieselSpray class handles the modeling of lagrangian particles and their submodels. Minor modifications were made in order to have more flexibility in boundary conditions and to adapt them to the experimental conditions.

The dieselFOAM solver couples the modeling of the lagrangian particles and the gas flow solution. The spray sources are explicitly treated and the coupling among variables is solved with PISO algorithm, see [3] and [2].

Minor modifications were added to the solution of low Mach number equations instead of the fully compressible formulation. They are briefly explained here, but the understanding requires from the reader some familiarity with OpenFOAM programming.

The thermodynamic pressure retained its original name `p` and is the pressure used in the state equation:

```
<createFields.H>
volScalarField& p = thermo.p();
```

and in the lagrangian models:

```
<createSpray.H>
spray dieselSpray
(
    U,
    rho,
    p,
    T,
    composition,
    gasProperties,
    thermo,
    g
);
```

A new scalar field was assigned to the dynamic pressure, `volumeScalarField pd`:

```
<createFields.H>
 volScalarField pd
(
    IOobject
    (
        "pd",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

The momentum equation was modified to be computed using the gradient of `pd` instead of `p` in the momentum predictor:

```
<UEqn.H>
     fvVectorMatrix UEqn
    (
        fvm::ddt(rho, U)
      + fvm::div(phi, U)
      + turbulence->divDevRhoReff(U)
     ==
        rho*g
      + dieselSpray.momentumSource()
    );

    if (momentumPredictor)
    {
        solve(UEqn == -fvc::grad(pd));
    }
```

Finally, the the pressure equation is now a Poisson equation for the dynamic pressure and it uses the thermodynamic pressure for computing the density.

```
<pEqn.H>

  fvScalarMatrix pdEqn
  (
      fvc::ddt(psi,p)
      + fvc::div(phi)
      - fvm::laplacian(rho*rUA, pd)
      ==
      Sevap
  );
```

where `psi` or Ψ is the isothermal compressibility. For an ideal gas:

$$\rho = p\Psi = \frac{p}{RWT}\,. \tag{1}$$

The time-dependence of the thermodynamic pressure was neglected (valid for open domains) and the therm `fvc::ddt(psi,p)` vanishes.

2 Folder Structure

The thesis work is organized in a typical SVN folder structure:

```
--- branches
--- tags
--- trunk
--- wiki
```

After finishing the thesis, everything was merged to **trunk** folder and the others were left empty.

./trunk folder has the following structure:

```
--- trunk
    +-- case
    |   +-- halle test case from Sommerfeld, and H.-H. Qiu (not fully finished).
    |   |   +-- 2D 2D mesh and case setup.
    |   |   +-- exp experimental data.
    |   |   +-- misc miscellaneous Python scripts.
    |   |
    |   +-- sydney test case from Chen et al (Sydney University). In the thesis.
    |       +-- case fine mesh setup.
    |       +-- case.coarse coarse mesh setup.
    |       +-- ic.coarse Initial conditions for the coarse mesh case.
    |       +-- ic.fine Initial conditions for the fine mesh case.
    |       +-- misc miscellaneous Python scripts for plotting results and building mesh.
    |
    +-- code
    |   +-- 1.7.x code based on 1.7.1 release from OpenCFD
    |       +-- lowmachLib lagrangian spray lib.
    |       +-- lowmachSolver low Mach number solver coupled with lowmachLib.
    |       +-- myLiquids acetone liquid and vapor properties
    |       +-- mypdfs spray probability density functions
    |
    +-- paper thesis results in paper format
    |   +-- elsevier Elsevier latex class
    |
    +-- ppt presentation given in 2011/nov/25 at ITA.
    |   +-- imgs figures in presentation.
    |       +-- visit figures post-processed in visIT
    |
    +-- thesis thesis .tex source files.
    |   +-- figuras figures with .dat files for each of the subsequent thesis appendix and
    |       +-- appA1
    |       +-- appA2
    |       +-- appA3
    |       +-- chap1
    |       +-- chap2
    |       +-- chap3
    |       +-- chap4
    |       +-- chap5
    |
    +-- tutorial this tutorial source files.
+-- html html files from latex2html.
```

3 Compiling

Compiling was tested with Ubuntu Natty (numbered 11.04) and .deb package of OpenFOAM 1.7.1 released by OpenCFD.

http://www.openfoam.org/archive/1.7.1/download/

Development tools for Ubuntu must be present. They can be installed by:

```
 sudo apt-get install build-essential flex bison cmake zlib1g-dev qt4-dev-tools libqt4-dev
```

For parallel runs, installation of the following libraries is also required:

```
sudo apt-get install libscotch-dev libopenmpi-dev
```

Code is located in ./trunk/code/1.7.x/

```
--- trunk
    +-- code
        +-- 1.7.x code based on 1.7.1 release from OpenCFD
            +-- lowmachLib lagrangian spray lib.
            +-- lowmachSolver low Mach number solver coupled with lowmachLib.
            +-- myLiquids acetone liquid and vapor properties
            +-- mypdfs spray probability density functions
```

Inside ./trunk/code/1.7.x/, the following scripts may be issued to:

```
# ./Allwmake compile the solver and all needed libraries.
# ./CleanSrc clean the source files for a new compiling.
# ./CleanAll clean both source and binary files for a new compiling.
```

Allwmake script will make a user installation after compiling the code.

4 Running

lowmachSolver can be run as any OpenFOAM solver with usual options:

```
#lowmachSolver -help

Usage: lowmachSolver [-parallel] [-case dir]  [-help]
```

See Test Case section 5 for an example.

5 Test Case

This is the test case used to provide results for the thesis. It consists of the numerical simulation of the experiment performed by [1] in Sydney University.

http://dx.doi.org/10.1016/j.ijmultiphaseflow.2005.09.002

Case folder structure

```
--- case
    +-- 0 initial and boundary conditions
    +-- chemkin chemkin files
    +-- constant mesh and physical properties
    +-- system solution parameters
```

./case/0/ Folder **case/0/** contains initial and boundary conditions. The type of each boundary conditions is presented in the thesis [5] and may be directly read in the files as well.

The initial conditions of the gaseous flow were obtained by simulating only the gas flow in ../cases/sydney/ic.fine/ and ../cases/sydney/ic.coarse/.

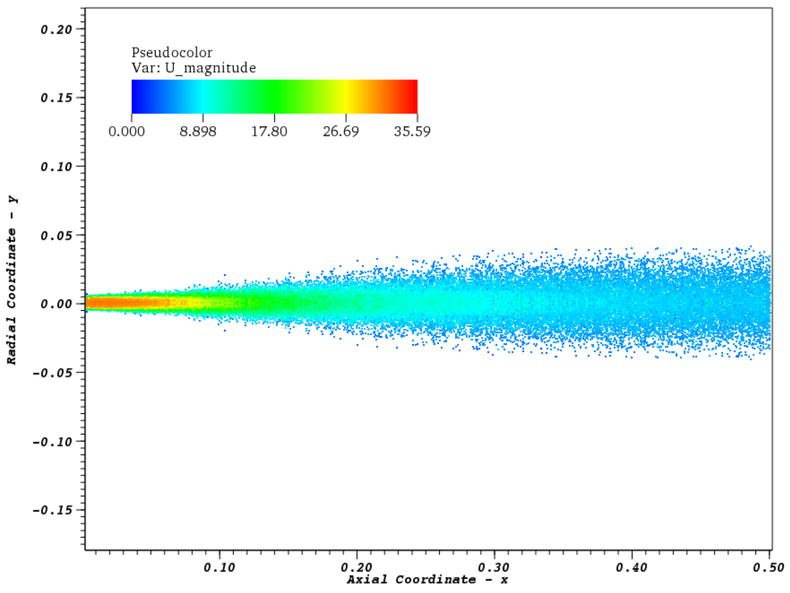

Figure 1: Magnitude of droplet velocities. The computational domain is only half of the shown above, it is here mirrored for visualization purpose.

./case/chemkin/ The chemkin folder contains chemical kinetic and thermodynamic data of all present species. No chemical reaction is present. Thermodynamic and transport properties of acetone are computed directly in the code.

./case/constant/

5.1 Some Results

6 Links

Openfoam: OpenFOAM OpenFOAM-extend
Academics: ITA LCFT/ITA

References

[1] Y.C. Chen, S.H. Stårner, and A.R. Masri. A detailed experimental investigation of well-defined, turbulent evaporating spray jets of acetone. *International journal of multiphase flow*, 32(4):389–412, 2006.

[2] J.H. Ferziger and M. Perić. *Computational methods for fluid dynamics*, volume 2. Springer Berlin, 1999.

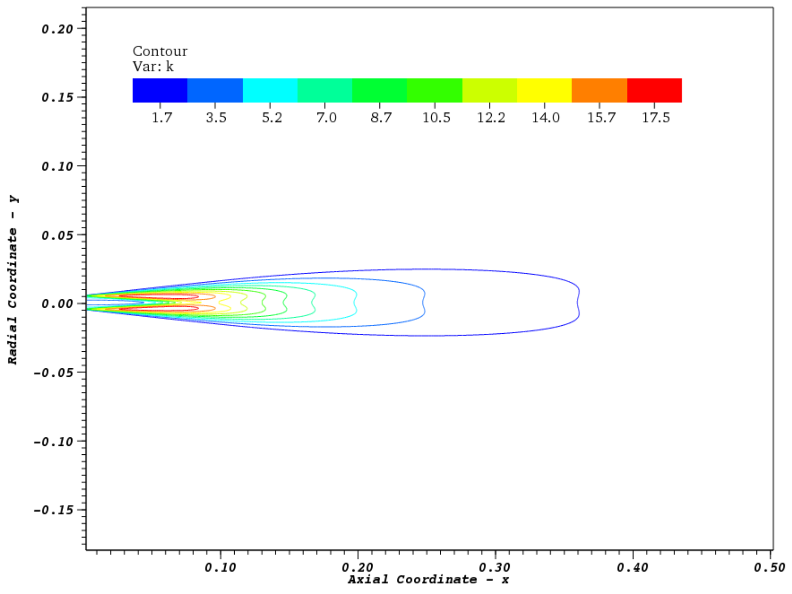

Figure 2: Gas mean turbulent kinetic energy. The computational domain is only half of the shown above, it is here mirrored for visualization purpose.

[3] H. Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows.* PhD thesis, Imperial College, London, June 1996.

[4] PA Nordin. *Complex chemistry modeling of diesel spray combustion.* PhD thesis, Chalmers University of Technology, Goteborg, 2001.

[5] R. B. Piccinini. Eulerian-lagrangian simulation of a turbulent evaporating spray. Master's thesis, ITA, 2011.

[6] H.G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12:620, 1998.