

# Practical Options for Desktop CFD Using Open-Source Software

David A. Janes

Department of Chemical Engineering, The University of the West Indies, St Augustine, Trinidad and Tobago, West Indies  
E-mail: david.janes@sta.uwi.edu

(Received 30 June 2011; Revised 19 November 2011; Accepted 23 January 2012)

**Abstract:** *There is an opportunity to develop a centre of competence for Computational Fluid Dynamics (CFD) to meet the demands of the process industries in Trinidad and Tobago. The CFD skills and techniques could be developed and used to foster improvements in the performance of the process industries. A review was undertaken of available open-source CFD software that is available in compiled executables, which runs on the Microsoft Windows line of operating systems, as a means of keeping budgets down while still building up local expertise. Two pre-processors (Gmsh and Netgen), three CFD-solver programs (Elmer, OpenFOAM and OpenFVM) and one post-processor (ParaView) were scoped. From these packages, a useful CFD toolbox consisting of a pre-processor, a solver and post-processor could perform useful CFD simulations on the Microsoft Windows platform. The right combination would depend on the future intentions of the CFD-user.*

**Keywords:** *Simulation of Industrial Processes, Computational Fluid Dynamics, Computer-Aided Engineering, Open Source Software, Trinidad and Tobago*

## 1. Introduction

Computational Fluid Dynamics, often abbreviated by the acronym CFD, is a simulation technique, which employs numerical methods and algorithms to solve problems involving fluid flow (Wilkes, 2006). The representation of the interactions between liquids and gases with surfaces, defined by boundary conditions, needs such computing performance and memory resources that the simulation of fluid flows will be a challenge for decades. So even with clusters of high performance computers, simplifying the models is part of the daily business. Nevertheless, CFD has matured from a mathematical and engineering oddity to become an indispensable tool used in the development of weather forecasts, aircraft and the simulation of furnaces (For a straightforward introduction to CFD, see either Shaw (1992) or Wilkes (2006)).

Published citations of CFD simulations of industrial process equipment in Trinidad and Tobago, such as a methanol plant and offshore gas pipelines, were recently reviewed (Janes, 2010a). It was found that CFD simulations were used for a variety of purposes such as process design, safety studies and performance evaluations. When an enterprise seeks to acquire engineering skills and techniques, it is good practice to source the enabling goods and services efficiently just like all indirect goods and services (Janes, 2010b). However, all the simulations were performed outside the Caribbean region. Hence, there is an opportunity and incentive to develop a centre of competence for CFD in Trinidad and Tobago.

Since developing CFD skills can have a steep

learning curve (Janes, 1996), a possible strategy is to find software that runs on existing computers within the enterprise, such as laptops and desktop personal computers (PC). Therefore, because the Microsoft Windows line of operating systems is ubiquitous on desktop computers, the utility of some CFD open-source software (OSS) that runs on these platforms has been investigated. Proprietary CFD packages that run in Microsoft operating systems on a desktop PC have been available, at some cost, since the mid-nineties (Janes, 1996). According to Janes (1996), it was not economically practical for small and medium businesses operating manufacturing processes to adopt the technology.

CFD specialists use high performance workstations, because they are optimised for the visualisation and manipulation of complex data such as 3D engineering simulation, rendering of images and animation. As is common with OSS, most open-source CFD software seems to be developed on and for the Unix-like Linux operating system, which is OSS itself. When starting out to build such centres of competence in emerging countries, OSS that works effectively on the Windows platform would be an advantage, because of the high costs of both the high performance computing systems and the commercial software packages.

When experience has been built up, a more informed decision can be made on purchasing appropriate computer hardware and whether to purchase a commercial CFD software package or a consultancy agreement with an OSS supplier. Alternatively, an engineer, whose company intends to contract a foreign

or local CFD consultancy to perform some simulations, may want to gain some simple understanding of the technology, so that what is desired, what is possible and what has been achieved are more effectively conveyed.

A CFD toolbox consists of three parts; a pre-processor (e.g. for grid generation), a solver (the CFD program), and a post-processor (e.g. for visualisation) (Shaw, 1992). An attempt was made to identify the scope and limitations of such OSS tools (including three solver programs) that are available in compiled executables to run on a non-specialist engineer's run-of-the-mill PC.

## 2. Pre-processors

The pre-processor program is used to define the problem to be analysed. When defining the geometry of a flow domain, geometrical data are required and sometimes this will come from a CAD system but most often a finite element pre-processor is used. The pre-processor program has to be given sufficient information for it to produce the data that the solver needs. In particular, mesh data is needed. These may be produced by the solver program but are more easily produced by a pre-processor. Two pre-processor examples, Gmsh and Netgen, were looked at in this paper.

### 2.1 Gmsh

Gmsh is a three-dimensional grid generator with built-in pre- and post-processors for grid-based numerical methods. The goal of the designers (Geuzaine and Remacle, 2009) was to provide a fast, light and user-friendly meshing tool with parametric input and advanced visualisation capabilities. Pre-compiled versions of Gmsh are available for Unix, Windows and Mac OS as well as in pre-packaged form for Linux and BSD. Gmsh consists of a very small kernel with four modules (such as geometry, mesh, solver and post-processing), not tied to any particular computational solver, and can be driven either using a user-friendly graphical interface (GUI) or text files using its own scripting language. This makes it possible to automate all treatments, using loops, conditionals and external system calls.

#### 2.1.2 Netgen

Netgen (Schöberl, 1997) is an automatic mesh generation tool for two and three dimensions with graphical user interface, primarily for end users who want an "off the shelf" tool for Finite Element Analysis (FEA) or CFD applications. Netgen is available for Unix/Linux, Windows and Mac OS. Input for 3D problems can be defined either by constructive solid geometries (CSG) or by the standard STL file format. When compiled with OpenCascade support (a CAD/CAE platform for 3D modelling applications) Netgen accepts input as Boundary Representations - BRep, IGES and STEP formats. Netgen generates

triangular or quadrilateral meshes in 2D, and tetrahedral meshes in 3D. Netgen contains modules for mesh optimisation and hierarchical mesh refinement. Figure 1 shows a triangular mesh that was created in Netgen and imported into the ElmerGUI.

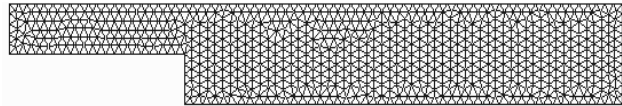


Figure 1. A triangular mesh created with Netgen and read into Elmer

### 2.3 Pre-processor Discussion

Gmsh is one the most popular academic mesh generation OSS packages, and is quite capable of building the simple mesh geometries a CFD novice requires. The more sophisticated Netgen could be more appropriate for someone who does not want to be confined by the limitations of Gmsh and, because Netgen can be connected to the powerful OpenCASCADE, it permits the development of expertise that will be useful in the long term.

## 3. Solvers

The output from the pre-processor is included in the input data to the solver program that solves sets of algebraic equations to produce a CFD solution. In most CFD simulations, these sets of equations are based on partial differential equations (PDE). For instance, the low-speed flow of an incompressible fluid is described by the continuity of mass equation (or simply the continuity equation) and the momentum equations (often known as the Navier-Stokes equations), which are derived from Newton's Second Law and account for the conservation of momentum in fluid flow (Wilkes, 2006). So the task is to solve simultaneously these equations of motion numerically, which means finding approximate solutions for the PDEs (Shaw, 1992). Most solvers use, either the finite element method (FEM) or the finite volume method (FVM).

The FVM is popular for approximating transport phenomena in CFD mainly due to two reasons. Firstly, it ensures that the discretisation is conservative, which means that the integral form of the conservation laws is satisfied to some degree of approximation for all control volumes of the computational domain. Secondly, it does not require a coordinate transformation when applied to irregular meshes, so that it can be easily adapted to unstructured meshes consisting of arbitrary polyhedra.

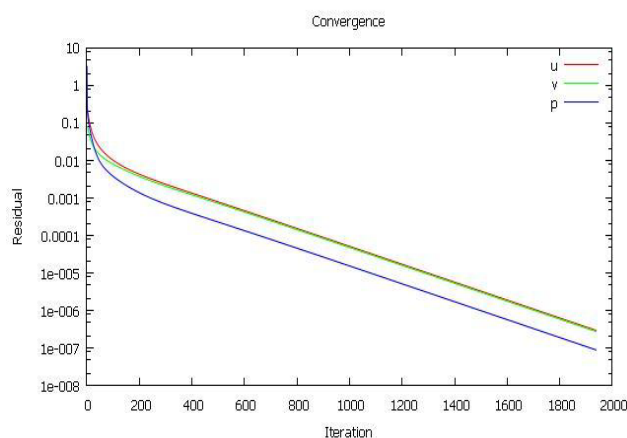
Because of its wide range of application areas, the FEM can be used to simulate CFD problems and mechanical problems (e.g. stress simulation). The OSS CFD solver packages reviewed in this paper, include one based on the FEM, Elmer (ver 6.0) and two using the FVM, OpenFOAM (ver 1.5) - or more precisely

OpenFOAM for MS Windows - and OpenFVM (ver. 1.3)

### 3.1 OpenFVM

OpenFVM is a general open source three-dimensional CFD solver for Linux and Windows. It was developed by a mechanical engineering graduate student from The University of Minho (Araujo et al., 2009) to simulate the flow in complex 3D geometries. It uses the unstructured FVM to simulate non-isothermal transient flow. Therefore, the mesh can be unstructured and contain control volumes with arbitrary shape. As well as solving the velocity and pressure fields, the code is capable of solving non-isothermal two-phase flow. Gmsh is used for pre- and post-processing. Input data for OpenFVM is put in four different files which define; the mesh (\*.msh), the boundary conditions (\*.bcd), materials (\*.mfl) and solver parameters (\*.par) such as tolerances, maximum number of iterations and output options.

The results for post-processing are stored in Gmsh human-readable “parsed” format files (\*.pos), so that the residuals convergence history can be tracked. As an example Figure 2 shows how the convergence of residuals from a lid-driven cavity simulation can be displayed with Gmsh. Lid-driven cavity flow is one of the most studied fluid flow problems in CFD, as the rectangular geometry and boundary conditions (i.e., a steadily moving lid and stationary walls) are simple and can be easily reproduced (Araujo et al., 2009).



**Figure 2.** Tracking convergence of residuals from a lid-driven cavity simulation

Unfortunately, OpenFVM only has one turbulence model, the industry standard k-epsilon model. This two-equation model is commonly used for most engineering problems. A second limitation is that the solver runs on Windows in a command-line shell, so the user has to be sufficiently knowledgeable to use the command-line interface.

### 3.2 OpenFOAM

The OpenFOAM (Open Field Operation and Manipulation) CFD Toolbox is a CFD software package produced by Silicon Graphics International. Development started originally in the late 1980s at Imperial College, London and was then carried on by a commercial company, OpenCFD Ltd. 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. At the core of OpenFOAM is a flexible set of efficient C++ modules (Weller et al., 1998) that are used to build a wide variety of solvers, utilities and libraries. The solvers simulate specific problems in engineering mechanics. The utilities perform pre- and post-processing tasks ranging from simple data manipulations to visualisation and mesh processing. The libraries are used to create toolboxes that are accessible to the solvers/utilities (e.g. libraries of physical models).

OpenFOAM has a large user base across most areas of engineering and science, from both commercial and academic organisations. It compares favourably with the capabilities of most leading general-purpose commercial closed-source CFD packages. OpenFOAM relies on the user's choice of third party pre- and post-processing utilities. OpenFOAM is officially released only as a binary for Linux platforms.

Whereas at the time of writing the current version of OpenFOAM is 2.0, there is a free native MS Windows release of OpenFOAM ver 1.5 from 2008. It was built with MinGW C++ as a set of native windows applications, which improve performance and eliminate the need for Unix emulation. It is supplied with a GUI called OpenFOAM Designer, which is intended to be an intuitively straightforward working environment for OpenFOAM applications. This “OpenFOAM for MS Windows” is the package that was used in this investigation. There is no guarantee that there will be a free updated binary release for Windows in the future, although more recent binaries-for-Windows versions are available for purchase from other third-party suppliers.

### 3.3 Elmer

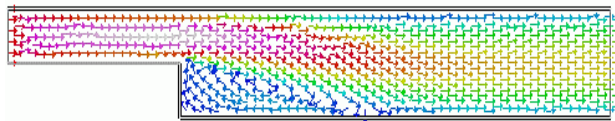
Elmer is a finite element software package for multi-physics simulations (Råback et al., 2007). Elmer can deal with a great number of different equations, which may be coupled in a generic manner making Elmer a versatile multi-physics tool. The following is a list of some of the specialised fields in which Elmer has a capability:

- 1) Heat transfer - models for conduction, radiation and phase change;
- 2) Fluid flow - the Navier-Stokes, Stokes and Reynolds equations, k- $\epsilon$  model;
- 3) Species transport - generic convection-diffusion equation;
- 4) Elasticity - general elasticity equations,

dimensionally reduced models for plates and shells;

- 5) Acoustics - the Helmholtz equation;
- 6) Electromagnetism - electrostatics, magnetostatics, induction.

The Elmer software package comprises several different parts that may also be used independently. The boundary conditions and physical models are defined in ElmerGUI. The resulting problem definition is solved by ElmerSolver, the most powerful of the component parts. Finally, ElmerPost may be used to visualise the results as in Figure 3, which shows the velocity field from the simulation of laminar flow past a step.



**Figure 3.** Simulated velocity vectors of laminar flow past a step in Elmer

The development of Elmer began in 1995 as part of a national CFD technology program in Finland. Since the initial project ended the development has been continued by the Finnish CSC – IT Center for Science Ltd in different application fields. In September 2005 Elmer was released under the GNU General Purpose Licence, which has widened the user community. In particular, the number of international users has grown. As open source software, Elmer gives the user the means to modify the existing solution procedures and to develop new solvers for equations of interest to the user. Elmer is easily compiled for most Unix systems, and it is also available for Windows and Macintosh machines as precompiled binaries.

Elmer has a number of weaknesses. For example, there are no strategies available for high Mach numbers and there is no proper multiphase modelling capability. Being based on the FEM, Elmer is more suited to elliptic problems rather than parabolic problems with their conservative nature being inherently more suitable for FVM solvers such as OpenFOAM and OpenFVM.

### 3.4 Solver Package Discussion

Of the three solvers, OpenFVM is the smallest package. If the only intention is to learn about CFD by solving simple systems on the MS Windows platform, OpenFVM is a good starting package.

On a Unix platform OpenFOAM compares well against most leading general-purpose closed-source CFD packages. However, support for the Windows platform is haphazard, whereas CSC provides a Windows executable. Perhaps this is one reason Elmer has a steadily growing user community but CSC estimate it is an order of magnitude smaller than OpenFOAM's.

Both OpenFOAM and Elmer often rely on external

mesh generators, and both may use ParaView for post-processing. Basic usage of Elmer is perhaps easier, since it comes with a well-integrated GUI, the file system is simpler compared with OpenFOAM's and a first solution can be obtained in minutes. For advanced CFD usage, OpenFOAM may be better, owing to cleaner basic programming and because there is more potential support and advanced study sharing from amongst the larger peer community of users. A comparative strength of Elmer is its capacity to deal with multiphysics coupling of fluid flow with structures, electromagnetism or heat.

## 4. Post-processors

When a CFD-solver runs, it produces a large amount of data that have to be analysed. By sending CFD-solver results to a post-processor or some other display software, the visualisation of data is separated from their creation, which can be an ergonomic advantage. Often, the post-processor allows data exploration to be done interactively or using batch processing. ParaView is the recommended external postprocessor for Elmer and an interface is provided to it in OpenFOAM. The smaller OpenFVM is designed to work with the post-processor in Gmsh.

### 4.1 Gmsh's Post-processing Module

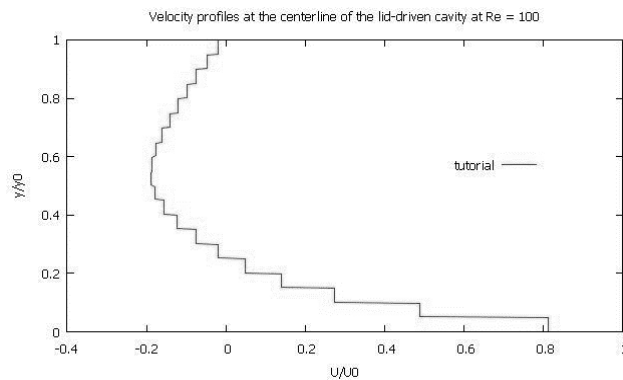
The post-processing module in Gmsh can be used to visualise multiple scalar, vector or tensor datasets along with the geometry and the mesh. Therefore, CFD results can be given in several formats: in human-readable "parsed" format files with a '.pos' extension, in native MSH files (ASCII or binary files with '.msh' extensions, or in standard third-party formats (e.g. MED files). Once loaded into Gmsh, scalar fields can be displayed as iso-value lines and surfaces or colour maps, whereas vector fields can be represented either by three-dimensional arrows or by displacement maps.

In Gmsh's jargon, each dataset is called a "view". Each view is given a name, and can be manipulated either individually or globally. By default, Gmsh treats all post-processing views as three-dimensional plots, i.e., draws the scalar, vector and tensor primitives (e.g., points, lines, triangles, and tetrahedra, etc.) in 3D space. However, Gmsh can also represent each post-processing view containing scalar points as either space- or time-oriented two-dimensional plots. Figure 4 depicts results from a lid-driven cavity simulation that were post-processed with Gmsh to extract the centreline velocity profile.

Although visualisation is usually mostly an interactive task, Gmsh's scripting language permits the post-processing process to be a completely automated.

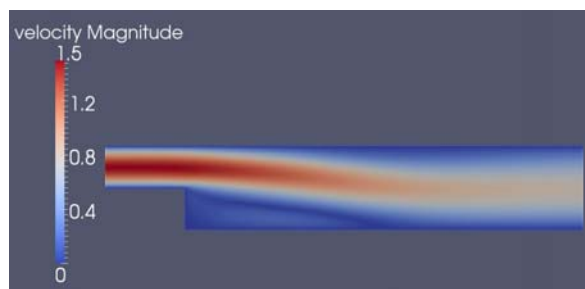
### 4.2 ParaView

ParaView is a scalable, multi-platform data analysis and visualisation application for two- and three-dimensional



**Figure 4.** OpenFVM simulation of lid-driven cavity flow post-processed in Gmsh

data sets. Figure 5 shows data from an Elmer simulation of flow past a step that has been visualised in Paraview. The application runs on a wide variety of platforms, ranging from a single-processor desktop machine to multi-processor shared-memory supercomputers or clusters of computers. The ParaView project started in 2000 as a collaborative effort between Kitware Inc. and Los Alamos National Laboratory. ParaView is based on the Visualisation Toolkit (VTK) library, a software system for 3D computer graphics, image processing, and visualisation. The size of the data sets ParaView can handle varies widely depending on the architecture on which the application is run.



**Figure 5.** Post-processing of Elmer simulation of flow past a step in Paraview

ParaView is regularly used by Sandia National Laboratories analysts (Cedilnik et al, 2006) to visualise meshes containing up to 6 billion structured cells and 250 million unstructured cells, and billions of cells in structured grids. ParaView has an open, flexible, and intuitive user interface. The extensible, modular architecture is based on open standards and commercial maintenance and support is available from Kitware Inc. Hence, it is not surprising that ParaView is used by many academic, government, and commercial institutions all over the world, and ParaView is downloaded about 3 thousand times each month.

### 4.3 Post-processor Discussion

Because ParaView is an industry standard and the VTK library supports a wide variety of visualisation algorithms including scalar, vector, tensor, texture, and volumetric methods, it is a powerful OSS package for visualising CFD results. However, the smaller, lightweight Gmsh is adequate for the post-processing of results from simple small simulations.

### 5. Conclusions

It is possible to perform CFD simulations with OSS running on the Microsoft Windows platform. From the OSS described above, a useful toolbox consisting of a pre-processor, a solver and post-processor could perform credible CFD simulations. The right combination would depend on the future intentions of the CFD-engineer. If the user just wants to gain some CFD experience, this could be accomplished with a combination of OpenFVM and Gmsh. However, if the user intends to do more advanced CFD simulations, but stick to the Microsoft Windows platform, a combination of Elmer, Netgen or Gmsh and ParaView is an attractive option.

Although in the long term, a Unix-like platform may still be better for a dedicated CFD workstation, in which case Elmer should be substituted with OpenFOAM. Moreover, since both are OSS, both the FEM Elmer and FVM OpenFOAM could be included in the CFD software toolbox and used according to the type of simulation problem that needs solving. This could be a means to start accumulating locally experience with CFD skills and techniques, which could then be used to fulfill some of the simulation needs of the process industries in Trinidad and Tobago and the Caribbean region.

### References:

- Araujo, B.J., Teixeira, J.C.F., Cunha, A.M. and Groth, C.P.T. (2009), "Parallel three-dimensional simulation of the injection molding process", *International Journal for Numerical Methods in Fluids*, No.59, pp.801-815
- Cedilnik, A., Berk Geveci, B., Moreland, K., Ahrens J. and Favre, J. (2006), "Remote large data visualisation in the ParaView framework", In: Heirich, A., Raffin, B. and dos Santos, L.P. (eds), *Eurographics Parallel Graphics and Visualisation*, May 2006, pp.163-170,
- Geuzaine, C. and Remacle, J.-F. (2009), "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities", *International Journal for Numerical Methods in Engineering*, Vol. 79, No.11, pp.1309-1331
- Janes D.A. (2010a), "Desktop computational fluid dynamics with open-source software", *Proceedings of the Second Industrial Engineering and Management Conference*; Faculty of Engineering, UWI, St Augustine, Trinidad & Tobago, October 2010, pp.78-83
- Janes D.A. (2010b), "Sourcing-improvement experiences in medium-sized enterprises with dispersed sites", *Proceedings of the Second Industrial Engineering and Management Conference*; Faculty of Engineering, UWI, St Augustine, Trinidad & Tobago, October 2010, pp.70-77
- Janes, D. (1996), *How to Use CFD in the Food Industry*,

- Leatherhead Food Research Association, Leatherhead, England available via <http://hdl.handle.net/10068/695304>
- Råback, P., Forsström, P.-L., Lyly, M. and Gröhn, M. (2007), "Elmer - finite element package for the solution of partial differential equations", *EGEE User Forum* (Poster Presentation), Manchester, UK
- Schöberl, J. (1997), "NETGEN - an advancing front 2D/3D-mesh generator based on abstract rules", *Computing and Visualisation in Science*, Vol.1, No.1, pp.41-52
- Shaw, C.T. (1992), *Using Computational Fluid Dynamics*, Prentice Hall, New York
- Weller, H.G., Tabor G., Jasak, H. and Fureby, C. (1998), "A Tensorial approach to CFD using object orientated techniques", *Computers in Physics*, Vol.12 No.6, pp.620-631
- Wilkes, J.O. (2006), *Fluid Mechanics for Chemical Engineers with Microfluidics and CFD*, 2nd edition, Prentice Hall, New York

#### **Author's Biographical Notes:**

David A. Janes has been a Senior Lecturer in Process Dynamics in the Department of Chemical Engineering at The University of the West Indies since 2009. Previously, he worked in the United Kingdom for CFD-software house Concentration Heat and Momentum Ltd and energy consultancy, KBC Advanced Technology plc., writing commercial engineering software, using CFD and process flow-sheeting programs to solve engineering problem. Dr. Janes also worked as a manufacturing advisory consultant, internationally on oil refining and food processing and on general manufacturing in the UK, where he is a Chartered Engineer. Amongst his research interests are simulation (including CFD and process flow-sheeting) and decision support systems applied to yield and spend management.

■