

SOME OPENFOAM® EXPERIENCES FOR SOLVING CFD PROBLEMS

Norberto M. Nigro^{a,b}, Santiago Márquez Damián^{a,b} and Juan M. Gimenez^b

^a*International Center for Computational Methods in Engineering (CIMEC), INTEC-UNL/CONICET, Güemes 3450, Santa Fe, Argentina, nnigro@intec.unl.edu.ar, <http://www.cimec.org.ar>, santiagomarquezd@gmail.com, <http://www.cimec.org.ar>*

^b*Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral. Ciudad Universitaria. Paraje "El Pozo". Santa Fe. Argentina. jmarcelogimenez@gmail.com, <http://www.fich.unl.edu.ar>*

Keywords: Laminar Solver, Large Eddy Simulation, OpenFOAM, Experimental Validation

Abstract.

During the last years the users involved in the usage of OpenFOAM® software to solve fluid dynamics problems are significantly increasing due to the available tools and solvers on the Web, due to the synergy that is present in several forums and specially due to the fact that OpenFOAM is open source with a GPL license type placing this kind of software in a competitive position against their commercial counterparts. More and more the widespread of CFD is taking place there will be more people interested in such a solution.

In this work we present several tests to understand the maturity of OpenFOAM software as a tool for Computational Fluid Dynamics. Due to our experience in the usage of others commercial and open source CFD software we are worried about a strong validation of OpenFOAM results against their counterpart reference solutions.

Finally some conclusions are presented as guidelines for using OpenFOAM for industrial problems.

1 INTRODUCTION

Benchmarking is a good practice in CFD, even without being an extensive Verification & Validation process (Stern *et al.*, 2006), it allows to set a basis for further calculus and to know the sensitivity of code and model parameters. Fluent[®] and OpenFOAM[®] (Weller *et al.*, 1998) are two well established codes, one on the closed code line and the other one being open to the community under the GNU General Public License.

Related to incompressible isothermal Navier-Stokes solutions, both in laminar and turbulent regimes there are two paradoxical benchmark problems, namely the Three Dimensional Lid-Driven Cavity Flow and the Backward Facing Step. Regarding the first test there have been solution for it from the late seventies (De Vahl Davis and Mallinson, 1976), nevertheless the quality of these results has been disputed because the limited computational resources used for the work (Tang *et al.*, 1995). For the purposes of this work later results will be referenced for the sake clarity and accuracy (Ding *et al.*, 2006; Bouffanais and Deville, 2007). Respect of the second test the foundational work is due to Armaly *et al.* (Armaly *et al.*, 1983). This work refers to a 2D flow and more insight in 3D structures, influence of upstream flow and boundary conditions will be discussed later.

Cited tests have the aim of checking the behavior of codes facing detached flow, specially in the second one. Another test was used for validation, based on the efforts of the European Research Community on Flow, Turbulence and Combustion (ERCOFTAC) (See [ERCOFTAC Classic Collection](#)) to have a reliable database of fluid experiments. The "Duct Flows with Smooth and Rough Walls" test was used to validate results for non-detached flow in turbulent regime, specially the influence of walls and the subgrid viscosity damping.

Turbulence is modeled by means of LES Smagorinsky Model as was proposed by Smagorinsky (Smagorinsky, 1963) and lately modified (Dynamic Smagorinsky Model) by Germano (Germano *et al.*, 1991) and Lilly (Lilly, 1992) and implemented in Fluent[®] following Kim (Kim, 2004) and in OpenFOAM[®] following Weller *et al.* (Weller *et al.*, 1998) and Fureby *et al.* (Fureby *et al.*, 1997).

2 THREE DIMENSIONAL FLOW IN A LID-DRIVEN CAVITY

As the first set of comparatives between Fluent[®] and OpenFOAM[®] lid-driven cavity is modelled in laminar and turbulent regimes. Simulations were carried out in a $60 \times 60 \times 60$ cells grid with refinement towards the wall. Domain extents from 0 to 1 in x , y and z directions, being the origin of coordinates in the inferior, back and left corner (see Figure 1). A fixed velocity of $V_x = 1 \frac{\text{m}}{\text{seg}}$ is applied on the top, so as is well known a big vortex is developed within the cavity. Comparisons were done at x and y centerlines, with coordinates $(x, 0.5, 0.5)$ and $(0.5, y, 0.5)$.

2.1 Laminar case

Laminar case was compared to numerical results given by Ding, Shu, Yeo and Xu (Ding *et al.*, 2006), taking the case of $\text{Re} = 100$. In Fluent[®] the case was set with a pressure based, segregated, steady solver with Green-Gauss Cell Based gradient treatment. SIMPLE algorithm selected for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum. The pressure was discretized with Standard discretisation and Second Order Up-

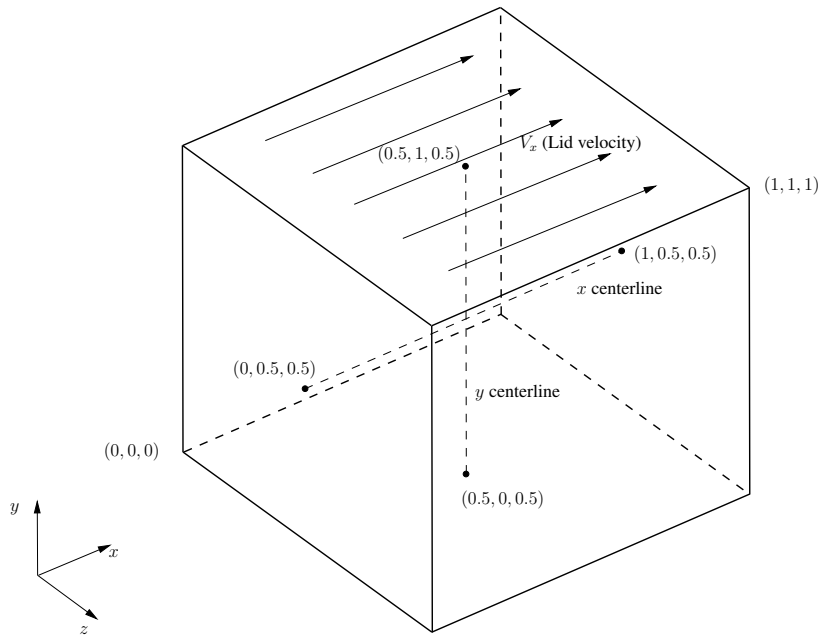


Figure 1: Detail of geometry used for Lid-Driven cavity simulations

wind discretisation for was set for momentum. Finally AMG¹ solver with default settings was used and residuals were reduced below of 1×10^{-5} for all variables. It is important to note that Second Order Upwind discretisation follows the work of Barth and Jespersen (Barth and Jespersen., 1989) where the gradient used for extrapolation from cell center to cell face is limited, so new maxima or minima are introduced.

For OpenFOAM[®] a pressure based, segregated, steady solver (`simpleFoam`) was used with SIMPLE algorithm for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum. Residuals were reduced below of 1×10^{-5} for all variables and Gauss Linear discretisation was set for pressure and divergence terms. Regarding residuals criteria it is possible to show that residual definition in both of used codes are quite similar, so then similar criteria for convergence were set (See Appendix A).

Regarding pressure discretisation both codes have used a Rhie and Chow (Rhie and Chow, 1983) based formulation, this was set by means of Standard Pressure Discretisation in Fluent[®] (See Fluent[®] 6.3.26 Users Guide, chapter 25.4.1) and Gauss Linear discretisation in OpenFOAM[®] (Peng Karrholm, 2008).

With the aim of comparing different strategies for linear system solution and advective terms discretisation particular settings were used, particularly a) Bi-Conjugate Gradient for solving and Full Upwind divergence terms discretisation (CG), b) Geometric Algebraic MultiGrid for solving and Full Upwind advective terms discretisation (GAMG), c) Bi-Conjugate Gradient for solving and Linear Corrected for divergence terms discretisation (CG-2nd.Order).

Results are shown in Figures 2 and 3. From these figures it is possible to conclude that no

¹No other solver is available in Fluent[®]

difference is found in using CG or GAMG in this case in OpenFOAM®, at least for results. Both cases were run with Full Upwind for divergence terms and have approximately 5% of maximum error. For Fluent® satisfactory results were found with initial settings. After the first set of running in OpenFOAM® a last one was done using Linear Corrected discretisation for divergence terms which allowed to obtain better agreement with reported results.

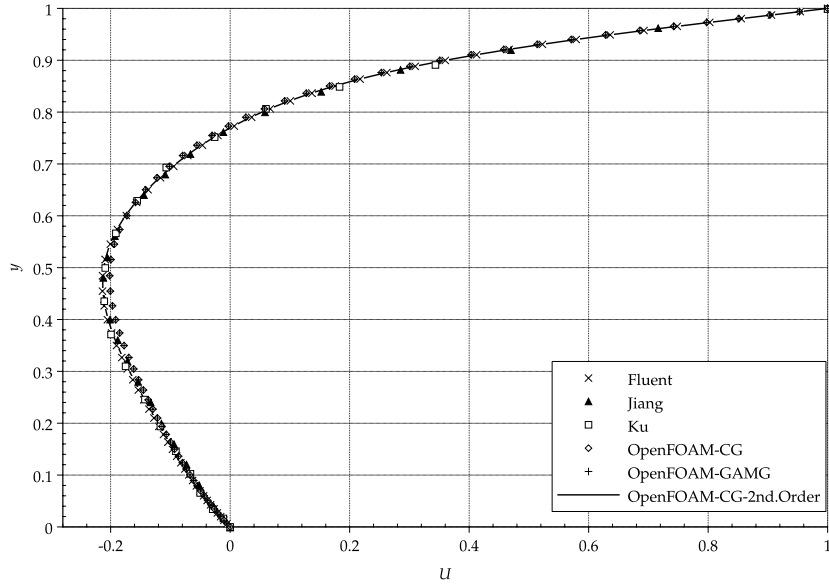


Figure 2: Profile for U velocity in the vertical centerline (y centerline) for laminar case ($Re = 100$).

More differences were found analyzing convergence behavior. Fluent, see Figure 4.a), shows monotone convergence and matches the residuals criteria at about 350 iterations. In OpenFOAM® CG and GAMG shows noisy convergence matching the convergence criteria not so clearly, see Figure 4.b, c), in both cases Full Upwind discretisation was used for divergence terms. Finally Figure 4.d), using CG and Linear Corrected for divergence terms in OpenFOAM® shows excellent convergence at first (almost finished work at 50 approximately iterations), but is not so clear again matching the convergence criteria. These examples show that nevertheless good agreement was obtained in solution there are some aspects of system solving in OpenFOAM® that have to be evaluated more deeply (See [CFD Online simpleFoam Convergence Problems](#) thread).

2.2 Theoretical background on Large Eddy Simulation (LES)

2.2.1 Fluent®

Filtered Navier-Stokes Equations Large Eddy Simulation model relies on magnitude filtering to avoid complete solving of Navier-Stokes Equations. In this process variables are filtered spatially so that big eddies are calculated and smaller ones are modelled. In the Finite Volume Method framework and particularly in Fluent® the applied filter is given by Equation 1

$$\bar{\phi}(\mathbf{x}) = \frac{1}{V} \int_{\mathcal{V}} \phi(\mathbf{x}') d\mathbf{x}', \quad \mathbf{x}' \in \mathcal{V} \quad (1)$$

where V is the volume of a computational cell. Taking into account a general filtering process

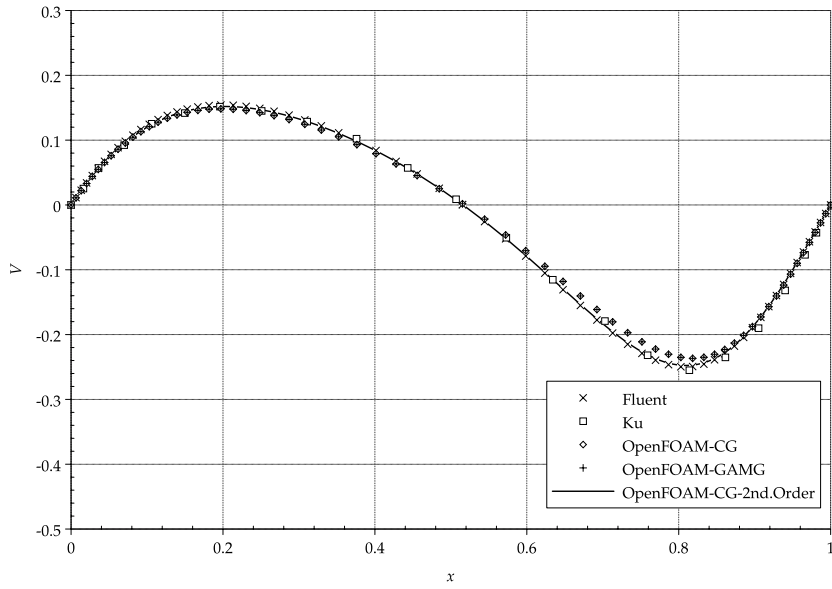


Figure 3: Profile for V velocity in the horizontal centerline (x centerline) for laminar case ($\mathbf{Re} = 100$).

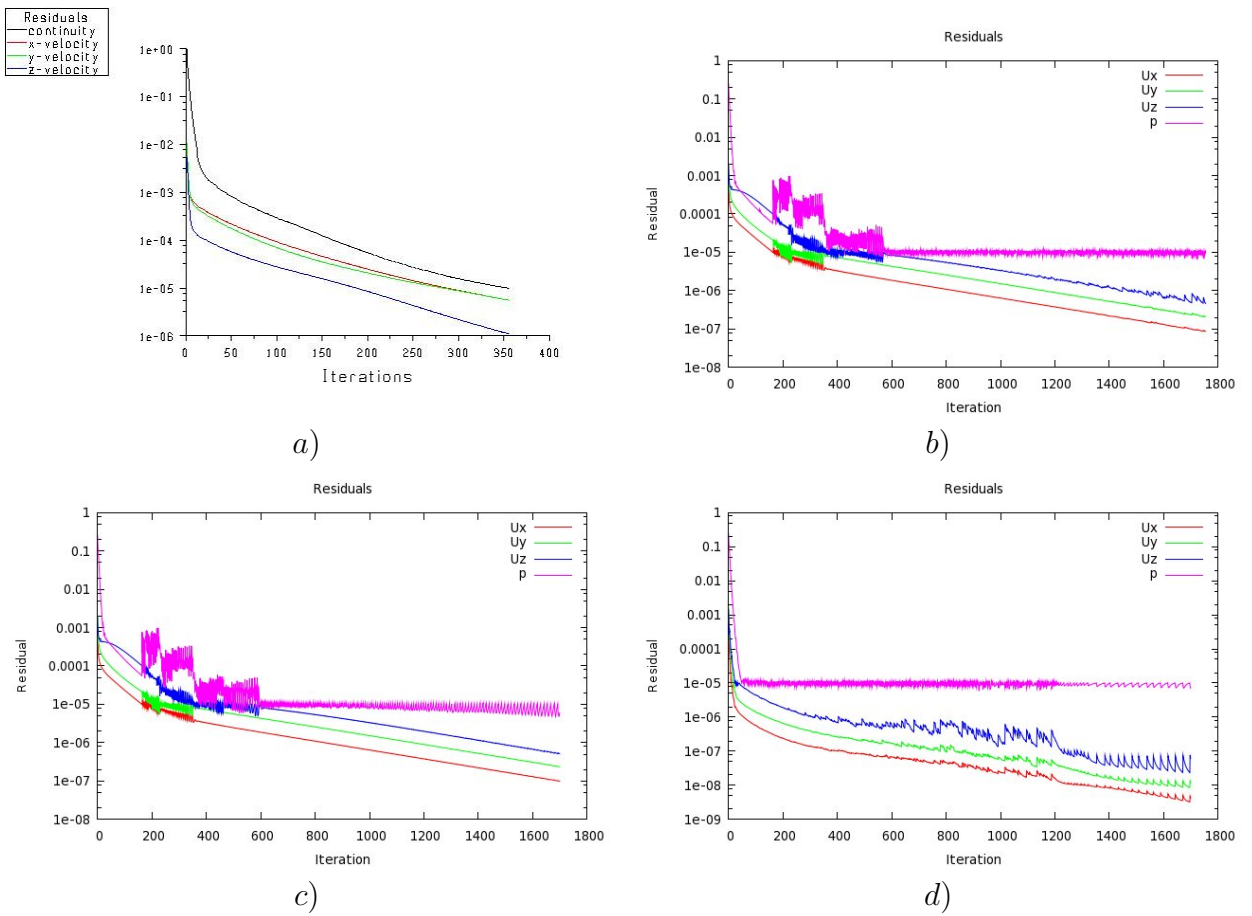


Figure 4: Convergence evolution a) for Fluent, b) for OpenFOAM[®] with CG and Full Upwind, c) OpenFOAM[®] GAMG and Full Upwind and d) OpenFOAM[®] with CG and Linear Corrected

like in Equation 2

$$\bar{\phi}(\mathbf{x}) = \int_{\mathcal{D}} \phi(\mathbf{x}') G(\mathbf{x}, \mathbf{x}') d\mathbf{x}' \quad (2)$$

where \mathcal{D} is the fluid domain, and G is the filter function that determines the scale of the resolved eddies. Here the filter function, $G(\mathbf{x}, \mathbf{x}')$ involved is given by Equation 3

$$G(\mathbf{x}, \mathbf{x}') = \begin{cases} 1/V, & \mathbf{x}' \in \mathcal{V} \\ 0, & \mathbf{x}' \text{ otherwise} \end{cases} \quad (3)$$

Finally filtered incompressibility (Equation 4) condition and Navier-Stokes (Equation 5) equations reads

$$\frac{\partial}{\partial x_i}(\rho \bar{u}_i) = 0 \quad (4)$$

and

$$\frac{\partial}{\partial t}(\rho \bar{u}_i) + \frac{\partial}{\partial x_j}(\rho \bar{u}_i \bar{u}_j) = \frac{\partial}{\partial x_j} \left(\mu \frac{\partial \sigma_{ij}}{\partial x_j} \right) - \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} \quad (5)$$

where σ_{ij} is the stress tensor due to molecular viscosity defined by

$$\sigma_{ij} \equiv \left[\mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right] - \frac{2}{3} \mu \frac{\partial \bar{u}_l}{\partial x_l} \delta_{ij} \quad (6)$$

and τ_{ij} is the subgrid-scale stress defined by

$$\tau_{ij} \equiv \rho \overline{u_i u_j} - \rho \bar{u}_i \bar{u}_j \quad (7)$$

Subgrid-Scale Models The subgrid-scale stresses resulting from the filtering operation (Equation 7) are unknown and require modeling. Due small eddies tends to be more isotropic than bigger ones it is possible to use simple methods, like RANS², to parametrize them. This method is applied in most of Subgrid-Scale (SGS) models (Germano et al., 1991) and are based on the Boussinesq hypothesis as in the RANS models. So that, subgrid-scale turbulent stresses are computed from Equation 8

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2\mu_t \bar{S}_{ij} \quad (8)$$

where μ_t is the subgrid-scale turbulent viscosity. The isotropic part of the subgrid-scale stresses τ_{kk} is not modeled, but added to the filtered static pressure term. \bar{S}_{ij} is the rate-of-strain tensor for the resolved scale defined by Equation 9

$$\bar{S}_{ij} \equiv \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (9)$$

²Reynolds-Averaged Navier Stokes

Smagorinsky-Lilly Model One of the most used models for the eddy viscosity μ_t in Equation 8 is that was given by Smagorinsky and improved by Lilly. In this model eddy-viscosity is modeled by Equation 10

$$\mu_t = \rho L_s^2 |\overline{S}| \quad (10)$$

where L_s is the mixing length for subgrid scales and $|\overline{S}| \equiv \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$. In Fluent[®], L_s is computed using

$$L_s = \min(\kappa d, C_s V^{1/3}) \quad (11)$$

where κ is the von Karman constant, d is the distance to the closest wall, C_s is the Smagorinsky constant, and V is the volume of the computational cell.

Since C_s must be tuned properly for each case, it became a serious shortcoming for this simple model. Piomelli *et al.* [See (Germano *et al.*, 1991)] found the optimum value of C_s to be around 0.1, it stands for a wide range of flows, and is the default value in Fluent^{®3}.

2.2.2 OpenFOAM[®]

LES Smagorinsky-Lilly formulation in OpenFOAM[®] is in general similar to Fluent, nevertheless there are some important differences to take in account at running time. For μ_t or μ_{SGS} as is referred in OpenFOAM[®] documents, its definition is given by:

$$\mu_{SGS} = \rho L_s^2 C_S^2 |\overline{S}| \quad (12)$$

like in Fluent[®] (See Eq. 10). L_s stands for

$$L_s = \min \left[\frac{\kappa y}{C_\Delta} \left(1 - e^{-\frac{y^+}{A^+}} \right), V^{1/3} \right] \quad (13)$$

where κ is the von Karman constant, y is the distance to the closest wall, C_Δ and A^+ are a model constants ($C_\Delta = 0.158$ and $A^+ = 26$ by default in OpenFOAM[®]), V is the volume of the computational cell. This approach is in the spirit of Van Driest (Van Driest, 1956) damping function for μ_{SGS} (De Villiers, 2006)

Working with the transport equation for the SGS and putting shear production equal to the dissipation (See CFD Online OpenFOAM[®] LES thread) is possible to arrive to this relationship

$$C_S = \sqrt{C_K \sqrt{\frac{C_K}{C_\epsilon}}} \quad (14)$$

Since $C_K = 0.07$ and $C_\epsilon = 1.05$ are the defaults for OpenFOAM[®] C_S has a value of 0.13.

³See Fluent[®] 6.3.26 manual chapter 12 for additional guidelines in C_s selection.

2.3 Dynamic Smagorinsky-Lilly Model

Both in Fluent[®] and OpenFOAM[®] Dynamic Smagorinsky-Lilly Model is based on the work of Germano (Germano et al., 1991) and Lilly (Lilly, 1992). In such a dynamic model C_S is evaluated dynamically, avoiding the necessity of constant tuning. In Fluent[®] particularly this job is done using Kim's implementation (Kim, 2004) for non-structured grids and a clipping criteria is applied to C_S (See Equation 15)

$$0 \leq C_S \leq 0.23 \quad (15)$$

In OpenFOAM[®], as in all cases, formulation can be extracted directly from the code⁴. The version present in code is explained and compared with other dynamic models by Fureby (Fureby et al., 1997). So, basic relations are given by Equations 16-19

$$\mathbf{B} = \frac{2}{3}k\mathbf{I} - 2\nu_{SGS} \text{dev}(\mathbf{B}) \quad (16)$$

$$\mathbf{B} = \frac{2}{3}k\mathbf{I} - 2\nu_{eff} \text{dev}(\mathbf{B}) \quad (17)$$

$$k = C_I \Delta^2 \|D\|^2 \quad (18)$$

$$\nu_{SGS} = C_D \Delta^2 \|D\|^2 \quad (19)$$

where $\text{dev}(\mathbf{D}) = \mathbf{D} - \frac{1}{3}[\text{tr}(\mathbf{D})]\mathbf{I}$, $\text{tr}(\mathbf{D}) = D_{11} + D_{22} + D_{33}$, $\nu_{eff} = \nu_{SGS} + \nu$ and Δ is a function of cell size. Constants are defined by Equations 20-21.

$$C_I = \frac{\langle K m \rangle}{\langle m m \rangle} \quad (20)$$

$$C_D = \frac{\langle L \cdot M \rangle}{\langle M M \rangle} \quad (21)$$

where $K = \frac{1}{2}(\overline{U\overline{U}} - \overline{U}\overline{U})$, $m = \Delta^2(4\|\overline{\mathbf{D}}\|^2 - \|\overline{\mathbf{D}}\|^2)$, $L = \text{dev}(\overline{U\overline{U}} - \overline{U}\overline{U})$ and $M = \Delta^2(\|\overline{\mathbf{D}}\|\overline{\mathbf{D}} - 4\|\overline{\mathbf{D}}\|)$.

2.4 Turbulent case

Turbulent case was carried out by means of Dynamic Smagorinsky Method (DSM) implemented both in Fluent[®] and OpenFOAM[®]. Results were compared with those given by Bouffanais and Deville (Bouffanais and Deville, 2007). Results are give at $\text{Re} = 12000$ from experiments and DSM implemented by Spectral Element Methods.

Settings for Fluent[®] were as follows: pressure based, segregated, unsteady, second order implicit solver with Green-Gauss Cell Based gradient treatment. SIMPLE algorithm for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum. Standard pressure discretisation and Bounded Central Difference for momentum. Residuals were reduced

⁴See `~/OpenFOAM/OpenFOAM-<version>/src/turbulenceModels/incompressibleLES/dynSmagorinsky/dynSmagorinsky.C`

below of 1×10^{-3} for all variables. For OpenFOAM[®]: pressure based, segregated, unsteady solver (`icoFoam`). PISO algorithm for pressure-velocity coupling. Residuals were reduced below of 1×10^{-5} for all variables (included the turbulent ones) except for p where residuals gone below of 1×10^{-6} . Gauss Linear discretisation for pressure and divergence terms. Euler scheme (first order implicit) was used for time discretisation.

In Figures 5 and 6 results for Fluent[®] and OpenFOAM[®] are shown. For U velocity OpenFOAM[®] appear to be more accurate than Fluent[®] but even though both softwares aren't in good agreement with experimental and DSM reported data. For V velocity presents good agreement with reported DSM data but OpenFOAM[®] exhibits a behavior similar to experiments in some of the point but not in the near-wall zones, so not concluding remarks can be done.

It's important to note that results shown for DSM are averaged results in fully developed regime.

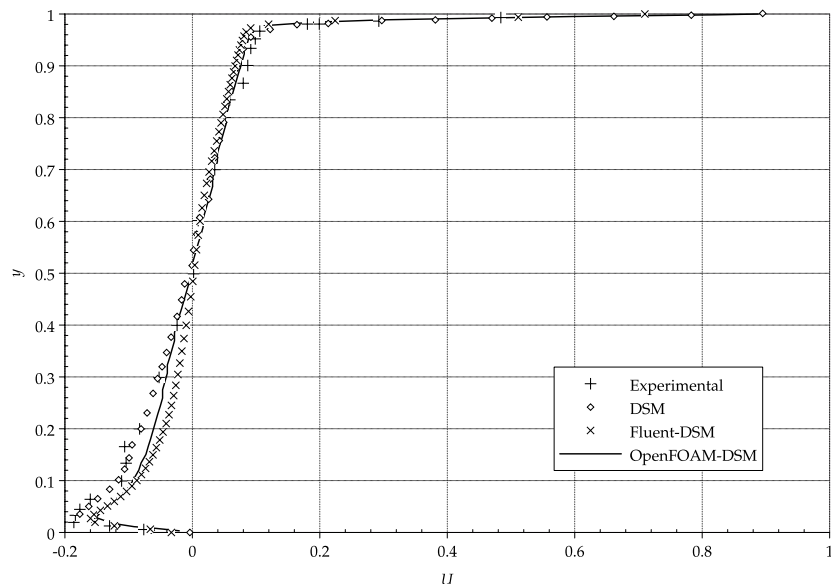


Figure 5: Profile for U velocity in the vertical centerline (y centerline) for turbulent case ($Re = 12000$).

3 LAMINAR FLOW IN A BACKWARD FACING-STEP

The second test that was carried out was the Backward Facing-Step. It was modelled in laminar regime. This flow allows to compare prediction of separated flow like is developed along the step in geometry. This is a classical test and was proposed by Armaly *et al.* (Armaly *et al.*, 1983). Interest in separated flows is based on taking this benchmark as a next step in CFD code characterization because the presence of recirculation, adverse pressure gradients, etc.

Laminar case was compared to experimental results from Armaly given by Chiang and Sheu (Chiang and Sheu, 1999), taking the case of $Re = 389$. Simulation was carried out in 2D and geometry was inspired in used by Chiang (see Figure 7)

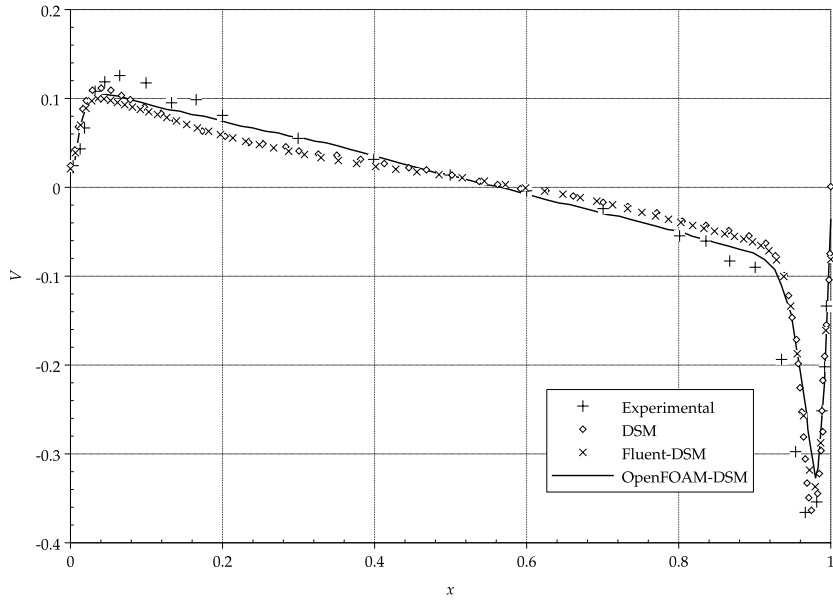


Figure 6: Profile for V velocity in the horizontal centerline (x centerline) for turbulent case ($Re = 12000$).

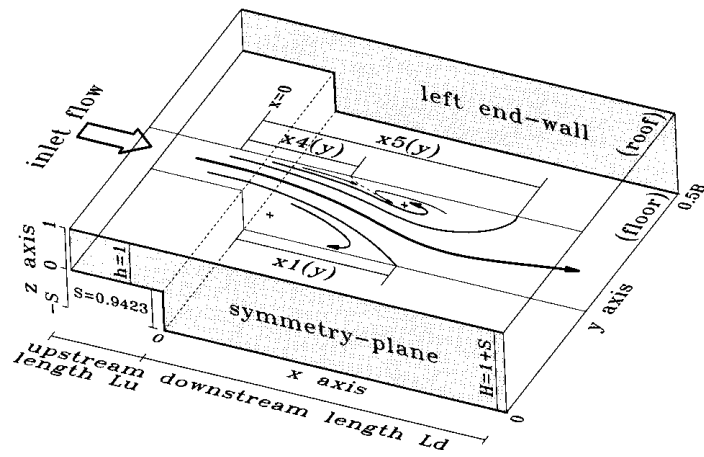


Figure 7: Backward-facing step channel geometry, from (Chiang and Sheu, 1999)

Dimensions h , S and $L_d = 55h$ were taken from Chiang, upstream length $L_u = h$ was selected following guidelines given by Williams (Williams and Baker, 1997). Domain was meshed with a regular grid of 30 cells in z direction in inlet zone, and 60 cells in z direction in expansion. In x direction 147 cells were used in expansion with Last/First cell ratio of 26.66, first cell longitude was about of 0.033 units and last one about of 0.88 units. In inlet zone 3 cells were used in x direction with a Last/First ratio of 10.

In Fluent® the case was set follows: pressure based, segregated, steady solver with Green-Gauss Cell Based gradient treatment. SIMPLE algorithm for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum. Standard pressure discretisation and Second Order Upwind/QUICK discretisation for momentum. Residuals were reduced below of 1×10^{-5} for all variables. For OpenFOAM® these were the general settings: pressure based, segregated, steady solver (simpleFoam). SIMPLE algorithm for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum. Residuals were reduced below of 1×10^{-5} for all variables. Gauss Linear discretisation for pressure and Linear Corrected/QUICK schemes for advective terms discretisation.

Results are shown in Figures 8 and 9.

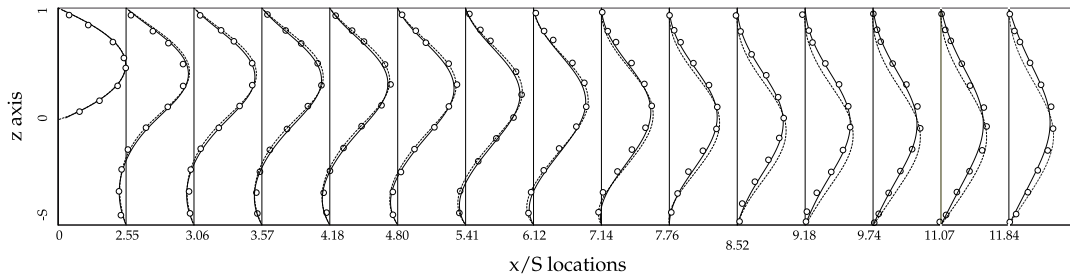


Figure 8: Streamwise velocity profiles for ($Re = 389$) with Second Order Upwind/Linear Corrected Scheme. Circles: Armaly results, dashed line: Fluent, continuous line: OpenFOAM

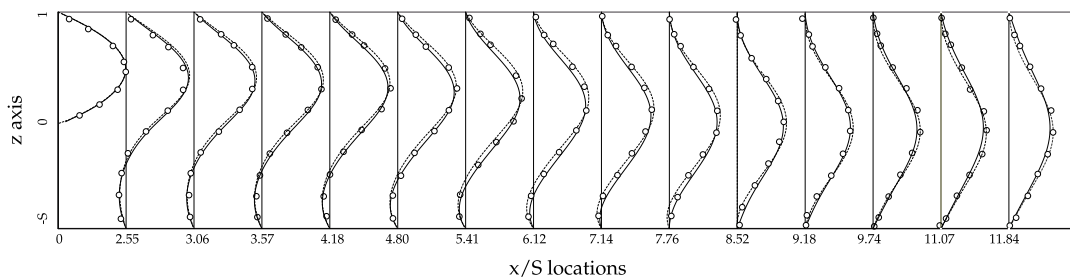


Figure 9: Streamwise velocity profiles for ($Re = 389$) with QUICK Scheme. Circles: Armaly results, dashed line: Fluent, continuous line: OpenFOAM

4 TURBULENT FLOW IN 3D SQUARE CHANNEL

4.1 Introduction

Finally a comparison between Fluent® and OpenFOAM® regarding Large Eddy Simulation Model in its Smagorinsky-Lilly implementation is presented. To do this, an ERCOFTAC Database example (See [ERCOFTAC Database Case 52. Duct Flows with Smooth and Rough Walls](#)) with experimental data is taken as a reference, looking for equal results in Fluent® and OpenFOAM® and fairly good agreement with experiments. In this case classical Smagorinsky-Lilly implementation is used due the simplicity to match models between both codes.

Problem consists in a square duct with a cross-section of $h = 0.05$ m and $L = 4$ m in length. Simulations were carried out at $Re = 6.5 \times 10^4$, so $\nu = 0.769 \times 10^{-6} \frac{m^2}{sec}$ and velocity at inlet was $V = 1 \frac{m}{sec}$ (See Figure 10).

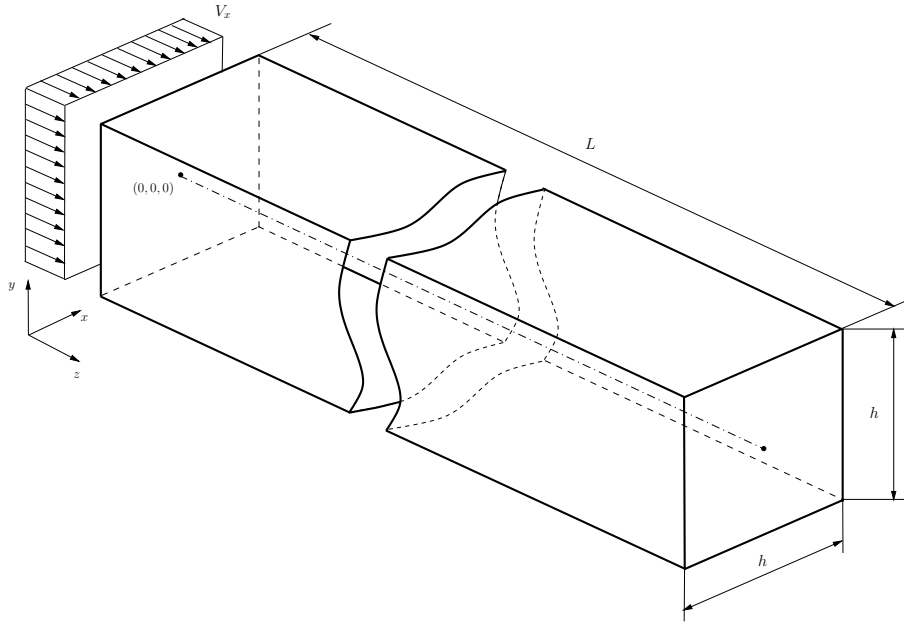


Figure 10: Detail of geometry used for ERCOFTAC 52 simulation

4.2 Theoretical background on LES Near-Wall Treatment

4.2.1 Fluent®

When the mesh is fine enough to resolve the laminar sublayer, the wall shear stress is obtained from the laminar stress-strain relationship:

$$\frac{\bar{u}}{u_\tau} = \frac{\rho u_\tau y}{\mu} \quad (22)$$

If the mesh is too coarse to resolve the laminar sublayer, it is assumed that the centroid of the wall-adjacent cell falls within the logarithmic region of the boundary layer, and the law-of-the-wall is employed:

$$\frac{\bar{u}}{u_\tau} = \frac{1}{\kappa} \ln E \left(\frac{\rho u_\tau y}{\mu} \right) \quad (23)$$

where κ is the von Kármán constant and $E = 9.793$. If the mesh is such that the first near wall point is within the buffer region, then two above laws are blended [as follows]

$$u^+ = e^\Gamma u_{\text{lam}}^+ + e^{\frac{1}{\Gamma}} u_{\text{turb}}^+ \quad (24)$$

where the blending function is given by:

$$\Gamma = -\frac{a(y^+)^4}{1 + by^+} \quad (25)$$

where $a = 0.01$, $b = 5$, $y^+ = y u_\tau / \nu$ and $u^+ = u / u_\tau$. Similarly, the general equation for the derivative $\frac{du^+}{dy^+}$ is

$$\frac{du^+}{dy^+} = e^\Gamma \frac{du_{\text{lam}}^+}{dy^+} + e^{\frac{1}{\Gamma}} \frac{du_{\text{turb}}^+}{dy^+} \quad (26)$$

This approach allows the fully turbulent law to be easily modified and extended to take into account other effects such as pressure gradients or variable properties. This formula also guarantees the correct asymptotic behavior for large and small values of y^+ and reasonable representation of velocity profiles in the cases where y^+ falls inside the wall buffer region ($3 < y^+ < 10$).

4.2.2 OpenFOAM®

Like Fluent®, blending function is provided in OpenFOAM® in order to manage different values of y^+ for first grid cell and its influence in near-wall function selection. So blending function is given by 'Spalding Law' (De Villiers, 2006) in the form

$$y^+ = u^+ \frac{1}{E} \left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{1}{2} (\kappa u^+)^2 - \frac{1}{6} (\kappa u^+)^3 \right] \quad (27)$$

where $\kappa = 0.4187$ and $E = 9$ as defaults in OpenFOAM®.

4.3 Results

Simulations were made on an hexahedral mesh of $120 \times 30 \times 30$ elements in $x \times y \times z$ directions. Double sided grading with First/Last ratio of 26 was used in y and z directions, refining toward the walls. In x First/Last ratio of 100 was used. Running settings for Fluent® were as follows: pressure based, segregated, unsteady, second order implicit solver with Green-Gauss Cell Based gradient treatment, SIMPLE algorithm for pressure-velocity coupling with relaxation factors of 0.3 for pressure and 0.7 for momentum, standard pressure discretisation and Bounded Central Difference for momentum [this is implemented based on a NVD limiter, particularly following Leonard (Leonard, 1991)]. Residuals were reduced below of 1×10^{-5} for all variables. In OpenFOAM® the model was set with a pressure based, segregated, unsteady solver (oodles), PISO algorithm for pressure-velocity coupling. Residuals were reduced below of 1×10^{-5} for all variables (included the turbulent ones) except for p where residuals gone below of 1×10^{-6} . Gauss Linear discretisation for pressure and general divergence terms and SFCD (Second order bounded) for $\frac{\partial}{\partial x_j} (\rho \bar{u}_i \bar{u}_j)$ term. Finally Backward scheme (second order implicit) was used for time discretisation.

Results from Fluent[®] were obtained following the theoretical background given previously with $C_S = 0.1$. In the other hand results in OpenFOAM[®] were obtained progressively changing free parameters sequentially as in indicated in other to mimic Fluent[®] results. First of all, OpenFOAM[®] case is run with default settings (Case 1). Then C_S is matched (Case 2) via Equation 14. As the third step changes are made in order to equalize ν_{SGS} damping towards the wall (Cases 3 and 4) by means of L_S calculation (Equation 13, compare with Equation 11). Finally, blending of wall functions is activated (Case 5).

1. Original parameters from OpenFOAM[®]
2. C_ϵ is changed from 1.05 to 3.43 ($C_S = 0.1$)
3. C_Δ is changed from 0.158 to 0.1. This allow to partially equalize OpenFOAM[®] and Fluent[®] ν_{SGS} damping functions.
4. A^+ is changed from 26 to 0.8 equalizing ν_{SGS} damping functions.
5. Finally `nuSgsWallFunction` option is activated forcing OpenFOAM[®] to use blended wall functions in near wall zones as in expressed in Eq. 27.

E constant in 'Spalding Law' wasn't change in OpenFOAM[®] because it has second order effect in solution. Results of this sequence are shown in Figure 11. Isolating the data referred to Fluent[®] and OpenFOAM[®] final comparison Figure 12 is obtained.

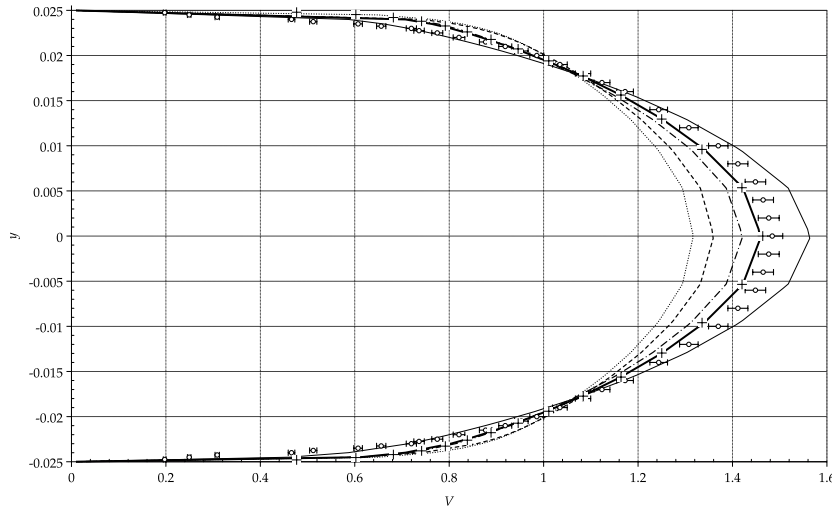


Figure 11: Sequence of solutions in OpenFOAM[®], solution in Fluent[®] and experimental results. Circles: experimental data by ERCOFTAC, error bars $\pm 1.5\%$; Crosses: Fluent; Dotted line: OpenFOAM[®] 1; Dashed line: OpenFOAM[®] 2; Dash-dotted line: OpenFOAM[®] 3; Thin continuous line: OpenFOAM[®] 4; Thick continuous line: OpenFOAM[®] 5.

Finally in Figure 13 results as is Figure 12 are shown but with comparing cell center values. Note that interpolation used before tends to smooth the solution, then to compare in a complete sense is necessary to use both figures.

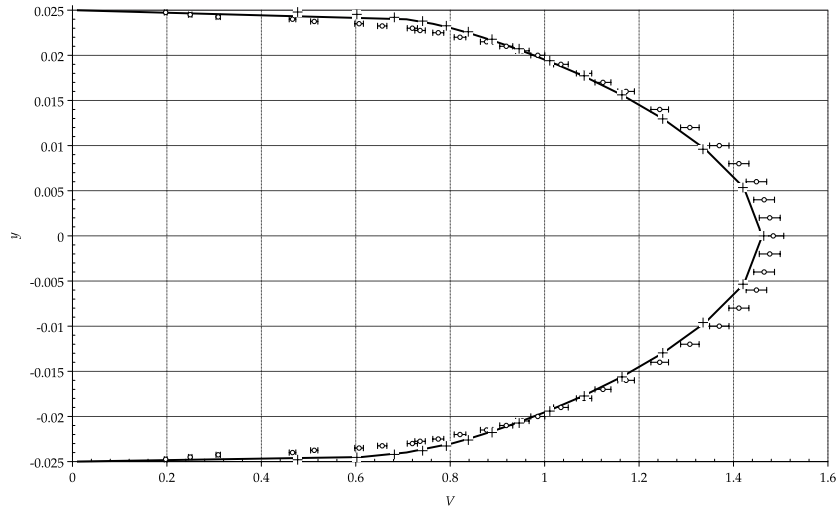


Figure 12: Comparison between Fluent[®] and OpenFOAM[®] (experimental reference include). Circles: experimental data by ERCOFTAC, error bars $\pm 1.5\%$; Crosses: Fluent[®]; Continuous line: OpenFOAM[®] final.

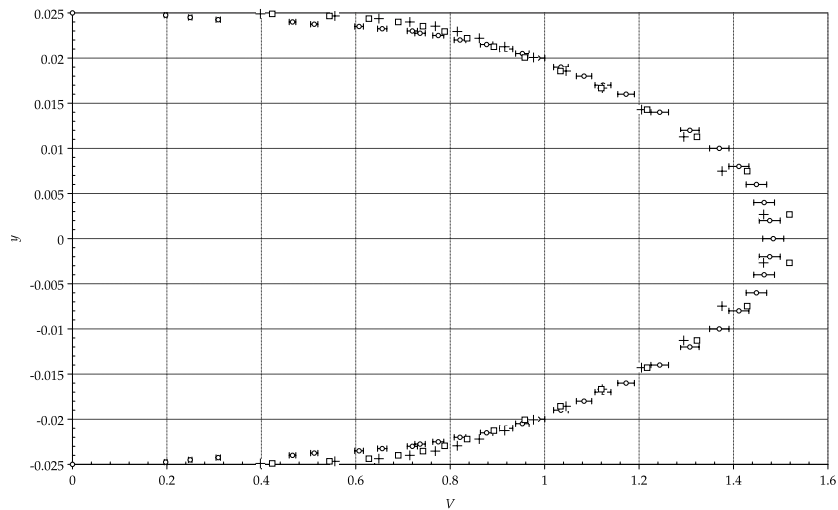


Figure 13: Comparison between Fluent[®] and OpenFOAM[®] (experimental reference include). Circles: experimental data by ERCOFTAC, error bars $\pm 1.5\%$; Crosses: Fluent[®]; Squares: OpenFOAM[®] final.

5 CONCLUSIONS

After tests OpenFOAM® appears as a reliable tool for CFD. Laminar solver has no particular details in its implementation and use. It gives close results to Fluent® ones and contrasting with experimental data, both in detached and non-detached flux. Usual care must be taken into account as mesh refinement, etc. Particularly in advective schemes QUICK results to be the best for Fluent® simulation and Linear Corrected in OpenFOAM® case.

Regarding turbulent simulations, Dynamic Smagorinsky model implementations give similar results at same level of convergence, as was shown in Lid-Driven Cavity test. Convergence criteria based on scaled residuals has been shown as similar in both codes, but attention must be given to noisy residual evolution in pressure (continuity) equation residuals in OpenFOAM®. In classical Smagorinsky-Lilly model necessity of model equalization have been proved. Such equalization is obtained via model parameters adjustment in order to obtain the same Smagorinsky constant and mixing length approximation.

Attention must be given to wall treatment between both codes. In wall dominated flows, wall effects not only are taken into account by ν damping functions but also in near wall treatment models. These models allow to manage different values in mesh y^+ avoiding use of extremely fine meshes. In this case only activation of ν near wall treatment model in OpenFOAM® was necessary to finally match Fluent® and experimental results, but more deep adjustments can be made via case dictionaries.

REFERENCES

- Armaly B., Durst F., Pereira J., and Schonung B. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.*, 127:473–496, 1983.
- Barth T.J. and Jespersen. D. The design and application of upwind schemes on unstructured meshes. In *Technical Report AIAA-89-0366, AIAA 27th Aerospace Sciences Meeting*. 1989.
- Bouffanais R. and Deville M.O. Large-eddy simulation of the flow in a lid-driven cubical cavity. *Physics of Fluids*, 19, 2007.
- Chiang T. and Sheu T. A numerical revisit of backward-facing step flow problem. *Physics of Fluids*, 11(4):862–874, 1999.
- De Vahl Davis G. and Mallinson G.D. An evaluation of upwind and central difference approximation by a steady recirculating flow. *Comput. Fluids*, 4:29–43, 1976.
- De Villiers E. *The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows*. Ph.D. thesis, Imperial College, London, 2006.
- Ding H., Shu C., Yeo K., and D. X. Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadratic differential quadrature method. *Comput. Methods Appl. Mech. Engrg.*, 195:516–533, 2006.
- Ferziger J. and Peric M. *Computational Methods for Fluid Dynamics*, volume I. Springer-Verlag, 2002.
- Fureby C., Tabor G., Weller H.G., and Gosman A.D. A comparative study of subgrid scale models in homogeneous isotropic turbulence. *Physics of Fluids*, 9(5):1416–1429, 1997.
- Germano M., Piomelli U., P. M., and H C.W. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids*, 3(7):1760–1765, 1991.
- Kim S.E. Large eddy simulation using unstructured meshes and dynamic subgrid-scale turbulence models. In *Technical Report AIAA-2004-2548, American Institute of Aeronautics and Astronautics, 34th Fluid Dynamics Conference and Exhibit*. 2004.
- Leonard B.P. The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Comput. Methods Appl. Mech. Engrg.*, 88:17–74, 1991.
- Lilly D.K. A proposed modification of the germano subgrid-scale closure model. *Physics of Fluids*, 4:633–635, 1992.
- Peng Karrholm F. *Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion*. Ph.D. thesis, Chalmers University of Technology, Goteborg, 2008.
- Rhie C.M. and Chow W.L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, 1983.
- Smagorinsky J. General circulation experiments with the primitive equations. 1. the basic experiment. *Mon. Weather Rev.*, 91:99–164, 1963.
- Stern F., Wilson R., and Shao J. Quantitative v&v of cfd simulations and certification of cfd codes. *Int. J. Numer. Meth. Fluids*, 50:1335–1355, 2006.
- Tang L.Q., Cheng T., and Tsang T. Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method. *Int. J. Numer. Meth. Fluids*, 21:413–432, 1995.
- Van Driest E.R. On turbulent flow near a wall. *Journal of the Aeronautical Sciences*, 23(11):1007–1011, 1956.
- Weller H.G., Tabor G., Jasak H., and Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computer in Physics*, 12(6):620–631, 1998.
- Williams P. and Baker A. Numerical simulations of laminar flow over a 3d backward-facing

step. *Int. J. Numer. Meth. Fluids*, 24:1159–1183, 1997.

A RESIDUALS DEFINITION FOR FLUENT® AND OPENFOAM® PRESSURE-BASED SOLVERS

Judging convergence by residuals is usual in CFD code utilization. Problems arise when is necessary to compare codes in such parameter. Following is a description of definition for residuals in Fluent® and OpenFOAM® pressure based solvers

A.1 Fluent

A.1.1 Theoretical background

By means of Finite Volume Method and after discretisation the conservation equation for a general variable, ϕ at a cell P can be written as⁵ in Equation 28

$$a_P \phi_P = \sum_N a_N \phi_N + b \quad (28)$$

Here a_P is the center coefficient, i.e. the contribution of all terms that involves the unknown at the cell center, a_N are the influence coefficients for the neighboring cells, namely the cells that share a face with the analyzed cell, and b is the contribution of the constant part of the source term S_c in $S = S_c + S_P \phi$ and of the boundary conditions.

The residual R^ϕ as is usually defined or non scaled residual in Fluent®'s nomenclature, is the imbalance in Equation 28 summed over all the computational cells as Equation 29 expresses.

$$R^\phi = \sum_{\text{cells } P} \left| \sum_N a_N \phi_N + b - a_P \phi_P \right| \quad (29)$$

In order to adimensionalize and to refer the residual to a similar basis, residuals are scaled. Fluent® scales the residual using a scaling factor representative of the flow rate of ϕ through the domain. This *scaled* residual is defined as in Equation 30

$$R^\phi = \frac{\sum_{\text{cells } P} \left| \sum_N a_N \phi_N + b - a_P \phi_P \right|}{\sum_{\text{cells } P} |a_P \phi_P|} \quad (30)$$

For the momentum equations the denominator term $a_P \phi_P$ is replaced by $a_P v_P$, where v_P is the magnitude of the velocity at cell P .

Analyzing Equation 30 it is possible to see that the imbalance (numerator) goes to zero along iterations, meanwhile denominator converges to a constant value, giving a reduction of overall residuals.

A.2 OpenFOAM

A.2.1 Theoretical background

OpenFOAM® residuals definition lies on scaled residuals theory too, nevertheless different scaling factor is used, an explanation was given by Jasak as follows (See [CFD Online OpenFOAM® Convergence on Segregated Solvers](#) thread). For a system:

$$\mathbf{A} x = b \quad (31)$$

⁵See Fluent® 6.3.26 Users Guide, chapter 25.18.1

residual is defined as

$$R = b - \mathbf{A}x \quad (32)$$

Then residual scaling is applied with the following normalization factor procedure:

$$x_{\text{Ref}} = \bar{x} \quad (33)$$

setting temporal variables

$$\begin{aligned} w_{\mathbf{A}} &= \mathbf{A}x \\ p_{\mathbf{A}} &= \mathbf{A}x_{\text{Ref}} \end{aligned} \quad (34)$$

now the scaling factor is:

$$\text{scaleFactor} = \sum |w_{\mathbf{A}} - p_{\mathbf{A}}| + |b - p_{\mathbf{A}}| + \text{matrix.small_} \quad (35)$$

where $\text{matrix.small_} = 1.0 \times 10^{-20}$. Then the scaled residual is:

$$R_S = \frac{\sum |b - w_{\mathbf{A}}|}{\text{scaleFactor}} \quad (36)$$

Again as in Fluent[®] denominator goes to a constant due difference to solution and average field value goes to constant too. In the other hand numerator goes to zero if solution converges, then residuals should go to zero as a evidence of convergence.

A.3 Comparisons and recommendations

As was shown scaled residuals definition for Fluent[®] and OpenFOAM[®] are similar, differences were found in scaling factor. Near convergence both numbers must be similar except for a multiplying constant.

Even though explained criteria is useful in most of cases, warning given Ferziger and Peric (Ferziger and Peric, 2002) must be taken into account: "A compromise is to use the reduction of the residual as a stopping criterion. Iteration is stopped when the residual norm has been reduced to some fraction of its original size (usually by three or four orders of magnitude). As we have shown, the iteration error is related to the residual via Equation 29 so reduction of the residual is accompanied by reduction of the iteration error. If the iteration is started from zero initial values, than the initial error is equal to the solution itself. When the residual level has fallen say three to four orders of magnitude below the initial level, the error is likely to have fallen by a comparable amount, i.e. it is of the order of 0.1% of the solution. The residual and the error usually do not fall in the same way at the beginning of iteration process; *caution is also needed because, if the matrix is poorly conditioned, the error may be large even when the residual is small*".