H14-168

ATMOSPHERIC WIND FIELD SIMULATIONS OVER COMPLEX TERRAIN USING PARTIALLY CONVERGED CFD CALCULATIONS – APPLICATION TO MODELLING OF ATMOSPHERIC DISPERSION FOR OPERATIONAL PURPOSES

Radi Sadek[1], Lionel Soulhac[1], Fabien Brocheton[2] and Emmanuel Buisson[2]

[1]Laboratoire de Mécanique des Fluides et d'Acoustique, Université de Lyon CNRS, Ecole Centrale de Lyon, INSA Lyon, Université Claude Bernard Lyon 1, 36 avenue Guy de Collongue, 69134 Ecully, France
[2]NUMTECH, Aubière, Lyon

Abstract: In modeling atmospheric dispersion over complex terrain (relief, roughness or heat flux change), Computational Fluid Dynamics (CFD) codes can be a powerful tool for simulating air flow with very high spatial resolution. However, the well-known drawback of the CFD approach is that it is time consuming. In this paper, a method which uses partially converged CFD solutions as a way of reducing CPU time, while keeping the precision of the solution at an acceptable level, is presented. We therefore demonstrate that it is possible to reach a wind field solution very close to the converged solution, in a small fraction of the CPU time needed to reach the fully converged solution. We present an optimum point of convergence,depending on the complexity of the terrain, for several cases of simulations with the commercial CFD code Fluent. Such complexities include steepness of hills and valleys, roughness of terrain and thermal stratification. We present an estimate of the error in comparison to the fully converged solution and evaluate the gain of CPU time following each case study.Finally, we strengthen our conclusions by a comparison with wind tunnel experiments in the presence of hills.

Key words: *Partially converged CFD, complex terrain, operational dispersion modelling*

INTRODUCTION

There is an increasing need for simulations of atmospheric flows in the presence of complex terrain (valleys and steep hills, rough terrain and presence of heat flux change) for proper monitoring of atmospheric pollutant dispersion in such regions. These simulations require the use ofa high horizontal resolution (10-100m) for proper modelling of the mean and turbulent flow and the capturing of local effect occurring in regions such as recirculation zones, thermal effects, etc. The so-called analytical models (FLOWSTAR, WASP …), generally give adequate results in simulations with high horizontal resolution along with a negligible CPU time. However, these models present a lack of precision in simulations where the terrain is too complex, and are unable to capture the major local effects occurring, as shown by Sadek*et al*. (2011). Computational Fluid Dynamics (CFD) codes have therefore emerged as a powerful toolfor simulating air flow with very high spatial resolution. However, the well-known drawback of the CFD approach is that it is time consuming, and consequently unsuitable for operational purposes. In this paper, we investigate the possibility of the use of partially convergedCFD calculations as a way of reducing the CPU time.

METHODOLOGY

Initialization of the computational domain

A key element of this study is the way of initializing the computational domain. We chose in our study to initialize the domain by means of an input profile.Alternatively, other methods exist;the simplest method is by giving an initial guess for all the variables. Another method consists of initializing the domain with data provided by a coarser mesh, thus leading to faster convergence. Nevertheless, this is not in the scope of this article, as the aim here is to study the behavior of convergence regardless of the method of initialization.

Definition of convergence

First, we present an attempt to thoroughly define a converged state of our CFD calculation, in addition to doing it in a more general sense. This is actually a very difficult task to perform as convergence presents no universal character and its behaviour depends on each simulation, i.e. it needs to be determined case by case. Consequently, judging convergence requires not only the monitoring of the scaled residuals as a function of the number of iterations, but also of the important quantities (mean flow, turbulence, etc.). In a pressure-based solver, the scaled residuals are defined as the sum, over all of the grid cells,of the imbalance for a given variable φ (velocity, turbulent kinetic energy, etc.), divided by a scaling factor representative of the flow rate of φ trough the domain.

However, it is commonly accepted by the CFD community that a calculation is converged when the residuals drop to a very small quantity, or more often, when these residuals arrive to a plateau, as shown in Figure 2. In fact, this is what the majority of CFD users employ for a definition of a converged state. We shall make use of this definition of convergence in the present study. Consequently, we define the CPU time of convergence as the time needed to arrive to the plateau (referenced by the symbol ∞ in what follows). In this study, the ratio between the CPU time for a given number of iterations and that of the fully converged solution is defined as τ (Equation 1).

$$\tau = \frac{CPU\ time}{CPU\ time_{\infty}} \tag{1}$$

We also introduce the concept of error, defined as the difference between the value of a variable φ (velocity, turbulent kinetic energy, etc.) for a given τ and its value at τ=1, as expressed in Equation (2).

$$Error = \frac{|\varphi_{\tau=x} - \varphi_{\tau=1}|}{\varphi_{\tau=1}} \quad (2)$$

STUDY OF A CFD DOMAIN WITH A STEEP HILL

Simulation case

First, we examine the behavior of the convergence process of a CFD calculation in the presence of a steep hill. We therefore consider a case of a 2D domain with ahill identical in shape to the one used in the Almeida *et al.* (1991) experiment.This hill is rather steep, with a ratio of height over length of about 0.26, consequently leading to the creation, during the experiment, of a large recirculation region downstream of the hilltop. The hill is immersed in a neutral boundary layer with a free-stream velocity of 2.147 m/s, a friction velocity of 0.079 m/s and a roughness length of 0.015m.The CFD code chosen for this study is the well-known code Fluent. In order to retain a more general approach, we choose the k-ε turbulence model along with Duynkerke constants. Concerning our computational domain, the hill is positioned in the center of the domain, as shown in Figure 1. There are 6,000 grid cells in the domain and the boundary conditions are as follows:

- At the inlet of the domain, we apply analytical profiles of velocity *U*, turbulent kinetic energy *k*, and turbulent dissipation*ε*. The profiles are given in Sadek *et al.* (2011)
- A wall condition on the ground with a specified roughness length
- A symmetry condition at the top
- At the outlet of the domain, a profile of constant pressure

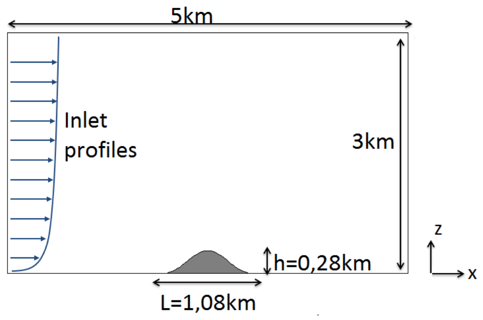


Figure 1 Schematic representation of the computational domain

As shownin Figure 2, the first iterations are the most costly in CPU time. In fact, the very first iteration is as costly as the next fiveiterations and as much as twenty iterations when falling to a lower threshold. These are particularly encouraging results becausethe more complex our terrain becomes, the more iterations the CFD codes needs in order to achieve a converged state, thus eliminating the costly effect of the first iterations.

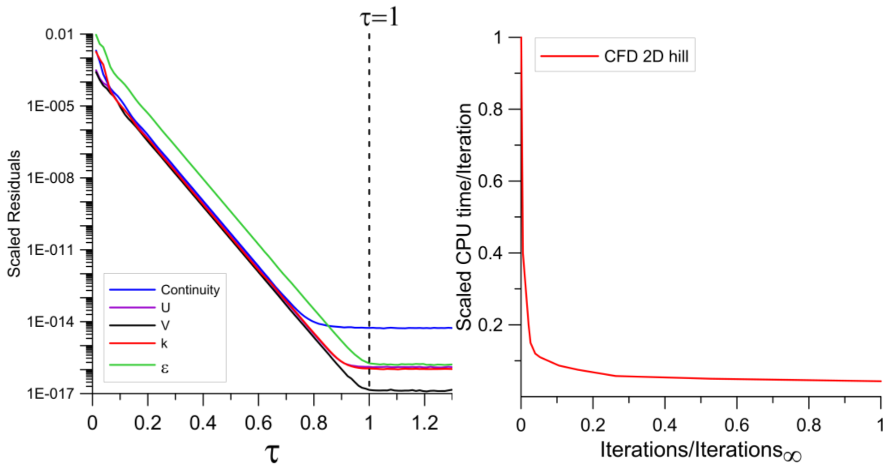


Figure2 Overview of the behaviour of iterations process of the case study of the 2D hill domain

Study of the flow and comparison with experimental data

In order to study the behavior of different variables (horizontal velocity U, vertical velocity W, turbulent kinetic energy k and dissipation rate ε) as a function of τ, we choose to plot these variables in locations where the flow is particularly complex: at the hilltop and downstream of the hill (Figure 3). By doing so, we can give a first estimate of the order of magnitude of τ necessary to give an acceptable solution.

As shown in Figure 3, after a short period of adaptation of the solution during the first iterations, the variables *U* and *k* quickly arrive at their converged values for τ=0.1. The same applies toW andε. Thus, the control points need 10 times less CPU time than the scaled residuals to arrive at a converged plateau. This can be explained by the fact that by definition, the scaled residuals sum the imbalance of a variable over all of the cells, including certain points of the flow where the residuals are particularly high. The numerous tests have proven that these points are usually located in small parts of the flow. Consequently, judging convergence by use of the residuals will not be as representative of the general physics of the entire flowas by using key control points. Using the experimental data from Almeida *et al.* (1991), we can compare profile results of *U* and *k* at different τ, in the recirculation region (at x=L), i.e. the most complex region of the flow. In this case, regarding U at τ=0.1, the solution gives a very good general representation of the flow (even in the recirculation region) along with a negligible error compared to the final solution. The U profile at τ=0.2 totally fits to that of the converged solution.
Concerning k, there is a significant difference between the CFD results at all τ, including τ=1, and the results of the experiment. Consequently, on average, choosing a solution at τ=0.1or τ=1 presents approximately the same error rate when compared to the experiment. The same argument can be made on the other variables.

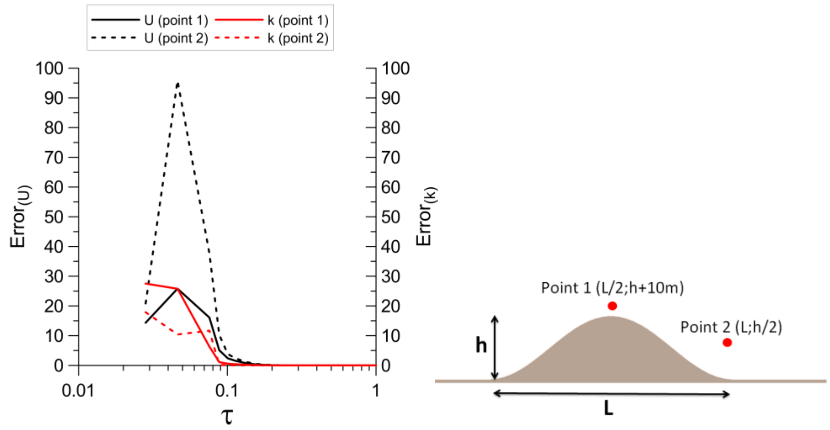


Figure 3 View of the evolution ofU andk at control points, as a function of τ

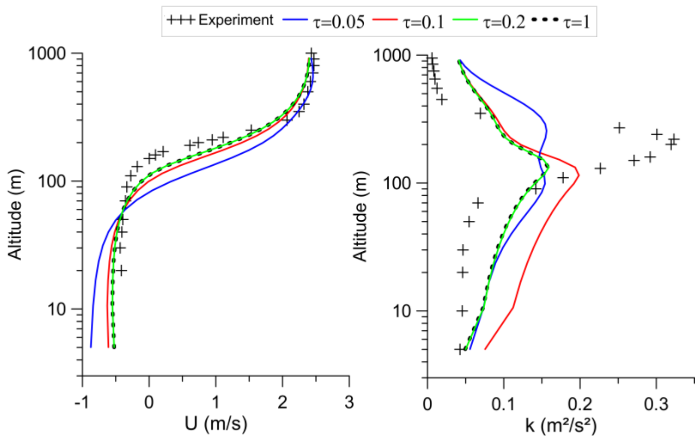


Figure 4 Profiles of U and k at x=L

Study of the recirculation region

One of the most important issues in pollutant dispersion studies is theprediction of recirculation regions, which may occur in region of steep hills and valleys. We therefore focus here on the ability of partially converged solution to simulate such an effect. As shown in Figure 5, the calculation at τ=0.1 simulates a well-developed recirculation region. Its height is approximately the same as that of the converged solution and its length is about 0.89L at τ=0.1 whereas it is 1.1L at τ=1.

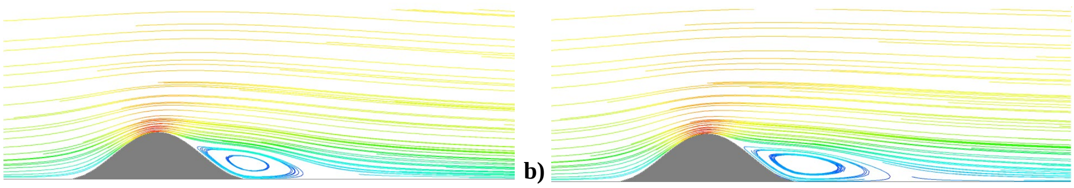
a)  b)

Figure 5 Streamlines of the flow, showing the recirculation region: τ=0.1 (a) and τ=1 (b)

In the experiment of Almeida *et al.* (1991), the recirculation length is approximately 1.3L, which indicates that the solution at τ=0.1 is more than acceptable. As we can see in the profile of U in Figure 4, the code simulates a recirculation zone even for τ=0.05, although in this case the height and the length are approximate.

Influence of the refinement of the computational domain and of the 2D/3D configuration

Convergence depends mainly on the complexity of the computational domain and on the refinement of the mesh. In an attempt to make a more general study, we perform a sensibility test by changing the number of grid cells while keeping the same geometrical domain. In what follows, we expressτ needed in order to achieve 5% error on Uat hilltop (point 1) for each of the different computational domains. We find that the more meshes our computational domain has, the lower the value of τ is at 5% error (Table 1). When multiplying the number of cells by a factor of 10 (the precision of the solution increases considerably), the CPU time is multiplied by approximately 17, whereas it is multiplied only by 4 when using a 5% error solution. The impact on the CPU time of a refined mesh is less important for a partially converged solution than for a fully converged one.

Table 6 τ at a 5% error of velocity U at hilltop (point 1) and the CPU time for different simulation cases with anIntel Core i5 750, along with a 3GB RAM .

Simulation Case	Number of cells	τ (5% error of U)	CPU time (s)	CPU time (5% error of U) (s)
2D hill	1500	0,184	48	8.8
2D hill	6000	0,088	123	10.8
2D hill	15000	0,046	440	20.2

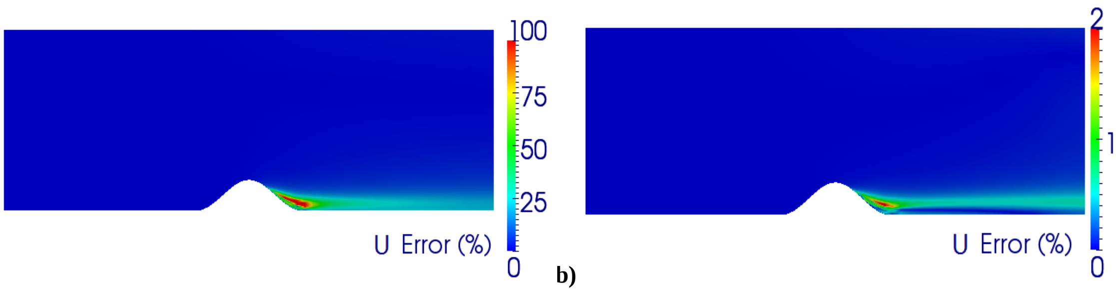
We now consider a 3D domain with a 3D hill, similar to the 2D Almeida *et al.* (1991) hill previously simulated but with an additional component. We observe that, similarly to the 2D case, τ at 5% error decreases with increasing number of grid cells. The rates of τ at 5% error are also similar, ranging from 0,085 to 0,115.

Next, we consider the case of two 3D domains with the same amount of grid cells (see Table 2), but while the first is with a regular horizontal meshing, the second is irregular, with more meshes near the hill than at the boundaries (with a ratio of 0.9).We find considerable differences between these two cases for a 5% error: τ=0.115 for the regular mesh while τ=0.038 for the irregular. In fact, there is a prediction of a recirculation region using only the irregular mesh.The higher calculations in the refined region near the hill and in the recirculation zone induces a larger number of iterations, which explains the low rate of τ in this case. Furthermore, Table 2 demonstrates that we can gain a considerable amount of CPU time by using an optimized irregular mesh with a partially converged solution.

Table 2τ at a 5% error of velocity U at hilltop (point 1) and the scaled CPU time for 3D caseswith a Intel Core i5 750, along with a 3GB RAM .

Simulation Case	Number of cells	τ (5% error of U)	CPU time (s)	CPU time (5% error of U) (s)
3D hill	219500	0,115	224	25.8
3D hill	219500(irregular mesh)	0,038	1102	41.9
3D hill	1182000	0,085	2910	247.3

We now retain the case of the 3D hill with the irregular mesh, and plot the contours of errors U, k and εat τ=0.05 and τ=0.1, in order to locate the maximum of error in the whole of the computational domain. InFigures 6, 7 and 8, we can see that the maximum errors are situated mainly in the recirculation region and downstream of the hill.One should note that the errors of U in the recirculation region are particularly high because of its low values in this region. This confirms the fact that one should monitor values at key control pointssituated in the wake of the hill, as in Figure 3, in order to judge of the convergence of a simulation, and not necessarily by monitoring the scaled residuals.



a) **b)**

Figure 6: Error contours for **U** a) τ=0.05 and b) τ=0.1 for the 3D irregular mesh

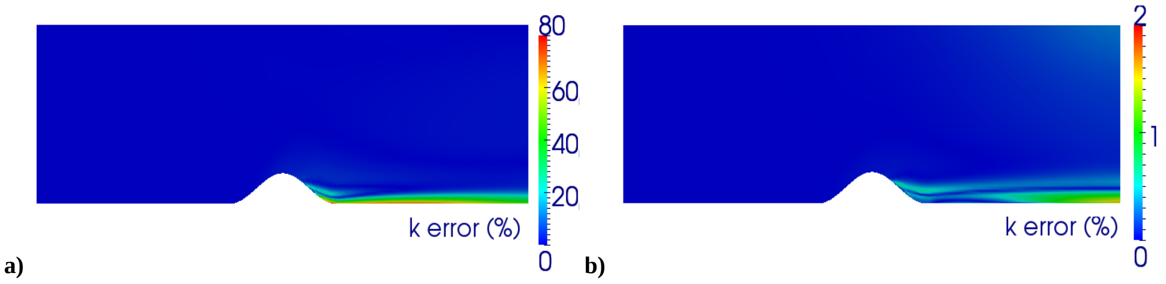


Figure 7: Error contours for **k** for a) τ=0.05 and b)τ=0.1 for the 3D irregular mesh

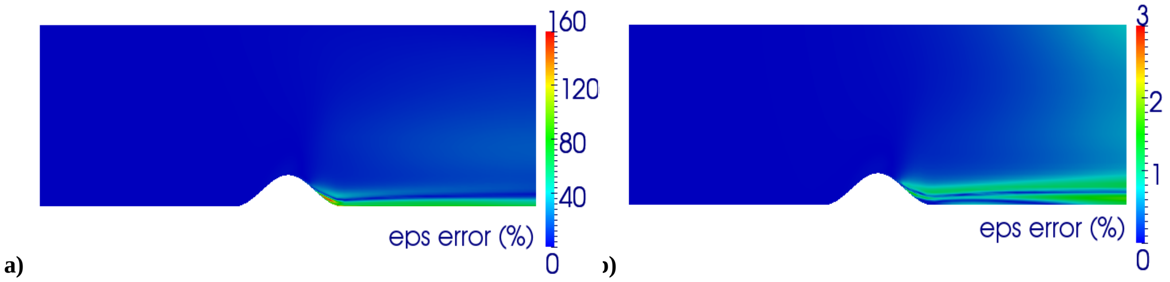


Figure 8: Error contours for **ε** for a) τ=0.05 and b) τ=0.1 for the 3D irregular mesh

We can also notice that the error increases along the direction of the flow.This is particularly interesting for applications in the presence of dispersion of pollutants. Indeed, as the pollutant moves away from the source, it becomes less dependent on theexact values of the flow's variables. Furthermore, several tests have proven that the effect on the pollutant dispersion of a flow which differs, for example, by 5% to that of the converged solution is very small.

STUDIES OF OTHER CASES

In order to be as general as possible, we simulate cases of hills with different ratios. We choose the hills corresponding to those of the EPA RUSHIL experiment (Khurshudyan *et al*., 1981), with ratios from h/L=0.062 to 0.166. Fluent gives very accurate results in simulation with these hills, as shown in Sadek *et al.* (2011). We find that τ at 5% error ranges between 0.1 and 0.12 for these hills, with a 12,000 cell computational domain. These rates are comparable to those found previously. Furthermore, we also investigate the possibility of using partially converged solutions in the presence of a surface heat flux or roughness change. In order to do so, we simulate two cases by replacing the 3D Almeida *et al.* (1991) hill with patches of either heat flux and roughness length of approximately the same horizontal length.

- A surface heat flux of 50 W/m² is applied at a patch of 1km X 1km in the centre of the domain, 0W/m² elsewhere
- A roughness length of 0.1m is applied at a patch of 1km X 1km in the centre of the domain, 0m elsewhere

In order to compare with the previous results in the presence of a hill, we express results in terms of τ needed to achieve 5% error on the vertical velocity W. We found that τ at 5% error is approximately 0.068at the centre of the heat flux patch and 0.074 at the centre of the roughness patch. These rates are comparable to the ones calculated in the presence of the hill.

CONCLUSION AND RECOMMENDATIONS

In spite of the non-general character of the convergence process, we have proven that it is possible to reduce the CPU time while keeping the error low for several simulations of different cases of complex terrain (relief, roughness and surface heat flux change). In each of our simulations, we found that 5-10% of the usual CPU time issufficient to give an acceptable solution, even in complex simulations as in regions of recirculations. This is an interesting idea that can be used for applications of atmospheric dispersion in operational air quality assessments.

For practical purposes, in order to judge of the convergence of a calculation,we recommend therefore the monitoring of the calculation process of certain variables at key control points that a user should identify depending on the complexity of the terrain.

REFERENCES

Almeida, G.P., Durao, D.F.G. & Heitor, M.V., 1992: Wake flows behind two dimensional model hills. *Exp. Thermal and Fluid Science, 7, p.87*

Khurshudyan, L.H and Snyder , W.H & Nekrasov, I.V, 1981: Flow and dispersion of pollutants over two-dimensional hills. *U.S. Env. Prot. Agcy. Rpt. No. EPA-600/4-81-067. Res. Tri. Pk., NC.*

Sadek, R., Soulhac, L., Brocheton, F., Buisson,E.,2011:Evaluation of wind field and dispersion models in complex terrain, *Accepted in HARMO 14 conference.*