



ANALYSIS OF THE NON-ORTHOGONALITY CORRECTION OF FINITE VOLUME DISCRETIZATION ON UNSTRUCTURED MESHES

Antonio de Oliveira Samel Moraes

Paulo Laranjeira da Cunha Lage

Programa de Engenharia Química. Laboratório de Termofluidodinâmica (LTFD/PEQ/COPPE/UFRJ), Rio de Janeiro, Brasil
amoraes@peq.coppe.ufrj.br, paulo@peq.coppe.ufrj.br

Guilherme Coelho Cunha

Instituto Nacional de Pesquisas Espaciais (INPE), Cachoeira Paulista/SP, Brasil
guilherme.cunha@lcp.inpe.br

Luiz Fernando Lopes Rodrigues da Silva

Escola de Química (EQ/UFRJ), Rio de Janeiro, Brasil
lflopes@eq.ufrj.br

Abstract. *Computational fluid mechanics with finite volume method needs high-quality computational meshes that can be structured or not. When properly constructed, structured meshes provide several advantages over unstructured ones as, for instance, (i) higher metrics quality, (ii) a corresponding matrix of fixed bandwidth and (iii) a straightforward numerical implementation due to a simple and effective variable indexation. However, their lower adaptability to complex geometries can be considered as their main drawback. On the other hand, unstructured meshes are more flexible with respect to geometrical topology, at the expense of some loss regarding the advantages cited above. As a matter of fact, the non-collinearity between the center-to-center vector of adjacent cells and the normal vector of the face that connects these cells makes difficult the approximation of the gradient of a field at the face. A common procedure to overcome this problem uses a non-orthogonality correction. In this work, a methodology to compute this correction taking account the mesh skewness was implemented and tested. It aims to obtain an iterative method that preserves second order of accuracy for the local approximation of the variables and their gradients on the mesh. Two test cases were addressed and the results compared to that obtained via the CFD softwares OpenFOAM® and ANSYS Fluent®.*

Keywords: *Numerical methods; Unstructured Meshes; Nonorthogonality correction; Skewness correction; Finite Volume Method*

1. Introduction

In real-life engineering applications, several physical and chemical phenomena are modelled by the continuum mechanic equations, e.g. those related to incompressible and compressible flows, multiphase flows, reactive flows, combustion and fluid-structure interaction. In the past decades, the continuous advances on computational means along with the development of numerical methods to solve these systems of equations in complex geometries allowed the simulation of real-life problems of engineering and enabled the diffusion of the so-called Computational Fluid Dynamics (CFD) in companies and research centers.

Among these numerical methods, the finite volume method (FVM) is among the most important (Hirsch, 1991; Patankar, 1981) and is the main method implemented in several dedicated softwares such as ANSYS CFX, ANSYS Fluent, StarCCM, CFD++ and OpenFOAM, the latter being open source. The main characteristic of the FVM, which is responsible for its disseminated use in CFD methods, is its flexibility using in generalized meshes. The method does not impose any constraints to the structure of the mesh neither to the type or form of its elements. In other words, the method has support to structured or non structured polyhedral meshes (Jasak, 1996).

Particularly, the advances in methods for automatic generation of unstructured meshes experienced in the past decades significantly facilitated the mesh generation for complex geometries. However, this type of meshes frequently has lower quality when compared to the structured ones. Two metrics frequently used to measure this quality are the (i) mesh non-orthogonality and (ii) skewness (Boris and Book, 1973; Leonard, 1979). These two metrics are strongly associated with numerical errors in the numerical solution calculated via FVM. It has to be noted that, if not corrected, the order of accuracy of the spatial discretization is likely to be degraded. A common procedure to overcome this problem uses a non-orthogonality correction.

In the present work, a methodology to compute the non-orthogonality correction including the skewness correction is implemented and tested. It aims at obtaining an iterative method that preserves second order of accuracy for the local approximation of the variables and their gradients on the mesh. In order to test the effect of this skewness correction, a stationary diffusion problem is addressed and the results compared to that obtained by the the OpenFOAM 2.1.1 and Fluent 13.0 CFD softwares.

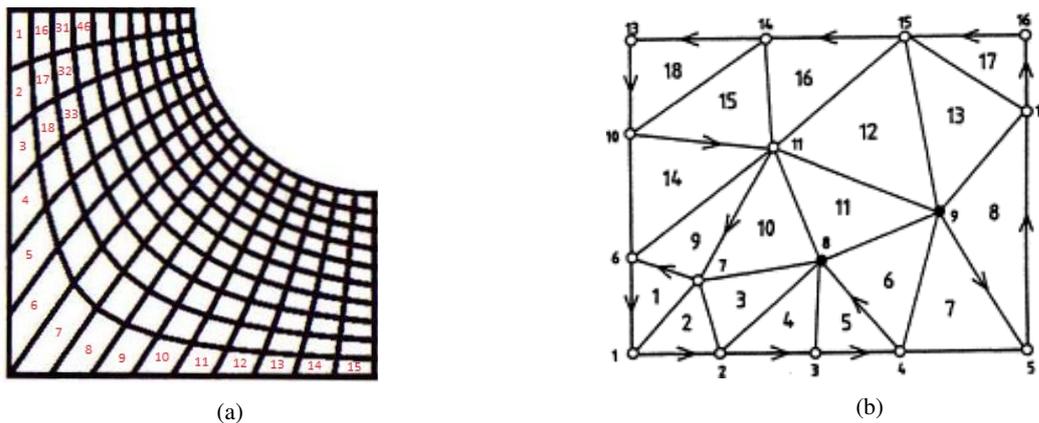


Figure 1: Examples of (a) structured and (b) unstructured meshes.

2. Structured and Unstructured Meshes

2.1 Structured Meshes

According to Maliska (1995), one mesh is called structured when (i) any of its internal elements have the same number of neighbours and (ii) the mesh is obtained from a coordinate system and, therefore, the elements have a natural sequence. In other words, the mesh has a connectivity that can be expressed as a two or three dimensional array. This mesh structure has several advantages, among which stand out: (i) the resulting matrix of the finite volume discretization has fixed bandwidth, (ii) the mesh ordering allows easy indexation of the variables and (iii) the excellent quality metrics, when properly constructed.

A fixed bandwidth is very important because the iterative linear solvers are more effective on this type of matrix (Maliska, 1995). Additionally, the explicit connectivity facilitates the variable indexation and, consequently, the increase in the discretization order, which is important when high accuracy is desired as, for example, in aero/acoustics simulations; as a matter of fact, due to the large disparities between aerodynamic and acoustics length scales, high order spatial discretization is necessary to preserve accuracy (Cunha and Redonnet, 2012). Finally, the quality metrics of the structured meshes are usually better than those of the unstructured ones.

In general, in structured meshes, the elements are better aligned and less deformed, which minimizes errors associated with (i) the non-orthogonality between the center-to-center and face normal vectors and (ii) the distance between the point where the center-to-center line crosses the face belonging to the two neighbours volumes and the face center point. The metric associated to the latter is defined as element skewness (Jasak, 1996).

As previously stated, the main drawback of structured meshes is their low versatility when it comes to fit complex geometries. In other words, it is difficult to build high-quality structured meshes for complex geometries, which are more a rule than an exception for real-life engineering problems. However, it should be highlighted that the investment in the construction of the high quality structured meshes (even for complex geometries) should be preferred whenever possible, because it usually pays off by diminishing the computational cost and increasing of the simulation accuracy.

2.2 Unstructured Meshes

Differently from structured meshes, the unstructured ones are characterized by the fact that: (i) the number of neighbours depends of the element geometric type, (ii) the connectivity is irregular and, therefore, the variable indexation can not be readily expressed as a two or three dimensional array in the computer memory and (iii) in general, the quality metrics are inferior to those obtained for structured properly constructed ones.

As a result of the first characteristic cited above, the matrix resulting from the finite volume discretization shows variable bandwidth. For this type of matrix, the iterative linear solvers are less effective or, in other words, require more iterations to achieve convergence. However, this problem can be minimized by the use of appropriate techniques for reducing a sparse matrix bandwidth, such as the *reversed Cuthill-McKee* algorithm (Cuthill and McKee, 1969).

Figure 1 shows examples of bi-dimensional structured and unstructured meshes. As can be seen for the structured mesh shown in Fig. 1(a), the bandwidth of all internal elements is 30. From the unstructured one, shown in Fig. 1(b), it can observe that, for example, the element 11 is connected to elements 6, 10 and 12, resulting in a bandwidth equal to 6 in the line 11 of the matrix. On the other hand, the element 10 is connected to elements 3, 9 and 11, resulting in a bandwidth equal to 8 in the line 10.

Regarding the characteristic (ii) aforementioned, the main consequence relies on the fact that the spatial discretization order is linked to the connectivity of the elements. As a matter of fact, only in very special cases is possible to obtain a

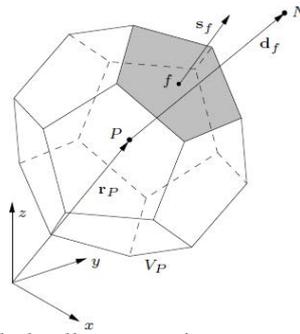


Figure 2: Polyhedral cell representing a control volume of FVM.

spatial discretization order greater than the one imposed by the maximum stencil size (which is limited by connectivity restrictions).

Finally, the quality metrics of the unstructured meshes is, in general, worse than those obtained for structured ones. As mentioned in Section 1, bad values of these metrics tend to introduce errors in the numerical solution and, consequently, degrade the spatial discretization order of accuracy. In order to overcome this problem, correction techniques are necessary. Once again, it is noteworthy that this type of mesh is very flexible to geometrical topologies, making easier the automatic mesh construction for complex geometries.

3. Non-orthogonality correction

Among the metrics used for calculating mesh quality is the angle between the face normal vector, s_f , and the element center-to-center vector, \overline{PN} , where P and N represent two neighbour cell centers, as shown in Fig. 2). This is the so-called mesh non-orthogonality, which cannot be neglected and, if not properly handled, can lead to loss in solution accuracy (Leonard, 1979).

The non-orthogonality correction is one of the ways to treat the problem and is applied to calculate the projection of the gradient over the normal face area vector through a given face f , usually associated with the diffusive term of a transport equation. The procedure proposed by Ferziger and Perić (1997) consisted in a *deferred correction approach*, in which the projection of the gradient vector (or tensor, depending on the vector or tensor nature) onto the face normal vector is split into an orthogonal contribution, which is implicitly treated, and a non-orthogonal contribution, which is treated explicitly (Jasak, 1996). Jasak (1996) proposed a modified form of this procedure, which is implemented in the OpenFOAM software (Weller *et al.*, 1998), which was the base of the non-orthogonality correction implemented in the present work. Before describing the non-orthogonal correction, the gradient evaluation in a generalized polyhedral mesh using the finite volume method is described in the following.

3.1 Gradient operator discretization

In the finite volume method, the gradient evaluation at the centroid P of a discretized control volume can be calculated in different ways. Two of them are: (i) the Gauss integration and (ii) the Least Square Method (C.R. Rao and Scheid, 1999). In the OpenFOAM and Fluent softwares, these methods as well as its face and cell limited versions are available. More particularly, the Gauss integration is based on the Gauss theorem, given by:

$$\int_{V_P} \nabla \phi dV = \oint_{\partial V_P} (\mathbf{n}\phi) dS \quad (1)$$

where ϕ is the scalar solution variable, V_P is a volume of a fluid region around the the point P , ∂V_P is the control surface and \mathbf{n} is the normal surface vector. Figure 2 illustrates the polyhedral cell that shall be considered in the present work.

Applying the mean value theorem (second order accurate) to Eq. 1, the discretized gradient at the centroid point P is given by:

$$(\nabla \phi)_P \cong \frac{\sum_f s_f \mathbf{n}_f \phi_f}{V_P} \quad (2)$$

The variable value ϕ_f at each face must be obtained by interpolation from the centroid values of the adjacent volumes. Assuming that the line \overline{PN} connecting two adjacent centroids crosses the face at its centroid, f , the linear interpolation, given by Eq. 3 preserves second order of accuracy.

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (3)$$

with:

$$f_x = \frac{\overline{fN}}{\overline{PN}} \quad (4)$$

where \overline{fN} is the distance between face centroid, f , and the centroid of element N and \overline{PN} is the distance between the centroids of the P and N elements (or cells). It should be noted that Eq. 3 ignores the term associated with the mesh skewness.

The Least Square Method is given by Eq. 5.

$$(\nabla\phi)_P = \sum_N w_N^2 \mathbf{G}^{-1} \mathbf{d}(\phi_N - \phi_P) \quad (5)$$

where N extends to all neighbours of the cell P , \mathbf{d} is the vector from centroid P to the centroid N and the weights w_N and the matrix \mathbf{G} are given by:

$$w_N = \frac{1}{\|\mathbf{d}\|} \quad (6)$$

$$\mathbf{G} = \sum_N w_N^2 \mathbf{d}_N \otimes \mathbf{d}_N \quad (7)$$

As can be seen, the two methods presented allow the calculation of the gradient at the control volume centroids. Such a calculation depends on the values of the solution variables at the neighbour centroid control volumes. It is also worth pointing out that the equations above do not impose any constraints on the element geometrical type. In other words, the finite volume discretization has polyhedral mesh support.

3.2 Diffusion operator discretization and the non-orthogonality correction

Analogously to the gradient operator discretization, the Gauss theorem can be used to evaluate the diffusion operator, according to Eq. 8.

$$\int_{V_P} \nabla \cdot [\gamma \nabla \phi] dV = \oint_{\partial V_P} \gamma (\mathbf{n} \cdot \nabla \phi) dS \quad (8)$$

where γ is the diffusion coefficient that, for anisotropic materials, can be a second order tensor.

Again, applying the mean value theorem, we get:

$$\oint_{\partial V_P} \gamma (\mathbf{n} \cdot \nabla \phi) dS \cong \sum_f \gamma_f (\nabla \phi)_f \cdot \mathbf{s}_f \quad (9)$$

where $\mathbf{s}_f = \mathbf{n}_f s_f$ being s_f the face area and $\hat{\mathbf{n}}_f$ is the normal unity vector evaluated at the face centroid.

As can be seen, Eq. 9 requires the normal gradient evaluation at different face centroids. The gradient at a given face could be obtained from interpolation of its values at adjacent volume centroids. However, for an implicit implementation of Eq. 9, a *larger computational molecule* would be needed (Ferziger and Perić, 1997). For example, a bi-dimensional mesh formed by quadrilaterals elements would involve 13 elements for a second order accurate discretization of the diffusion operator.

For computational efficiency, one tends to use only the elements adjacent to the face to approximate the derivative of ϕ in the normal direction using the gradient projected in the \mathbf{d} direction, according to Eq. 10.

$$\mathbf{s}_f \cdot (\nabla \phi)_f = |\mathbf{s}_f| \frac{\phi_N - \phi_P}{|\mathbf{d}|} \quad (10)$$

If the \mathbf{d} and the \mathbf{s}_f vectors are aligned, Eq. 10 is second order accurate, because $\hat{\mathbf{d}} \cdot \nabla \phi = \hat{\mathbf{s}}_f \cdot \nabla \phi$. However, for non-aligned elements (or cells), in order to preserve second order accuracy a correction is necessary. The *deferred correction approach* is suggest by Ferziger and Perić (1997) and followed by Jasak (1996). In this procedure, the projection of the face gradient onto the face normal vector is separated in a projection onto the orthogonal contribution vector, Δ , and the non-orthogonal correction one, which is its projection onto the \mathbf{k} vector, as showed in Fig. 3, where $\mathbf{s}_f = \Delta + \mathbf{k}$.

Jasak (1996) suggest three approaches to calculate the orthogonal vector. In the present work, we use the over-relaxed one, given by Eq. 11.

$$\Delta = \frac{\mathbf{d}}{\mathbf{d} \cdot \mathbf{s}_f} |\mathbf{s}_f|^2 \quad (11)$$

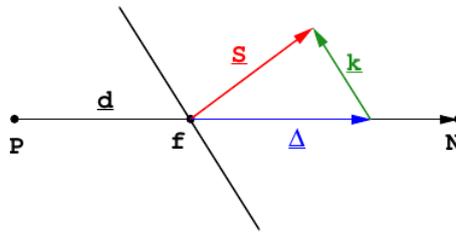


Figure 3: Nonorthogonality between center-to-center and face normal vector. Treatment in the over-relaxed approach. Figure extracted from Jasak (1996).

Therefore, the deferred correction approach is given by Eq. 12.

$$(\nabla\phi)_f \cdot \mathbf{s}_f \cong (\nabla\phi)_f \cdot \Delta + [(\nabla\phi)_f \cdot \mathbf{k}]^{expl} \tag{12}$$

where the superscript *expl* refers to an explicit evaluation of the non-orthogonal term using the current value of the gradient at the centroid of the face, which is obtained by interpolation from the gradient values at the centroids of the adjacent volumes of the face. Again, assuming that the line connecting two adjacent centroids, *P* and *N*, crosses the face also at its centroid, the linear interpolation, as showed in Eq. 13, ensures second-order order of accuracy. The gradient value at the element centroids can be calculated using Eq. 2 or 5.

$$(\nabla\phi)_f = f_x(\nabla\phi)_P + (1 - f_x)(\nabla\phi)_N \tag{13}$$

The orthogonal contribution of the gradient is calculated using the value of the variable at the centroid of the adjacent volumes, as in Eq. 14.

$$\Delta \cdot (\nabla\phi)_f = |\Delta| \frac{\phi_N - \phi_P}{|d|} \tag{14}$$

Therefore, the non-orthogonality correction requires an iterative procedure, which is internal to the solution procedure of the transport equations. Given that one has to ensure that $\mathbf{k} = \mathbf{s}_f - \Delta$, if the procedure is convergent, the correction will preserve 2^{nd} order of accuracy. However, when the non-orthogonal term is larger than the orthogonal one, the iterative correction can become unbounded and then needs to be limited, which is made according to Eq. 15.

$$\Delta \cdot (\nabla\phi)_f > \lambda[\mathbf{k}_f \cdot (\nabla\phi)_f] \tag{15}$$

where λ is the limiter. The limitation can ensure convergence, but degrades the second-order accuracy of the discretization. Typical values used for λ are 0.5 and 0.333.

3.3 Skewness correction

Equations 3 and 13, which interpolate the variable and its gradient to the faces centroids, are second order accurate only when the vector \mathbf{m} (so-called *skewness vector*), showed in Fig. 4, is null. As can be seen, the skewness vector is the vector defined by two points: (i) the point where the vector \mathbf{d} crosses the respective face and (ii) the centroid face, *f*.

Therefore, the non-orthogonality correction procedure described in the previous section, when not artificially bounded, also preserves second order of accuracy only when the vector \mathbf{m} is null. When it is not the case, the equations cited above must be corrected. In order to do so, we consider the second order Taylor series expansion around the centroid of the element *P* to approximate the solution variable at the face centroid, given by Eq. 16.

$$\phi_f = \phi_P + (\nabla\phi)_P \cdot \mathbf{d}_{Pf} = \phi_P + (\nabla\phi)_f \cdot \mathbf{d}_{Pf} \tag{16}$$

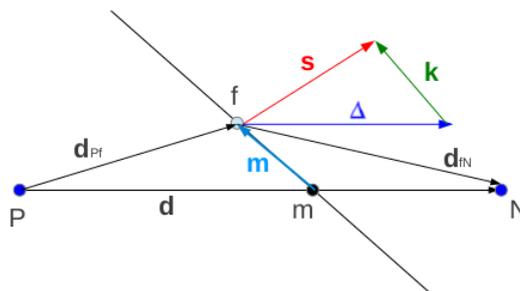


Figure 4: Skewness vector connecting the crossing point and the face centroid point.

From Fig. 4, we can be seen that:

$$\mathbf{d}_{Pf} = \mathbf{d}_{Pm} + \mathbf{m} = (\mathbf{d} - \mathbf{d}_{mN}) + \mathbf{m} \quad (17)$$

where \mathbf{d}_{Pm} is the vector from point P to point m , \mathbf{d}_{mN} is the vector from point m to point N and \mathbf{d} is the vector from point P to point N . Therefore:

$$\phi_f = \phi_P + (\nabla\phi)_f \cdot \mathbf{d} - (\nabla\phi)_f \cdot \mathbf{d}_{mN} + (\nabla\phi)_f \cdot \mathbf{m} \quad (18)$$

The second and third terms at the right hand side of the Eq. 18 are given, respectively, by Eqs. 19 and 20.

$$(\nabla\phi)_f \cdot \mathbf{d} = \phi_N - \phi_P \quad (19)$$

$$(\nabla\phi)_f \cdot \mathbf{d}_{mN} = \frac{\phi_N - \phi_P}{|\mathbf{d}|} |\mathbf{d}_{mN}| \quad (20)$$

Inserting Eqs. 19 and 20 in Eq. 18, we have:

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N + \mathbf{m} \cdot (\nabla\phi)_f \quad (21)$$

with, in this case:

$$f_x = \frac{|\mathbf{d}_{mN}|}{|\mathbf{d}|} \quad (22)$$

Finally, the gradient interpolation at the face centroid is given by:

$$\nabla\phi_f = f_x \nabla\phi_P + (1 - f_x) \nabla\phi_N + \nabla[\mathbf{m} \cdot (\nabla\phi)_f] \quad (23)$$

Therefore, in order to preserve second order of accuracy by taking the skewness error into account, the variable and its gradient at the face must be approximated by Eqs. 21 and 23. If these equations are used for this propose, the non-orthogonality correction equation given by Eq. 14 will be also second order accurate, in the absence of artificial limitation correction.

In the present work, we implement a finite volume discretization considering the skewness correction given only by Eq. 21. The correction given by the Eq. 23 is the subject of future work. Therefore, the skewness correction implemented here is still partial. It is noteworthy that, analogously to the non-orthogonality correction, the skewness correction is explicitly included in the Eq. 21 and in the same loop of that iterative procedure. In other words, the implemented skewness correction corresponds to replacement of the Eq. 3 by Eq. 21 in the non-orthogonality correction procedure described in the section 3.2

4. Results

Two test problems were used in the comparison among the results of three different codes: the OpenFOAM, ANSYS Fluent and an in-house code with the proposed correction method implemented. For both test cases, a steady-state diffusion problem with constant diffusion coefficient is addressed or, more specifically, the Laplace equation was solved. Equation 24 shows the model to be used for the evaluations. The focus on a pure diffusive problem was given because, in general, the non-orthogonality effects are more important for this kind of problem.

$$\nabla \cdot [\nabla[\phi]] = 0 \quad (24)$$

The finite volume discretized equations, including the nonorthogonality correction, is given by Eq. 25.

$$\sum_f [(\nabla\phi)_f \cdot \Delta]^k = -b_P^{k-1} \quad (25)$$

where:

$$b_P^{k-1} = \sum_f [(\nabla\phi)_f \cdot (\mathbf{s}_f - \Delta)]^{k-1} \quad (26)$$

and k indicates the current iteration.

First and second type boundary conditions are used for this problem. Their implementation is made according to Jasak (1996), where it is assumed that the specified value for the boundary condition is valid along the entire boundary face and, therefore, for a constant specified value, the non-orthogonality correction is not applied in the boundary faces.

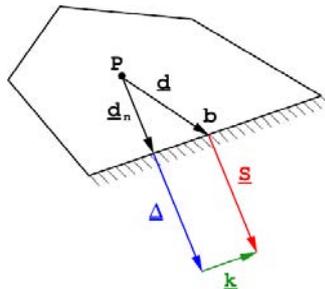


Figure 5: Control volume with a boundary face. Extracted from Jasak (1996). In this figure, the vector s_b is represented by the vector \mathbf{S} and the vector \mathbf{d} is the vector from cell centroid P to face centroid b .

The vector \mathbf{d}_n , shown in Fig. 5, which links the cell center to the boundary face, is normal to the boundary and it is given by:

$$\mathbf{d}_n = \frac{\mathbf{d} \cdot \mathbf{s}_b}{|\mathbf{s}_b|} \frac{\mathbf{s}_b}{|\mathbf{s}_b|} \quad (27)$$

where, in this case, \mathbf{d} is the vector from cell centroid P to face centroid b , and s_b is the boundary face area vector, represented in the Fig. 5 by the vector \mathbf{S} .

Equations 28 e 29 illustrate the implementation of the boundary conditions of first and second types.

- **First type BC (fixed-value BC):**

$$\mathbf{s}_b \cdot (\nabla \phi)_b = |\mathbf{s}_f| \frac{\phi_b - \phi_P}{|\mathbf{d}_n|} \quad (28)$$

- **Second type BC (fixed-gradient BC):**

$$\mathbf{s}_b \cdot (\nabla \phi)_b = g_b \quad (29)$$

where ϕ_b and g_b are the specified boundary value for the solution variable and its gradient, respectively.

4.1 Test problem 1: Benchmark Problem

In order to validate the implemented in-house code, we first consider a non-orthogonal mesh, as showed in Fig. 6. The coordinates of the geometric model are defined for $x \in [0, 2]$ and $y \in [0, 1]$. For this mesh the angle between the face area vector and the center-to-center vector is 45° for the left and right faces and it is zero for the top and bottom ones. The skewness vector, as defined in the present work, is null for all faces of this mesh. For LEFT and BOTTOM boundary faces, a second type boundary condition with $g_b = 0$ was used, whereas for the TOP and RIGHT boundary faces a first type one was preferred, with $\phi_b = 1$ at the RIGHT boundary and $\phi_b = 0.1$ on the TOP one.

Figure 7 shows the results of the simulation with OpenFOAM 2.1.1, our in-house code, from now called Own Code, and Fluent 13.0. For OpenFOAM and Own Code two iterations of the non-orthogonality corrections were used. It was not possible to configure numerical schemes for diffusion term in Fluent 13.0 directly; Nevertheless, the results suggest that this software uses some kind non-orthogonality correction procedure.

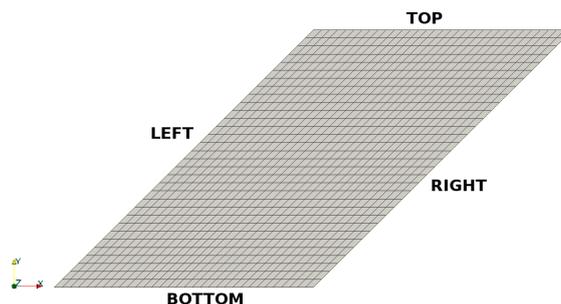


Figure 6: Mesh used for test problem 1.

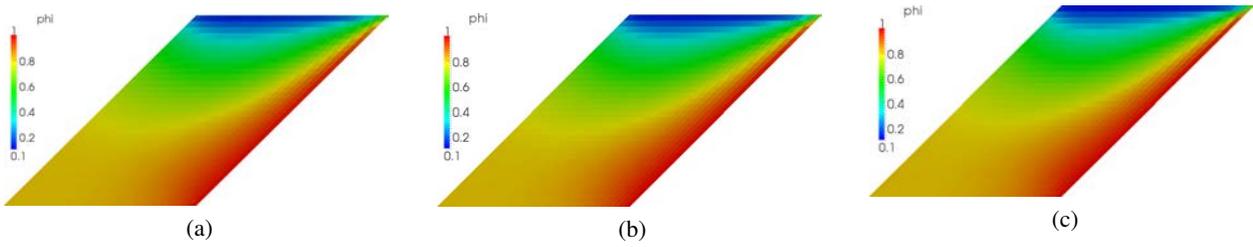


Figure 7: Contour plot of ϕ obtained by (a) OpenFOAM 2.1.1, (b) Own Code and (c) Fluent 13.0.

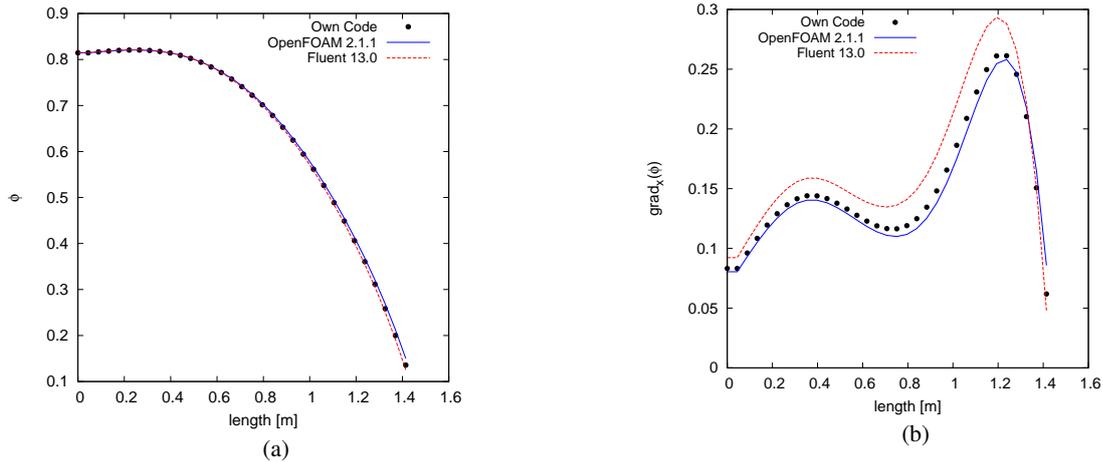


Figure 8: Profiles along the straight line that joint the points $(0.5, 0, 0.05)$ and $(1.5, 1, 0.05)$ using non-orthogonality correction approach. (a) ϕ , (b) x-component of $\nabla\phi$.

Figure 8 shown the profile of the solution variable (ϕ) and the x-component of its gradient along the straight line that joint the points $(0.5, 0, 0.05)$ and $(1.5, 1, 0.05)$ obtained with Own Code, OpenFOAM 2.1.1 and Fluent 13.0. Again, the results with Own Code and OpenFOAM 2.1.1 were obtained using two iterations in the correction loop. Figures 7 and 8 show a very good agreement between the results obtained with the three solvers and, therefore, the Own Code can be considered to be validated.

Since the Own Code is validated, its results depicted in Fig. 9 can be used to illustrate the effect of the non-orthogonal correction on the solution. As can be seen, the contour plot of the solution variable is greatly influenced by the correction. Figure 10 shows the profiles of the solution variable and the x-component of its gradient along the line described above. The effect of the non-orthogonal correction is even more pronounced for the gradient values.

4.2 Test Problem 2: Analysis of skewness correction

The present test case addresses the validation of the skewness correction method proposed in the present work. For doing so, substituted Eq. 3 by Eq. 21 in the evaluation of the gradient at the cell centers which are used in the non-orthogonality procedure, being the cell centers gradients evaluated by Gauss integration (cf. section 3.1). Such a feature is implemented in our Own Code in order to compare its results to those obtained with OpenFOAM 2.1.1 and ANSYS Fluent 13.0. The coordinates of the geometric model are defined for $x \in [0, 6.1]$ and $y \in [0, 4]$, being the origin of ARC1 and ARC2, respectively, the points $(0, 0, 0)$ and $(6.1, 0, 0)$. Equation 24 was solved on two meshes: a coarse unstructured mesh with large skewness and bad element size transitions and a fine structured mesh exclusively formed by quadrilateral

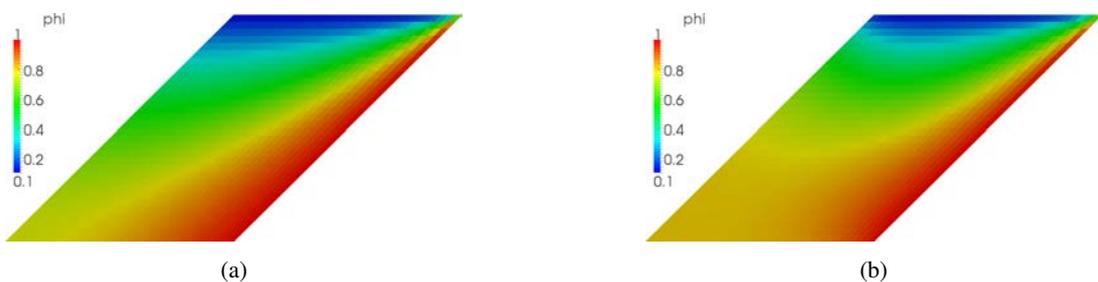


Figure 9: Effects of the non-orthogonal correction on the solution variable (ϕ). (a) Uncorrected, (b) Corrected.

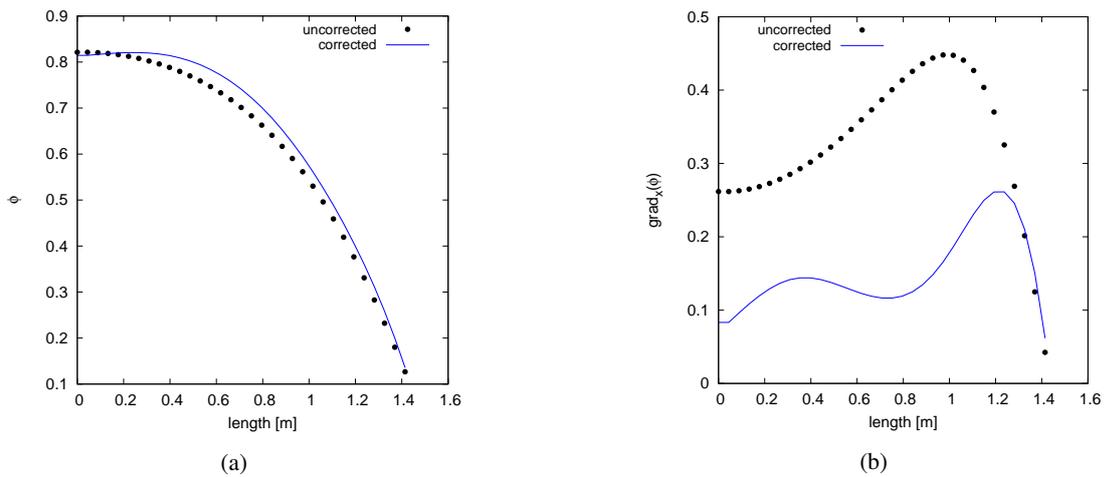


Figure 10: Effects of the non-orthogonal correction on the profiles along the straight line that joint the points (0.5, 0, 0.05) and (1.5, 1, 0.05) of the (a) Solution variable and (b) the x-component its gradient.

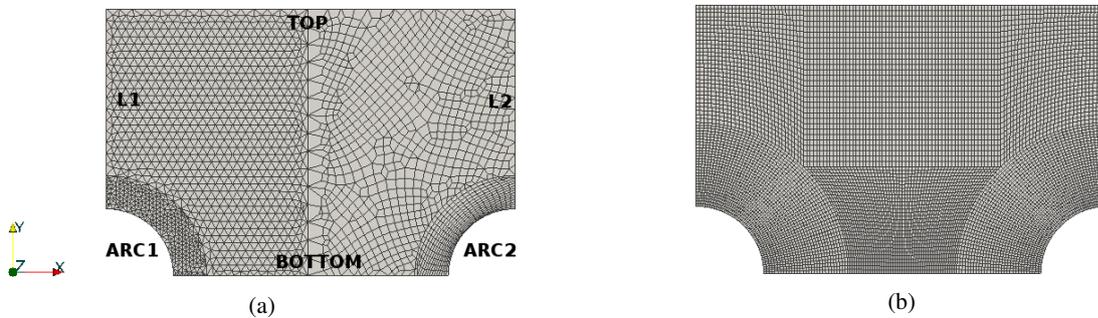


Figure 11: Computational meshes used for test problem 2: (a) the coarse unstructured mixed mesh with triangular and quadrilaterals elements and (b) The structured quadrilateral meshes.

elements. Figure 11 shows the meshes. For the L1, L2 and BOTTOM boundary faces a second type boundary condition with $g_b = 0$ was used and, for the ARC1, ARC2 and TOP boundary faces, first type one was applied with $\phi_b = 0.1$, $\phi_b = 1$ and $\phi_b = 10$, respectively.

The mixed mesh is composed with 1117 quadrilaterals and 2114 triangles, whereas the structured mesh has 9000 quadrilaterals, both constructed using the Salome Platform (Salome, 2013). Figure 12 shows the *Equiangular Skewness* of the two meshes. It has to be noted that the definition of *Equiangular Skew* is different from the skewness definition used in the present work. Nevertheless, their values are strongly related. According to the Tgrid®manual (FLUENT Inc., 2006), the *Normalized Equiangular Skewness* is defined by Eq. 30:

$$Skew = \max \left[\frac{Q_{max} - Q_e}{180 - Q_e}, \frac{Q_e - Q_{min}}{Q_e} \right] \tag{30}$$

where Q_{max} is the largest angle on the face or cell, Q_{min} is the smallest one and Q_e is the angle of equiangular face or cell (60° for a triangle and 90° for a square).

Figure 13 compares the contour plot of the solution variable obtained on unstructured mixed mesh using our Own

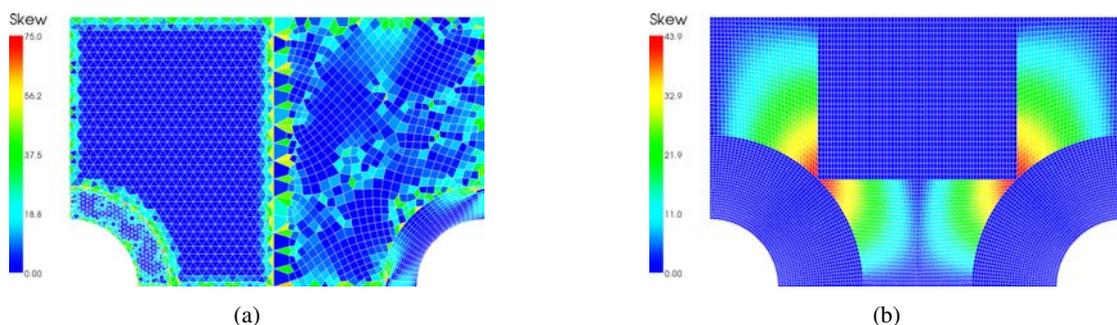


Figure 12: Equiangular Skewness: (a) unstructured (b) structured.

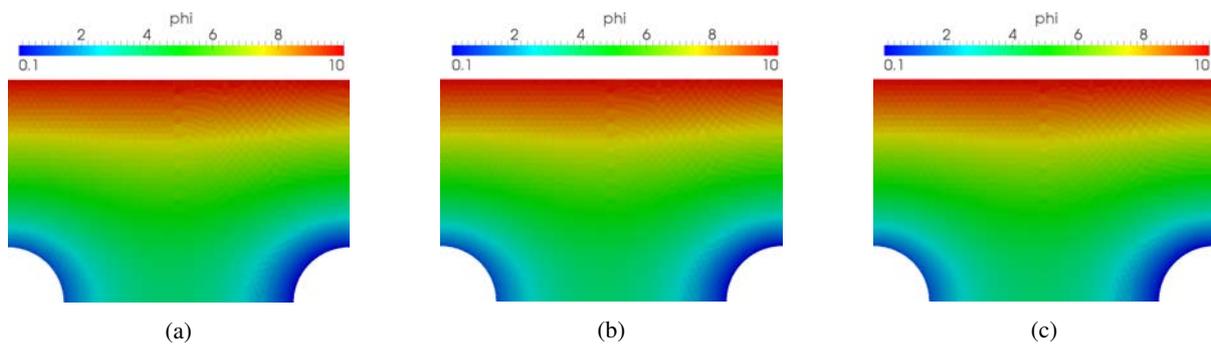


Figure 13: Contour plot of the solution variable on the unstructured mixed mesh: (a) Own Code, (b) OpenFOAM 2.1.1 and (c) Ansys Fluent 13.0 solutions.

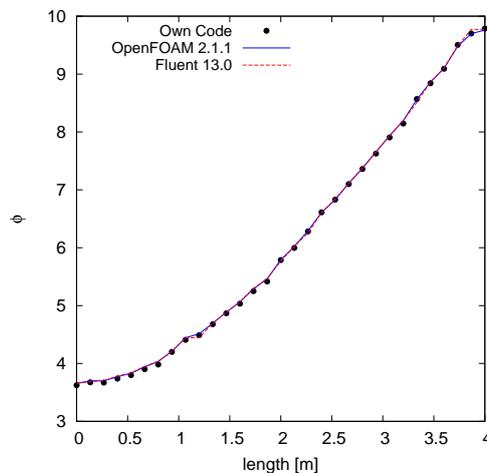


Figure 14: Solution variable profile along the straight vertical line that joint the points $(3.05, 0, 0.05) \times (3.05, 4, 0.05)$.

Code, OpenFOAM 2.1.1 and Ansys Fluent 13.0. Again, all ϕ fields are in good agreement. Figure 14 shows the graphics of the solution variable along the straight vertical line that joint the points $(3.05, 0, 0.05) \times (3.05, 4, 0.05)$, confirming the validity of our implementation.

Figures 15 and 16 depict the contour plot of the x-component and y-component of the gradient, respectively, for the simulations carried out using our Own Code on the unstructured, taking account the skewness in the non-orthogonality correction, and using OpenFOAM 2.1.1 on the structured and unstructured meshes with its procedure of non-orthogonality correction. The effect of the skewness correction is quite relevant and can be easily observed. Figure 17 shows the profiles of the x and y-component of the gradient along the straight vertical line described above, obtained in these simulations. Again, the effect of the skewness correction is apparent.

Figures 18 and 19 show, respectively, the x and y components of $\nabla\phi$ obtained by our Own Code with skewness correction taken account in the non-orthogonality correction procedure and by OpenFOAM 2.1.1 with its non-orthogonality correction procedure, both using the structured mesh. The skewness correction improves the result on the regions of higher skewness of this refined mesh.

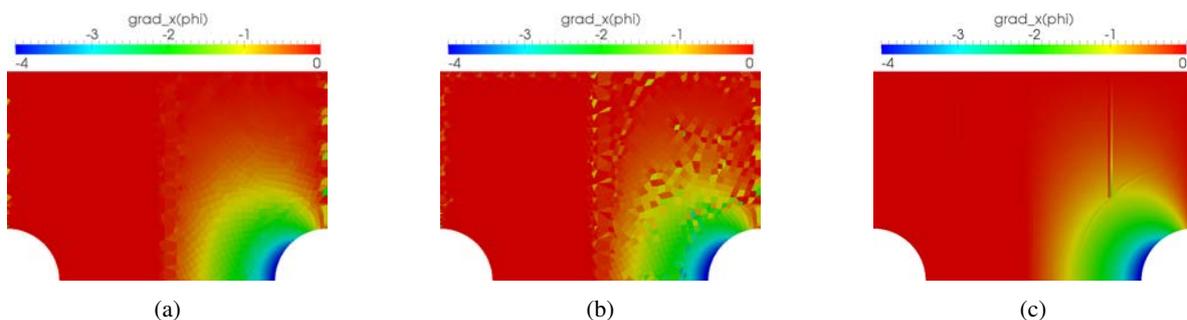


Figure 15: Contour plot of the x-component of the solution variable gradient on the unstructured mesh: (a) Own Code, (b) OpenFOAM 2.1.1. (c) Benchmark result on the structured mesh obtained with OpenFOAM 2.1.1.

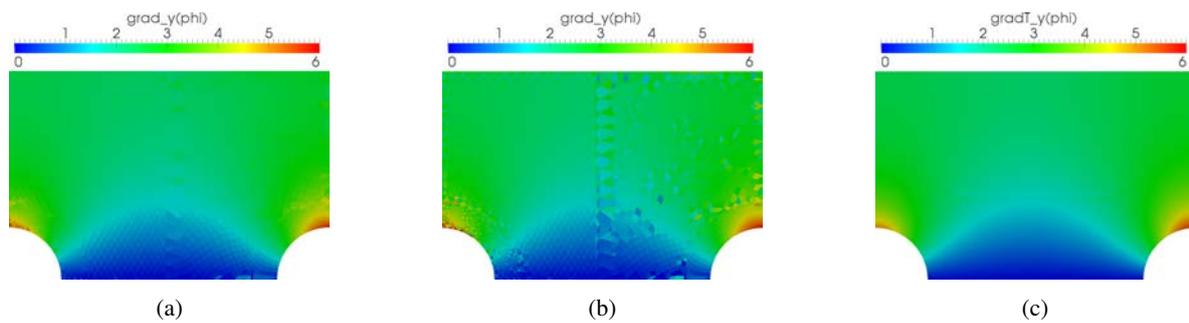


Figure 16: Contour plot of the y-component of the solution variable gradient on the unstructured mesh: (a) Own Code, (b) OpenFOAM 2.1.1. (c) Benchmark result on the structured mesh obtained with OpenFOAM 2.1.1.

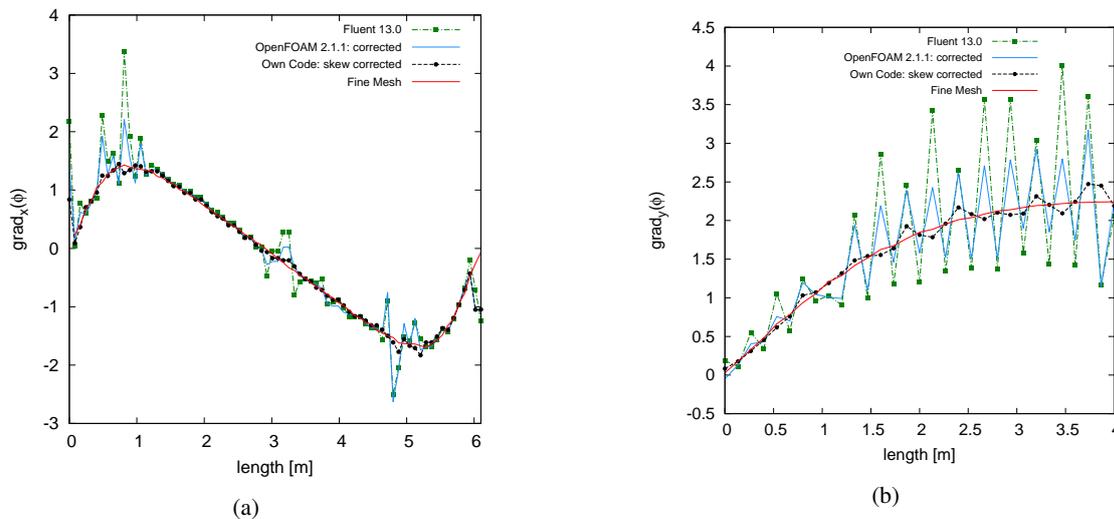


Figure 17: Profiles of the solution variable gradient along the straight vertical line that joint the points $(3.05, 0, 0.05)$ and $(3.05, 4, 0.05)$. (a) x-component and (b) y-component.

For all the simulated cases, the solution variable ϕ was not sensible to the skewness correction. Both OpenFOAM and our Own Code provided almost the same results, as can be seen in Fig. 14. This was expected because, it can be demonstrated that the contribution of the skew correction to the solution of diffusive problems is of the order of magnitude $\nabla[(\nabla\phi)_f \cdot \mathbf{m}] \cdot \mathbf{k}$, which is small for relatively smooth ϕ solutions on meshes of average quality.

5. Conclusion

In the present work, we implemented an in-house finite volume code with a non-orthogonality correction approach that take into account the mesh skewness. This code was used to analyse the effects of the mesh skewness on the numerical solution of a purely diffusive model. This code was validated against OpenFOAM 2.1.1 and ANSYS Fluent 13.0 by solving of a diffusive problem on a non-orthogonal but skewness free structured mesh.

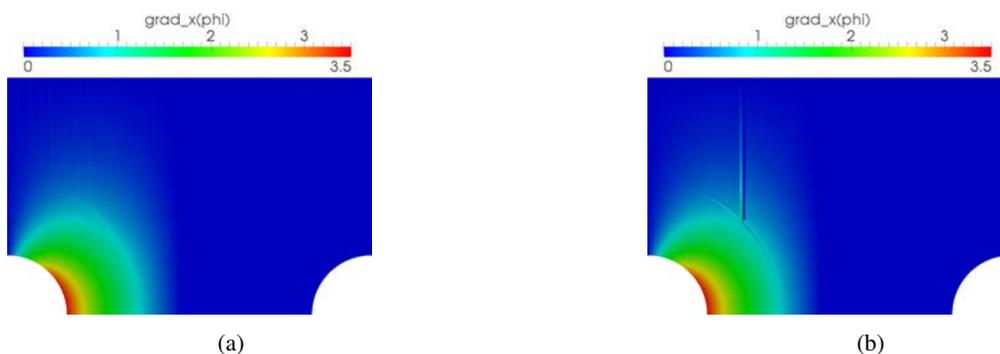


Figure 18: Contour plot of the x-component of the gradient on the structured mesh, obtained using (a) our Own Code and (b) OpenFOAM 2.1.1. Only gradient values in the $[0, 3.5]$ range were plotted.

Antonio de O. S. Moraes, Paulo L. C. Lage, Guilherme C. Cunha and Luiz Fernando L. R. Silva
 Analysis of the Non-Orthogonality Correction of Finite Volume Discretization on Unstructured Meshes

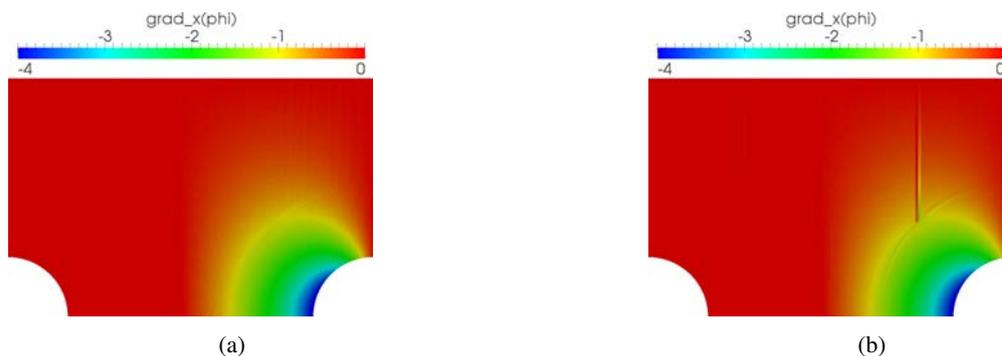


Figure 19: Contour plot of the x-component of the gradient on the structured mesh, obtained using (a) our Own Code and (b) OpenFOAM 2.1.1. Only gradient values in the $[-4,0]$ range were plotted

For meshes with high values of skewness, the Eqs. 21 and 23 showed that, in order not to deteriorate the second order accuracy of the numerical method, the *skewness correction* has to be taken into account in the interpolation of the solution variable and its gradients to the faces. The numerical results obtained in a second test problem showed that the skewness correction can have a non-negligible effect. More precisely, when the results obtained for an unstructured mesh with high skewness values were compared with those obtained for a fine structured mesh, the accuracy of the gradient values at the centers of the control volumes was significantly increased when the proposed skewness correction was taken into account. We observed, however, that the implemented code does not yet take into account the skewness correction in the gradient interpolation to the faces.

6. ACKNOWLEDGEMENTS

Antonio de O. S. Moraes acknowledges the financial support from CNPq, grants no. 140951/2012-1. Paulo L. C. Lage acknowledges the financial support from CNPq, grants nos. 302963/2011-1 and 478589/2011-5, and FAPERJ, grant no. E-26/111.361/2010.

7. REFERENCES

- Boris, J.P. and Book, D.L., 1973. "Flux-corrected transport. i. shasta, a fluid transport algorithm that works". *J. Comp. Physics*, pp. 38–69.
- C.R. Rao, H. Toutenburg, A.F.C.H.T.N. and Scheid, S., 1999. *Linear Models: Least Squares and Alternatives*. Springer Series in Statistics.
- Cunha, G. and Redonnet, S., 2012. "On the signal degradation induced by interpolation and the sampling rate reduction in aeroacoustics hybrid methods". *International Journal for Numerical Methods in Fluids*.
- Cuthill, E. and McKee, J., 1969. "Reducing the bandwidth of sparse symmetric matrices". *Proceedings of the 1969 24th National Conference*, pp. 157–172.
- Ferziger, J.H. and Perić, M., 1997. *Computational Methods for Fluid Dynamics*. Springer, Germany, 2nd edition.
- FLUENT Inc., 2006. "Tgrid 4.0 user's guide". URL http://www.tchpc.tcd.ie/fluent/Unpacked_IS0s/TGrid_4.0_Documentation/tgrid4.0/help/html/ug/node318.htm.
- Hirsch, C., 1991. *Numerical computation of internal and external flows*. John Wiley & Sons.
- Jasak, H., 1996. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Ph.D. thesis, Imperial College of Science, Technology and Medicine, London.
- Leonard, B.P., 1979. "A stable and accurate convective modelling procedure based on quadratic upstream interpolation". *Comp. Meth. Appl. Mech. Engineering*, pp. 59–98.
- Maliska, C.R., 1995. *Transferência de Calor e Mecânica dos Fluidos Computacional*. LTC, Rio de Janeiro, 1st edition.
- Patankar, S.V., 1981. *Numerical heat transfer and fluid flow*. Hemisphere Publishing Corporation.
- Salome, 2013. "The open source integration platform for numerical simulation". URL <http://www.salome-platform.org/>.
- Weller, H.G., Tabor, G., Jasak, H. and Fureby, C., 1998. "A tensorial approach to computational continuum mechanics using object-oriented techniques". *American Institute of Physics*.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.