Journal of Computational Physics 228 (2009) 5148–5159

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A robust and efficient finite volume scheme for the discretization of diffusive flux on extremely skewed meshes in complex geometries

Philippe Traoré [a,*], Yves Marcel Ahipo [b], Christophe Louste [a]

[a] Laboratoire d'Etudes Aérodynamiques, Université de Poitiers, Bd Marie et Pierre Curie, BP 30179, Téléport 2, Chasseneuil Cedex, France
[b] Laboratoire de Mathématiques pour l'Industrie et la Physique, Université Paul Sabatier Toulouse 3, 31062 Toulouse, Cedex 9, France

ARTICLE INFO

Article history:
Received 16 May 2008
Received in revised form 3 April 2009
Accepted 7 April 2009
Available online 18 April 2009

Keywords:
Finite volume
Diffusive flux discretization
Poisson equation
Deferred correction
Skewed meshes
Distorted grid

ABSTRACT

In this paper an improved finite volume scheme to discretize diffusive flux on a non-orthogonal mesh is proposed. This approach, based on an iterative technique initially suggested by Khosla [P.K. Khosla, S.G. Rubin, A diagonally dominant second-order accurate implicit scheme, Computers and Fluids 2 (1974) 207–209] and known as deferred correction, has been intensively utilized by Muzaferija [S. Muzaferija, Adaptative finite volume method for flow prediction using unstructured meshes and multigrid approach, Ph.D. Thesis, Imperial College, 1994] and later Fergizer and Peric [J.H. Fergizer, M. Peric, Computational Methods for Fluid Dynamics, Springer, 2002] to deal with the non-orthogonality of the control volumes. Using a more suitable decomposition of the normal gradient, our scheme gives accurate solutions in geometries where the basic idea of Muzaferija fails. First the performances of both schemes are compared for a Poisson problem solved in quadrangular domains where control volumes are increasingly skewed in order to test their robustness and efficiency. It is shown that convergence properties and the accuracy order of the solution are not degraded even on extremely skewed mesh. Next, the very stable behavior of the method is successfully demonstrated on a randomly distorted grid as well as on an anisotropically distorted one. Finally we compare the solution obtained for quadrilateral control volumes to the ones obtained with a finite element code and with an unstructured version of our finite volume code for triangular control volumes. No differences can be observed between the different solutions, which demonstrates the effectiveness of our approach.

1. Introduction

When we are faced with domains with complex geometry, which is often the case in industrial configurations, we have to deal with control volumes with arbitrary shapes or at least non-orthogonal control volumes. Elements which are triangular in 2D or tetrahedral in 3D, leading to an unstructured grid, are often used because of their ability to mesh any kind of geometry. But computational development can become more critical regarding data structure complexity and bookkeeping. Moreover the structure of the outcome discretization matrix generally leads to additional computational time requirements to solve the related linear system. Because of these drawbacks, some people prefer structured or at least block-structured grids of control volumes which are quadrilateral in 2D or hexahedral in 3D. The weakness of this alternative is that quadrilateral or hexahedral control volumes cannot fit any kind of complex geometry without introducing non-orthogonal control volumes. In most cases, this non-orthogonality could lead to a strongly skewed mesh, that is, mesh made of control volumes with very

* Corresponding author. Tel.: +33 5 49 49 69 30; fax: +33 5 49 49 69 68.
E-mail address: philippe.traore@univ-poitiers.fr (Ph. Traoré).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.007

obtuse internal angles. A common idea propagated in the CFD community asserts that the accuracy order of numerical schemes degrades as soon as control volumes are not orthogonally quadrilateral. This idea, which is shared by a great majority of authors, comes from the fact that effectively skewed control volumes involve consistency problems in the discretization of second derivative or diffusive flux F^d through the surface S_P surrounding a given control volume V_P:

$$F^d = \int_{S_P} (\nabla\phi)\cdot\vec{n}\,ds \tag{1}$$

$S_P = \bigcup_{k=neighbours} S_k^P$, where S_k^P represents the surface of the face k and $\vec{n}_k^P$ is the unit vector normal to this face relative to the control volume V_P. $\nabla\phi$ is the gradient of the scalar function ϕ. For simplicity of graphical representation we shall reason with 2D quadrilateral control volumes, but our developments remain valid for control volumes of arbitrary shape in 3D.

Using the middle point rule in the approximation of (1) leads to:

$$F^d \simeq \sum_{k=n,e,n,w,s} S_k^P (\nabla\phi)_k^P \cdot \vec{n}_k^P \tag{2}$$

The key problem of the diffusive flux discretization in complex geometry reduces then to the search for a consistent approximation of the gradient of ϕ in the $\vec{n}_k^P$ direction, that is $(\nabla\phi)_k^P \cdot \vec{n}_k^P$, on each face S_k^P of the control volume V_P. It is possible to design particular meshes in such a way that $\vec{n}_k^P$ is always parallel to $\vec{\xi}_k^P$, the unit vector of the line joining the two nodes located on both sides of the face S_k^P (see Fig. 1). These meshes are called *admissible* meshes and were proposed by Herbin [4,5]. The advantage of such meshes is that they allow the use of simple schemes for the discretization of $(\nabla\phi)_k^P \cdot \vec{n}_k^P$. However, in practice it is not always feasible to build such *admissible* meshes because they imply strong constraints on algorithms dedicated to such a task and generally, meshing software does not have this interesting feature. So authors who have worked so far with finite volume discretization in a general grid context [6–9] have proposed formulations to approach $(\nabla\phi)_k^P \cdot \vec{n}_k^P$ which involved not only the centers of the control volumes but also the vertices of each corner: *nw, ne, se, sw* (see Fig. 1).

The drawback of these approaches is the use of additional variables, which increases the discretization stencil and thus the complexity of the discretization matrix. The solution of such a resulting linear system would become trickier or more expensive in terms of computational time. These difficulties could be avoided by using a strategy well-known in the CFD community proposed by Muzaferija [2] and spread by Peric [3,10–12], based on the technique of *deferred correction* introduced by Khosla [1].

In this context, deferred correction is used as a simple and efficient iterative method to approximate diffusive flux especially in the solution of the Navier–Stokes equations on non-orthogonal grids [3,11,12,14]. This technique can also be found in the most popular numerical codes in fluid mechanics like Star-CD or Fluent.

The approximation of $(\nabla\phi)_k^P.\vec{n}_k^P$ will be achieved through an iterative process detailed in [2,3,13,14] which must converge towards the desired quantity:

$$\begin{aligned} &(\nabla\phi)_k^m \cdot \vec{n}_k^P \to (\nabla\phi)_k^P \cdot \vec{n}_k^P \\ &m \to \infty \end{aligned} \tag{3}$$

Certainly because of the loss of accuracy or potential divergence of the solution in the case of strongly skewed meshes, some authors (Muzaferija [2], and guidelines in user manuals of industrial numerical codes) have suggested some restrictions before using this method. The main restriction is the requirement for an internal angle that is not too obtuse, namely > 45°, which is not systematically feasible in cases of very complex geometry, especially when using quadrilateral control volumes. As the usual approach leads to convergence problems in the case of severe non-orthogonality, the aim of our study is to provide a modified version of the deferred correction method which is more robust and more efficient.

The structure of the paper is as follows: In Section 2, the mathematical formulation will be introduced, including in paragraph 2.1 the presentation of the *Standard Deferred Correction* statements, which will be abbreviated for simplicity to the acronym *SDC*. In paragraph 2.2 we will highlight the basis of our new scheme. The implementation details and the

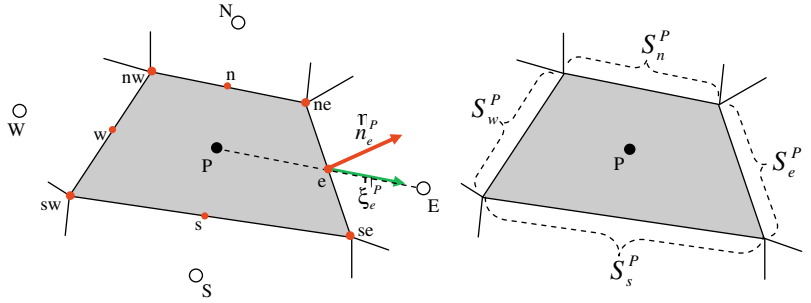

Fig. 1. Notations for a control volume P and its neighbours. Surfaces around the control volume P.

discretization of the involved operators will be presented in Section 2.3. In Section 3, three different test cases and examples of applications will be performed in order to establish the efficiency, robustness and advantageousness of our new scheme, which will be referred to as *Improved Deferred Correction* and abbreviated to *IDC*.

2. Mathematical formulation

Solving a Poisson problem allows us to build an analytical solution which will be used to test the accuracy of the numerical one.

Poisson equations with homogeneous Dirichlet boundary conditions in a domain (Ω) are formulated as:

$$\Delta\phi = g \quad \text{in } (\Omega) \tag{4}$$

$$\phi = f \quad \text{on } (\partial\Omega) \tag{5}$$

where ϕ is the unknown function and g and f are two given functions. Eq. (4) is integrated over a control volume P of volume V_P. Expressing Eq. (4) in its divergence form and using the Green–Ostrogradsky theorem [3,15], we obtain:

$$\int_{S_P} (\nabla\phi)\cdot\vec{n}\,ds = \int_{V_P} g\,dv \tag{6}$$

Using the middle point rule, which is a second order quadrature formula, Eq. (6) can be written for a given control volume, with the notation of Fig. 1, in the following form:

$$\sum_{k=n,e,w,s} S_k^P (\nabla\phi)_k^P \cdot \vec{n}_k^P = (g)_P \cdot V_P \tag{7}$$

For clarity in the following sections, the index P which refers to "control volume P" will now be dropped.

2.1. The Standard Deferred Correction (SDC)

The deferred correction approach as proposed by [2,3] consists in noticing that $(\nabla\phi)_k \cdot \vec{n}_k = (\nabla\phi)_k \cdot (vecn_k - \vec{\xi}_k) + (\nabla\phi)_k \cdot \vec{\xi}_k$.

Muzaferija [2] and Fergizer and Peric [3] introduce an iterative process such as:

$$\sum_{k=n,e,w,s} S_k (\nabla\phi)_k^m \cdot \vec{\xi}_k = (g)_P . V_P - \sum_{k=n,e,w,s} S_k (\nabla\phi)_k^{m-1} \cdot \left(\vec{n}_k - \vec{\xi}_k\right) \tag{8}$$

The left hand side of (8) can be easily approximated by a centered formula. For example, using a second order centered scheme should lead to:

$$S_k (\nabla\phi)_k^m \cdot \vec{\xi}_k = d_k\left(\phi_K^m - \phi_P^m\right) \tag{9}$$

with $d_k = S_k/d(P,K)$, where $d(P,K)$ is the distance between P and K; ϕ_K^m and ϕ_P^m, respectively, are the mth iterative value of ϕ at nodes K and P, the centers of two neighboring control volumes.

The right hand side can be explicitly computed as it depends only on terms from the previous iteration. This can be achieved using the Gauss formula, least-squares method or any shape function.

Numerical tests have shown that the former deferred correction approach gives fairly good results only when θ_k (see Fig. 2) is small, and diverges otherwise. Then, it appears that the convergence of the deferred correction is closely linked to the skewness of the control volumes. Since the initial approach of SDC fails in cases where $\theta_k > 38°$, is it possible to improve the scheme and provide solutions especially in strongly skewed meshes?

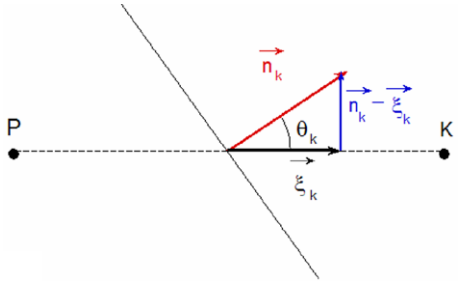

Fig. 2. $\vec{n}_k$ decomposition in SDC.

2.2. Proposition of an Improved Deferred Correction (IDC) scheme

If $\vec{t}_k$ is the unit vector parallel to a face S_k, then the normal $\vec{n}_k$ to this face can be expressed as:

$$\vec{n}_k = \lambda_k \vec{\xi}_k + \beta_k \vec{t}_k \tag{10}$$

where $\lambda_k = \frac{1}{\cos\theta_k}$ and $\beta_k = \tan\theta_k$

Note that $\cos\theta_k$ is the projection of the cell face normal $\vec{n}_k$ onto $\vec{\xi}_k$ and the projection of $\vec{t}_k$ onto $\vec{\xi}_k$ is $-\sin\theta_k$. Hence $\vec{\xi}_k = \cos\theta_k \vec{n}_k - \sin\theta_k \vec{t}_k$ (see Fig. 3).

With these considerations, Eq. (7) becomes:

$$\sum_{k=n,e,w,s} \lambda_k S_k (\nabla\phi)_k \cdot \vec{\xi}_k = (g)_P \cdot V_P - \sum_{k=n,e,w,s} \beta_k S_k (\nabla\phi)_k . \vec{t}_k \tag{11}$$

and then we can introduce the IDC process such as:

$$\sum_{k=n,e,w,s} \lambda_k S_k (\nabla\phi)_k^m \cdot \vec{\xi}_k = (g)_P \cdot V_P - \sum_{k=n,e,w,s} \beta_k S_k (\nabla\phi)_k^{m-1} \cdot \vec{t}_k \tag{12}$$

2.3. Implementation details

The left hand side of (12) is still approximated by a centered formula of second order.

$$\lambda_k S_k (\nabla\phi)_k^m \cdot \vec{\xi}_k = \frac{\lambda_k S_k (\phi_K^m - \phi_P^m)}{d(P,K)} \tag{13}$$

The second term of the right hand side is explicitly determined from the gradient at nodal values of the previous iteration interpolated at the k location:

$$(\nabla\phi)_k^{m-1} = (1-\omega_K)(\nabla\phi)_P^{m-1} + \omega_K (\nabla\phi)_K^{m-1} \tag{14}$$

where ω_K is the linear interpolation factor between the two neighbors node P and K.

In this study, the gradient at nodal values is computed with the Gauss formula:

$$(\nabla\phi)_P^j \approx \frac{\sum_{k=n,e,w,s} S_k \phi_k (\vec{n}_k \cdot \vec{i}_j)}{V_P} \tag{15}$$

where $\vec{i}_j$ is the unit vector in the jth cartesian coordinate and ϕ_k are the interpolated nodal values of ϕ at the center k of face S_k.

$$\phi_k = (1-\omega_K)\phi_P + \omega_K \phi_K \tag{16}$$

In accordance with globally accepted finite volume notations [16], if we define the generic equation as $A_P \phi_P^m + \sum_{K=NEWS} A_K \phi_K^m = (RHS)_P$, we would have the following coefficients:

$$\begin{aligned} A_K &= \frac{\lambda_k S_k}{d(P,K)} \\ A_P &= -\sum_{K=NEWS} A_K \\ (RHS)_P &= (g)_P V_P - \sum_{k=news} \beta_k S_k (\nabla\phi)_k^{m-1} \cdot \vec{t}_k \end{aligned} \tag{17}$$

Eq. (12), written for every control volume defining the calculation domain, leads to a linear system $A\vec{x} = \vec{b}$ that we solve with the Bi-Conjugate Gradient Stabilized (Bi-CGSTAB) of Van der Vorst [17] with a Modified Incomplete LU factorization preconditioner according to Gustafsson [18]. So, the resolution process is made of two major loops. The first loop, which

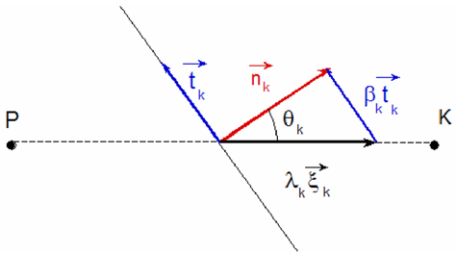

Fig. 3. $\vec{n}_k$ decomposition in IDC.

we call *sweep iterations*, is associated with the m index of the deferred correction in Eq. (12). The second loop, which we call the *inner loop*, is associated this time with the iterative nature of the solver. For each m index of the sweep iterations we have to solve the linear system, which includes the inner loop.

Two different stopping criteria should be defined to stop the iterative processes. For the m loop we shall iterate until the quantity $\frac{|\phi_P^m - \phi_P^{m-1}|}{|\phi_P^{m-1}|} > \varepsilon_{sweep}$ for all nodes P of the domain. For loop k of the linear system solver we use the classical stopping criteria based on the residual which should be less than a given tolerance ε_{solver}.

To improve the performances in terms of CPU time consuming, parallelization is often suggested. Although the parallelization of our algorithm has not been carried out in the present paper, we should nevertheless mention that nothing stands opposed to the development of a full parallel version of our approach. Indeed, as in such situations of Poisson problems, parallelization mainly hinges on the linear system solver; thus, if the solver used is parallelizable then the full method will be parallelizable too. The solver BI-CGSTAB used in this study has been successfully parallelized in other contexts [19,20].

3. Numerical experiments

3.1. Performance comparison tests between SDC and IDC

For this first test we consider the domain depicted in Fig. 4, whose skewness is defined by angle θ.
We solve the following Poisson equation:

$$\Delta\phi = 6x + 2 \quad \text{in } (\Omega) \tag{18}$$

$$\phi(x,y) = x^3 + y^2 + xy \quad \text{on } (\partial\Omega) \tag{19}$$

The exact solution is given by $\phi(x,y) = x^3 + y^2 + xy$. The grid will be skewed successively by an angle θ increasing from 20° to 89.9°, which is roughly acute. Typically in commercial CFD codes it is recommended that a value of 45° should not be exceeded. We shall compare the convergence of the two methods, the former deferred correction, SDC, and our new scheme, IDC. Using the software Matlab we can compute, over a given grid, the spectral radius of each iteration matrix for the both approaches. Then it is possible to study their sensitivity to the grid skewness defined by θ. In Table 1, it can be seen that the spectral radius of the iteration matrix of SDC method tends to 1 when θ comes close to 38°. It still increases when θ exceeds 38°, which will cause the divergence of the computation. At the same time the IDC iteration matrix keeps a spectral radius that is always lower than 1 whatever the considered skewness. It is clearly apparent that the IDC scheme converges whatever the value of θ, whereas the SDC approach converges only for values of θ lower than 38°, which indicates its limitation.

This result is confirmed in Table 2, where the values of the $L2$ norm versus angle θ have been reported for each scheme. Whatever the value of θ, our approach converges and provides solutions with a satisfactory error. For the SDC scheme, "–" means that the code has diverged.

We have computed the evolution of the ratio T [15], defined as: $T = \frac{\ln\left(\frac{err_1}{err_2}\right)}{\ln 2}$, versus θ, where err_1 and err_2 denote the discrete $L2$ errors made respectively on Grid1 and Grid2, two successive grids; Grid2 has twice as many control volumes as Grid1 in each direction of meshing. T is a good estimation of the accuracy order of the scheme. We considered two cases:

Case 1: Grid1: 20×20 and Grid2: 40×40
Case 2: Grid1: 40×40 and Grid2: 80×80

The results in Table 3 reveal that for θ lower than 38°, the solutions obtained by each of the two methods both have the expected second order. Beyond 38°, this second order is preserved with the IDC scheme whereas the former approach, SDC, diverges dramatically From these results, it is clear that the accuracy order of the method is not damaged as the skewness is increased.

In Table 4, we have reported the number of sweeps required in the deferred correction process versus θ. The tolerance for the stopping criterion is intentionally very severe and set to $\varepsilon_1 = 1 \cdot e^{-7}$. The grid is made of 40×40 control volumes. We still observe the limit of $\theta = 38°$over which the SDC scheme diverges. Even for the most extreme angle of 89.9° the number of sweeps required for our scheme is four times less than the number of sweeps required for the SDC scheme in the limit case, $\theta = 38°$.

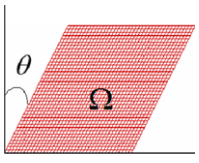

Fig. 4. Computational domain and grid.

Table 1
Spectral radius of the iteration matrix versus angle θ for a 40×40 grid.

θ	20°	30°	38°	50°	60°	70°	75°
ρ_{SDC}	0.46	0.71	0.99	1.7	2.6	4.27	5.88
ρ_{IDC}	0.35	0.48	0.58	0.7	0.78	0.82	0.82

Table 2
$L2$ norm versus angle θ for a 40×40 grid.

θ	20°	30°	40°	50°	60°	70°	75°
$L2_{SDC}$	2.71 e^{-05}	8·e^{-05}	–	–	–	–	–
$L2_{IDC}$	2.64 e^{-05}	7.09 e^{-05}	1.01 e^{-04}	2.25 e^{-04}	4.89 e^{-04}	1.26 e^{-03}	2.48 e^{-03}

Table 3
Evolution of T versus angle θ for two cases.

θ	20°	30°	38°	50°	60°	70°	75°
Case 1							
T_{SDC}	1.98	1.988	1.982	–	–	–	–
T_{IDC}	1.988	1.988	1.982	1.984	1.987	1.99	1.991
Case 2							
T_{SDC}	1.991	1.9	1.984	–	–	–	–
T_{IDC}	1.988	1.991	1.991	2.006	1.994	1.994	1.994

Table 4
Number of sweeps versus θ.

θ	20°	30°	38°	45°	50°	60°	70°	80°	85°	89.9°
Sweep $_{SDC}$	14	30	872	–	–	–	–	–	–	–
Sweep $_{IDC}$	12	17	23	30	37	57	91	9	161	169

Table 5
Total number of iterations versus θ.

θ	20°	30°	38°	45°	50°	60°	70°	80°	85°	89.9°
It_{SDC}	7	303	7764	–	–	–	–	–	–	–
It_{IDC}	121	173	236	304	371	560	872	09	1501	1575

Table 6
CPU time needed for the schemes (s).

θ	SDC	IDC
10°	0.036 s	0.032 s
35°	0.14 s	0.068 s
89.5°	–	0.32 s

In Table 5, we have pointed out the total number of iterations versus angle θ for the Bi-CGSTAB linear system solver and for the two schemes: SDC and IDC. The total number of iterations is the sum of iterations of the k loop for each sweep of the m loop.

In Table 6, we have reported the CPU time for the two schemes and for three different angles. The difference is not very sensitive for small θ angles, but increases when θ gets closer to the limit of 38°. For $\theta = 35°$, the CPU time consumption of SDC is around 2.5 times higher than that of IDC for the same 40×40 grid.

Comparing the two schemes, SDC is obviously much more demanding in terms of iterations and thus requires more computing time than IDC. These results show that the skewness of the control volumes has no impact on the accuracy order of the solution itself, since the convergence criteria are satisfied, but only on the number of iterations and sweeps.

In Fig. 5, we have plotted the evolution of a monitored point solution versus the number of sweeps for the two schemes in the previous limit case of $\theta = 38°$. It is remarkable that in the case of the SDC scheme, the solution oscillates very much before

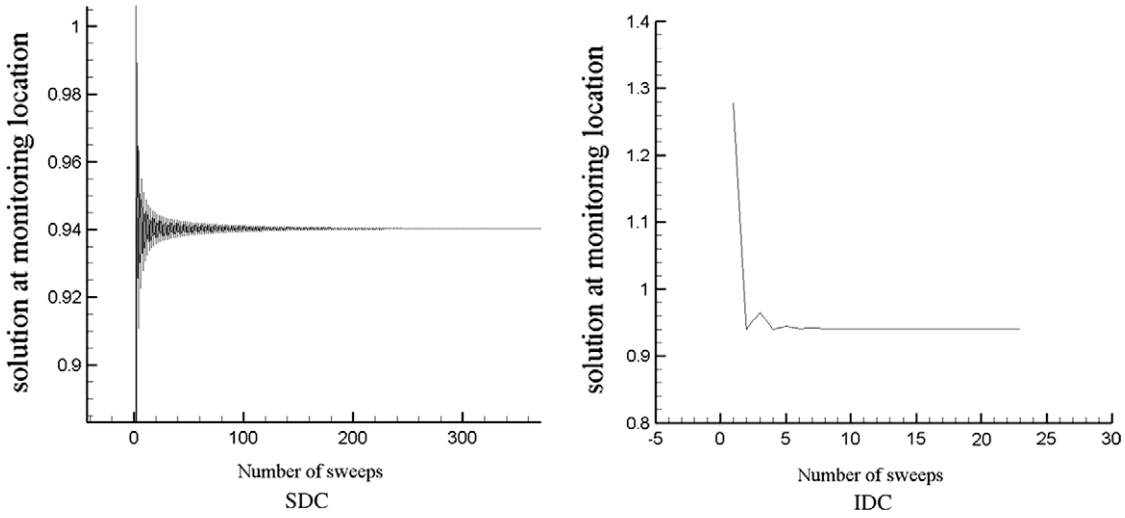

Fig. 5. Evolution of a monitored point solution versus the number of sweeps for $\theta = 38°$.

reaching the asymptotic value. In contrast, with IDC we have only two periods of oscillation before obtaining the stabilized solution below the tolerance of $1{\cdot}e^{-7}$. Please note also that the solution is obtained for less than 25 sweeps in our case while it still oscillates after 400 sweeps in the case of the SDC scheme. The criteria will in fact be satisfied after 872 sweeps.

3.2. Random mesh and severe grid distortion tests

With the goal of confronting the IDC scheme with stiffest configurations we have designed two special, more drastic grids as test cases. Following the procedure of Shashkov and Steinberg [21] we have generated a randomly distorted grid and another that is particularly skewed in anisotropic directions as depicted in Fig. 6. The orthogonal grid will serve as a reference case. The objective is not only to compare the solution obtained on these two grids with the orthogonal one but also to compare the conditions in which those solutions are obtained.

We intend to solve the following Poisson equation:

$$\Delta\phi = 2\pi^2(\cos^2(\pi x) - \sin^2(\pi x)) + 2\pi^2(\cos^2(\pi y) - \sin^2(\pi y)) \quad \text{in } (\Omega) \tag{20}$$

$$\phi = \sin^2(\pi x) + \sin^2(\pi y) \quad \text{on } (\partial\Omega) \tag{21}$$

$\Omega$is then a square domain $[1 \times 1]$ that we have meshed with the above three grids. These three grids: orthogonal, randomly distorted and anisotropically skewed, each have 2601 control volumes (51 control volumes in each direction). The analytical solution of this problem is the following surface shown in Fig. 7, which has been chosen to be not too smooth, to stiffen the problem. In this figure, z represents the magnitude of this analytical solution.

We have to mention that for the previous test case, a *node centered* version of our code was used. With these particular grids and especially the third one, due to the anisotropy of the mesh distortion, the *node centered* configuration does not guarantee that the node P (supposed to be the cell center) is always inside the cell itself (see Fig. 8). To avoid these pathologic

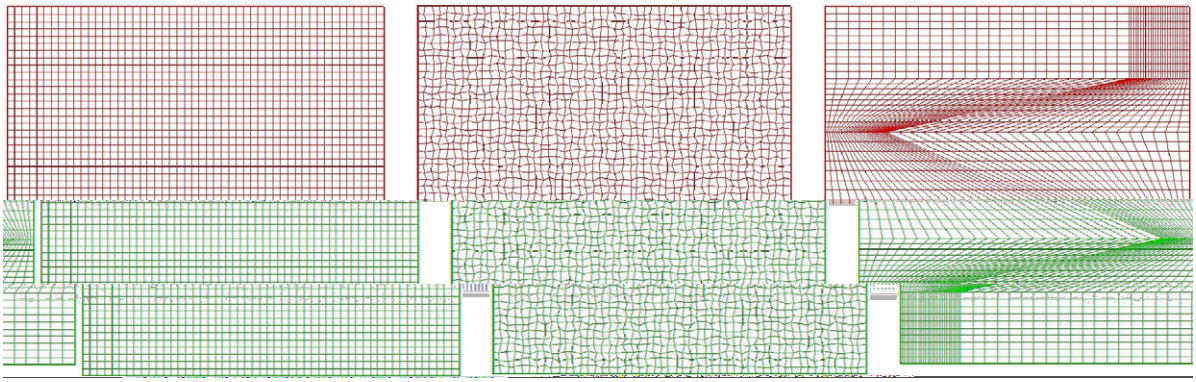

Fig. 6. Different meshes used for the second test case.

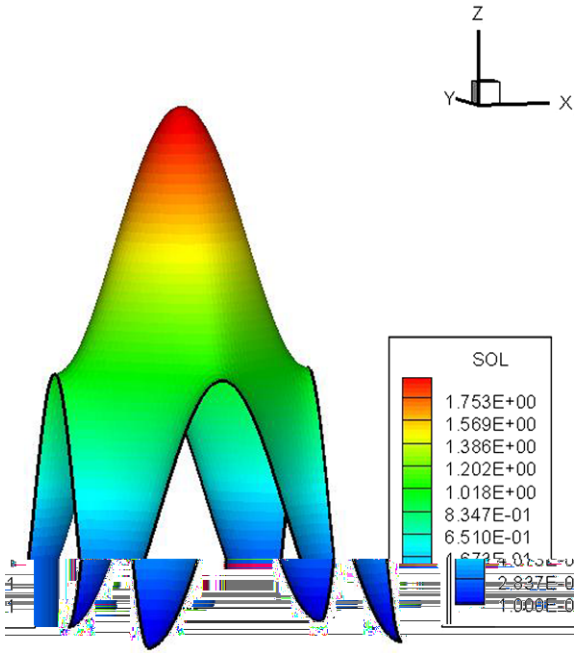

Fig. 7. Analytical solution.

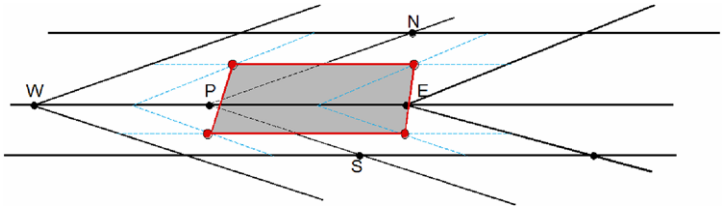

Fig. 8. Pathologic configuration where the node *P* could be outside its control volume.

situations typical of *node centered* configurations, we had to implement a *cell centered* version of the code as for this configuration, these situations cannot occur.

The convergence conditions are summarized in Table 7.

The more obtuse the internal angles, the more iterations will be required in the iterative process. In Fig. 9 we have presented the absolute error made on the three successive grids. As expected the anisotropically skewed mesh produces the highest error with a maximum of 9.25% relative error while the randomly distorted mesh gives fairly good results with only 0.95% relative error.

The difference in terms of number of sweeps, iterations and accuracy between the orthogonal grid and the others is emphasized for the anisotropically distorted one. These differences are, according to us, due more to the anisotropy of the grid itself than to its skewness. Indeed we can observe that the errors are magnified in regions where the grid is strongly anisotropic, that is, where the angle between two adjacent cell faces of two contiguous cells is changing drastically. It is more this factor which contributes to the stiffness of the problem and which plays a role in the discrepancy between the different results, rather than the skewness itself.

Table 7
Total number of iterations, sweeps, CPU time and errors for the three tested grids.

	Orthogonal	Randomly distorted	Anisotropically distorted
Total number of iterations	20	104	35
Total number of sweeps	2	8	82
CPU time	0.068 s	0.088 s	0.44 s
$L2$ norm	1.34 e^{-05}	1.69 e^{-04}	1.1 e^{-03}
Max error	9.5e^{-04}	1.9 e^{-02}	1.85 e^{-01}

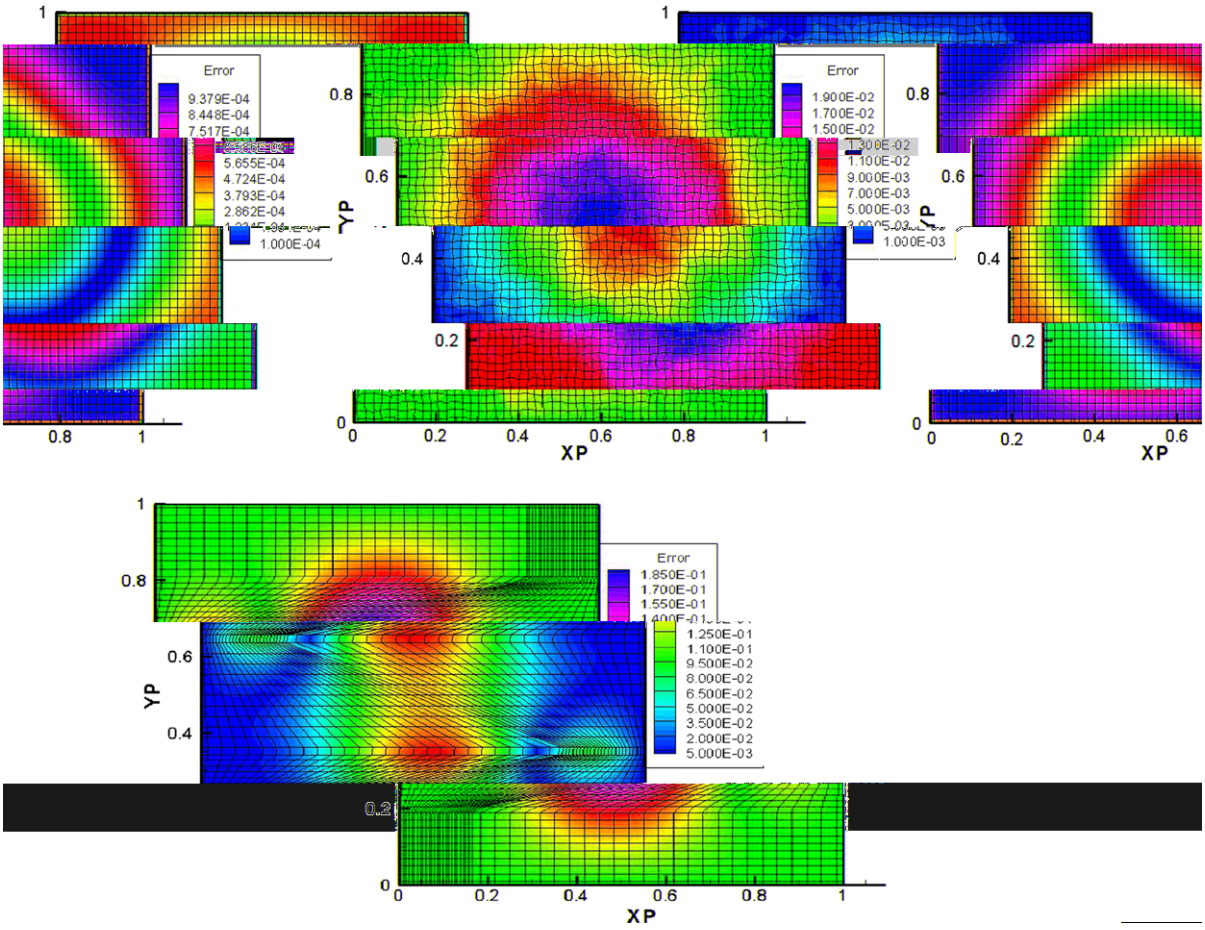

Fig. 9. Iso-contours of the absolute error made on the three grids.

It should be noted that in the case of the randomly and anisotropically distorted meshes, refining these grids will reduce these errors, as the scheme has been found to keep its theoretical second order of accuracy. In Table 8, the *L*2 Norm and maximum error for different refined grids of the randomly distorted one are reported.

We computed the *T* ratio for three pairs of grids (Grid0/Grid1, Grid1/Grid2 and Grid2/Grid3). We obtained: $T_{Grid0/Grid1} = 1.69$, $T_{Grid1/Grid2} = 1.9$ and $T_{Grid2/Grid3} = 1.96$.

Therefore the use of orthogonal control volumes instead of skewed ones will have an impact only on the number of iterations needed for the deferred correction but not on the accuracy order itself, which tends to 2: the expected theoretical order.

Despite the fact that our scheme preserves the second order of accuracy, the discrepancy between the results obtained for the different grids has another source. In a rather distorted grid another pathologic configuration as depicted in Fig. 10 can occur. The line connecting the two nodes *P* and *K* crosses the cell face not at its midpoint "*k*", where the diffusive flux ought to be calculated, but at another location *k*′. This pathologic situation, which is this time typical of *cell centered* configuration, will have consequences for the overall accuracy. Indeed, for any interpolated quantities ψ, $(1\text{-}\omega_K)\psi_P + \omega_K\psi_K = \psi_{k'} \neq \psi_k$. In some extreme cases it is even possible to have the point *k*′ which does not belong to the cell face S_k. This source of inaccuracy is more linked to the *cell centered* configuration than to the discretization scheme, whichever one is used.

Table 8
*L*2 Norm and maximum of error for different grid sizes of the randomly distorted grid.

	Grid0: 51 × 51	Grid1: 102 × 102	Grid2: 204 × 204	Grid3: 408 × 408
*L*2 Norm	$1.698e^{-04}$	$5.27e^{-05}$	$1.41e^{-05}$	$3.64e^{-06}$
Max error	$1.9e^{-02}$	$6.65e^{-03}$	$3.57e^{-03}$	$1.83e^{-03}$

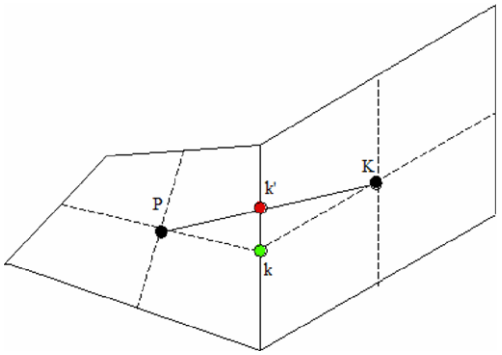

Fig. 10. Pathologic configuration due to grid distortion.

3.3. Quadrilateral and unstructured triangular grids

In a final numerical test we solved the Laplace equation, which governs the electrical potential distribution, in a trapezoidal cavity entirely defined by lengths $L1$, $L2$ and H, with boundary conditions indicated in Fig. 11.

This problem is solved by three different methods and/or codes.

FVQ refers to our finite volume code based on an IDC scheme on quadrilateral control volumes.
FVT refers to an unstructured version of our finite volume code based on an IDC scheme but applied to triangular control volumes.
FET refers to the Matlab (PDE toolbox) finite element solver on triangles.

Dimensionless lengths $L2$, H and l are kept fixed (at 4, 4 and 3) while the length $L1$ is extended to 10, 15, 20 and 30. When using a quadrilateral CV, increasing $L1$ is equivalent to increasing the θ angle. In the case of triangular control volumes or elements the mesher used is optimized in such a way that the internal angle is rather constant. So solving the Laplace equation with the **FVT** or **FET** solvers is a good way to get a solution which will not be sensitive to the skewness, unlike **FVQ**, which uses quadrilateral control volumes. Thus these two solutions would be interesting references to compare the solution obtained with the IDC approach using quadrilateral control volumes.

Only the extreme case $L1 = 30$ is presented. The first mesh used for the **FVT** and **FET** solvers has 3101 nodes and 6005 triangles. For the **FVQ** code, the mesh is made of 6400 control volumes (see Figs. 12 and 13) for objective comparison.

In Fig. 14 we have depicted the iso-contours of the solutions obtained with the three solvers. No differences can be observed. Therefore the finite volume discretization for triangles using our deferred correction scheme is completely consistent with the finite element discretization by the Matlab software. The solution of the **FVQ** solver is identical to the two previous ones.

In Fig. 15, we have plotted the vertical and horizontal profiles of the solution in the mid sections, $X_s = L1/2$ and $Y_s = H/2$. For all solvers it can be seen that the solution perfectly matches. These results show, from a more quantitative point of view, that the different types of discretization (finite volume or finite element) or the different shapes of control volumes (quadrilateral or triangular) lead to the same solution. Thus when the non-orthogonality of the grid is well taken into account in the discretization process of the diffusive flux, the solution becomes rather independent of the kind of discretization or shape of control volumes. This should be the rule with a consistent management of the non-orthogonality through diffusive flux discretization.

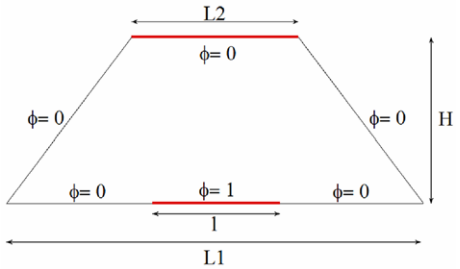

Fig. 11. Domain configuration and dimensionless boundary conditions for the third test case.

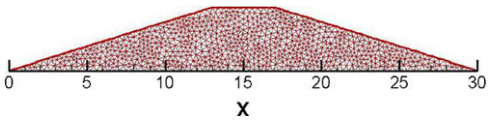

Fig. 12. Grid used for the **FVT** and **FET** solvers. $L1 = 30$.

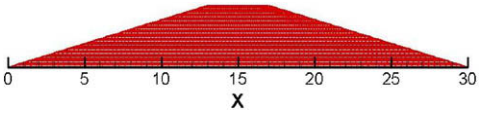

Fig. 13. Grid used for **FVQ** solver. $L1 = 30$.

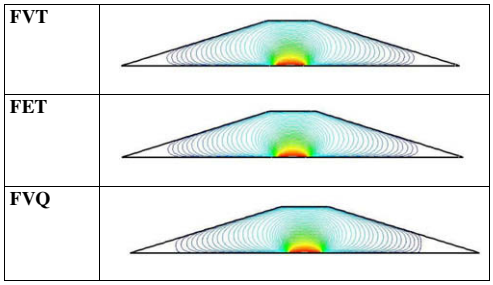

Fig. 14. Iso-contours of the solution for each solver.

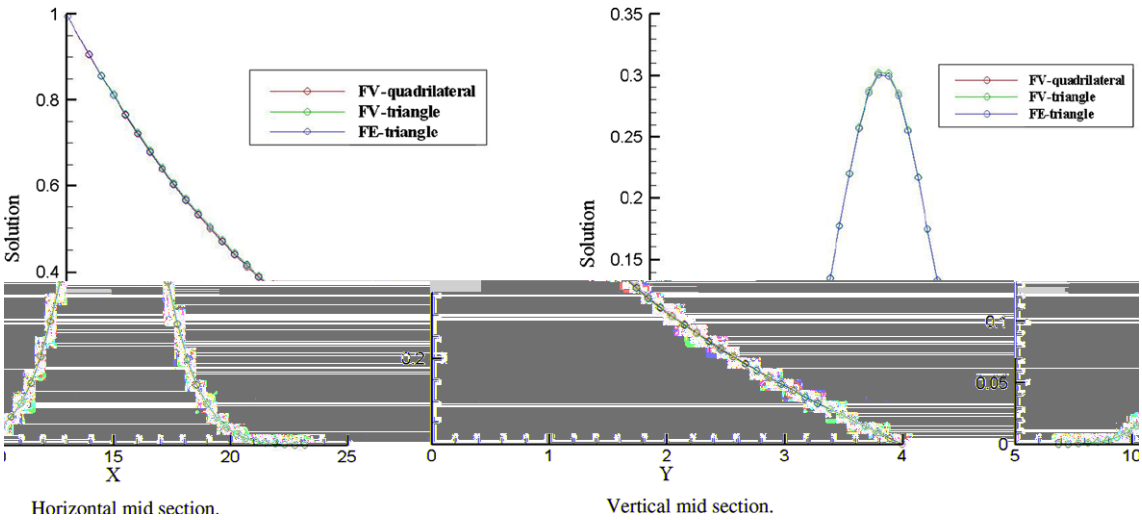

Fig. 15. Solution profile in different sections.

4. Summary and conclusion

In this article, a new Improved Deferred Correction scheme (IDC) suitable for the discretization of diffusive flux with a finite volume method in a context of extremely skewed mesh has been developed. Through a dedicated test case and according to different criteria, IDC has been shown to perform much better than SDC, which is still widely used. Whatever the skewness, the order of accuracy is always kept very near to 2, which is the theoretical order expected in our case. Therefore, the common idea disseminated by the CFD community, which asserts that the accuracy order of numerical schemes degrades as soon as control volumes are not orthogonal quadrilateral ones, must be reconsidered. It is certain that it is the way the diffusive flux is discretized that influences the accuracy order of the solution, rather than the shape of the control volume or the skewness itself. With a suitable discretization process the order of accuracy is fully recovered even for extre-

mely skewed grids. A complete check of the grid quality, which is often processed in industrial numerical codes before launching a run, could now become less restrictive since the borderline case for the internal angle is now pushed away.

In a final test, the IDC scheme shows good performance whatever the shape of the control volume, quadrilateral of triangular. The solution given by our scheme for these two types of control volumes has been compared with a finite element solver on the same triangular mesh. No noticeable differences were found, which establishes the validity of our new approach. Due to its performance, the new improved scheme should now be preferred by CFD developers to the standard approach.

References

[1] P.K. Khosla, S.G. Rubin, A diagonally dominant second-order accurate implicit scheme, Computers and Fluids 2 (1974) 207–209.
[2] S. Muzaferija, Adaptative finite volume method for flow prediction using unstructured meshes and multigrid approach, Ph.D. Thesis, Imperial College, 1994.
[3] J.H. Fergizer, M. Peric, Computational Methods for Fluid Dynamics, Springer, 2002.
[4] R. Eymard, R. Herbin, A cell-centered finite volume scheme on general meshes for Stokes equations in two spaces dimensions, Comptes Rendus de l'Académie des sciences Paris 337 (2003) 125–128.
[5] R. Eymard, T. Gallouet, R. Herbin, Error estimate for approximate solutions of a nonlinear convection–diffusion problem, Advances in Differential Equations 7 (2002) 419–440.
[6] Y. Coudière, J.P. Vila, P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection–diffusion, Mathematical Modelling and Numerical Analysis 33 (2000) 493–516.
[7] K. Domelevo, P. Omnès, A finite volume approach for the Laplace equation on almost arbitrary two dimensional grids, Mathematical Modelling and Numerical Analysis 39 (2005) 1203–1249.
[8] F. Hermeline, A finite volume method for the approximation of diffusion operators on distorted meshes, Journal of Computational Physics 160 (2000) 481–499.
[9] P. McCorquodale, P. Colella, D.P. Grote, J-L Vay, A node-centered local refinement algorithm for Poisson's equation in complex geometries, Journal of Computational Physics 201 (2004) 34–60.
[10] I. Demirdzic, Z. Lilek, M. Peric, A colocated finite volume method for predicting flows at all speeds, International Journal for Numerical Methods in Fluids 16 (1993) 1029–1050.
[11] I. Demirdzic, Z. Lilek, M. Peric, Fluid flow and heat transfer test problems for non-orthogonal grids: bench-mark solutions, International Journal for Numerical Methods in Fluids 15 (1992) 329–354.
[12] I. Demirdzic, M. Peric, Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries, International Journal for Numerical Methods in Fluids 10 (1990) 771–790.
[13] Y. Ahipo, Ph. Traoré, Nouvelle approche pour la discrétisation de flux diffusifs en volumes finis à forte obliquité, Comptes Rendus de l'Académie des sciences Paris 6 (2006) 380–386.
[14] Y. Ahipo, Contribution à la simulation numérique d'écoulements incompressibles en géométries complexes, thèse de doctorat, Université de Poitiers, 2007.
[15] J.H. Ferziger, M. Peric, Further discussion of numerical errors in CFD, International Journal for Numerical Methods in Fluids 23 (1996) 1263–1274.
[16] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing, 1980.
[17] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631–644.
[18] I. Gustafsson, On First and Second Order Symmetric Factorization Methods for the Solution of Elliptic Difference Equations, Chalmers University of Technology, 1978.
[19] T. Iwashita, T. Tami, M. Kanazawa, Parallel processing of ILU preconditioned BiCGSTAB solver using algebraic multi-color ordering method, in: PARELEC 2002, International Conference on Parallel Computing in Electrical Engineering, 2002, pp. 293–298.
[20] N. Takashi, T. Naoto, in: The Parallelization of the Incomplete LU Factorization on AP1000, Lecture Notes in Computer Science, vol. 1470, Springer, 1998.
[21] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, Journal of Computational Physics 129 (1996) 383–405.